The Dissertation Committee for Jialin Wu
certifies that this is the approved version of the following dissertation:

Incorporating External Information for Visual Question Answering

Committee:

Raymond J. Mooney, Supervisor

Greg Durrett

David Harwath

Dhruv Batra

Roozbeh Mottaghi

Incorporating External Information for Visual Question Answering

by

Jialin Wu

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2022

Acknowledgments

This dissertation would not have been possible without the support and encouragement from lots of people that helped me start pursuing a Ph.D. and go through these five years.

To begin with, I would like to express my most profound appreciation to my advisor Raymond J. Mooney for the guidance and inspiration during the last few years. I admire Ray's vision and knowledge of the development of open science and his passion throughout his career. I have always learned a lot from our weekly meetings and group meetings, where we discuss exciting ideas and future directions.

I am extremely grateful to the rest of my committee members, Greg Durrett, David Harwath, Dhruv Batra, and Roozebh Mottaghi, for providing helpful feedback on the dissertation.

My graduate school journey would never have been that smooth without the encouragement and feedback from friends and labmates. I am thankful to Prasoon Goyal, Sheena Panthaplackel, Angela Lin, Jierui Li, Albert Yu, Vanya Cohen, Aishwarya Padmakumar, Jiacheng Xu, Jifan Chen, Yasumasa Onoe, Yi Wang, Yan Han, Jiasen Lu.

I want to thank my undergrad advisor, Xiangyang Ji, for introducing me to AI research. Furthermore, I want to extend my sincere thanks to Prof. Qixing Huang and Prof. Chandrajit Bajaj for supporting my first Ph.D. year and equipping me with

a mathematics background. Finally, I am grateful to my internship mentors at AI2, Jiasen Lu, Roozbeh Mottaghi, and Ashish Sabharwal, for the feedback and advice on our work.

I owe what I am today to my parents, who provide me with the most comfortable environment to pursue my Ph.D. degree aboard.

Part of the work in this dissertation is supported by the DARPA XAI program and the Institute for Foundations of Machine Learning (IFML).

Incorporating External Information for Visual Question Answering

Publication No. _____

Jialin Wu, Ph.D.
The University of Texas at Austin, 2022

Supervisor: Raymond J. Mooney

Visual question answering (VQA) has recently emerged as a challenging multi-modal task and has gained popularity. The goal is to answer questions that query information associated with the visual content in the given image. Since the required information could be from both inside and outside the image, common types of visual features, such as object and attribute detection, fail to provide enough materials for answering the questions. External information, such as captions, explanations, encyclopedia articles, and commonsense databases, can help VQA systems comprehensively understand the image, reason following the right path, and access external facts. Specifically, they provide concise descriptions of the image, precise reasons for the correct answer, and factual knowledge beyond the image. In this dissertation, we present our work on generating image captions that are targeted to help answer a specific visual question. We use explanations to recognize the

critical objects to prevent the VQA models from taking language prior shortcuts. We introduce an approach that generates textual explanations and utilizes them to determine which answer is mostly supported. At last, we explore retrieving and exploiting external knowledge beyond the visual content, which is indispensable, to help answer knowledge-based visual questions.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Visual question answering (VQA) is a recently introduced "AI-complete" task that requires AI models to utilize multimodal knowledge beyond a single sub-domain. Building more trustworthy VQA models has many beneficial real-world daily life applications in education, medicine, and other areas that involve answering questions that require both visual and textual information. Also, it provides direct accessibility tools for helping the visually impaired. The goal of this task is to answer questions that query information associated with the visual content in the given image.

Figure 1.1: Different types of visual questions.

VQA [Antol et al., 2015, Agrawal et al., 2018, Park et al., 2018, Wang et al., 2018, Marino et al., 2019] is a broad topic that covers different categories

of questions requiring different types of information and reasoning skills. General visual questions mainly query the visual attributes of a specific object in the image, for example, the first question in Figure 1.1 asks about whether the man is wearing a full wetsuit. These questions require VQA models first to comprehend the question and the image and then localize the relevant objects or scenes to find the desired attributes. The middle one shows a commonsense visual question that the necessary information could be from both inside and outside the image.

However, our world and language systems are complicated, and the common types of visual features, such as object and attribute detection, fail to provide enough information to answer the questions. For example, the relations among multiple detected objects, the commonsense behind the objects, and the factual knowledge are indispensable to answering some types of questions. External information, such as captions, explanations, encyclopedia articles, and commonsense databases, can serve as information supplements to the question and the given image. This information helps VQA models better comprehend the image, understand the right reason for the answers, and access external facts. For example, the middle question in Figure 1.1 requires the VQA models to understand the common knowledge that the characters are Japanese and that means that is an Asian country. The last question shows a knowledge-based case where the external knowledge "*Teddy bear is named after President Roosevelt.*" is needed.

In Chapter 3, we explore using automatically generated image captions as the external information to help VQA. The captions highlight salient visual concepts in the image with a concise description of their relationships, complementing the

individual objects' features used in most VQA models. Moreover, we found that the generated question-relevant captions using our approach can serve stronger complements than normal image captions, further improving the VQA performance.

In Chapter 4, we utilize human explanations to debias the VQA model under changing answer distribution. These explanations highlight objects the VQA model should focus on when predicting the answer. We use visual and textual explanations to find this set of influential objects. We force this set of objects to contribute to the right answer more than other objects in the image.

In Chapter 5, we discuss our work on generating object-level faithful explanations for VQA that focuses on the same set of objects as the VQA model. In Chapter 6, we show that explanations could also be helpful when the answer distribution is unchanged. We generate explanations for a set of answer candidates and judge how well the explanations support the answer candidates. We obtain the explanations using either generation- or retrieval-based approaches.

In Chapter 7, we present our work on Outside-Knowledge VQA (OK-VQA), where external commonsense or facts are required. We introduce a multimodal answer validation (MAVEx) framework that judges the supportiveness of the retrieved knowledge for each answer candidate. On the knowledge side, we propose an entity-focused retrieval model to retrieve question-relevant knowledge, focusing on a set of critical objects for answering the question.

Chapter 2

Background and Related Work

This chapter presents the related work and background knowledge supporting this dissertation. To begin with, we introduce object detection, which serves as standard visual features for VQA. Then, we present the task of VQA and discuss some common challenges with typical VQA models. After that, we discuss three tasks that equip VQA models with relevant external information beyond the question and the image, i.e., image captioning, explanation generation, and passage retrieval.

2.1 Object Detection

Object detection is a foundation computer vision task aiming to discover salient semantic objects or scenes in images or videos. Detection provides an abstract characterization of the visual content and is commonly used as a visual feature for downstream tasks, such as image captioning and VQA. In practice, objects in an image or a single video frame are frequently expressed as bounding boxes [Ren et al., 2015b] or points [Zhou et al., 2019]. We introduce this task in the following two aspects, i.e., the categories of objects involved and common detection frameworks.

Categories of objects: Different from image classification, where only the image-level categories need to be predicted, the object detection task requires an AI system

to produce the precise bounding boxes of the objects of interest. Therefore, more human annotations must be collected on the objects' location. Despite the difficulties of collecting required annotations, the scope of object categories has been widely enlarged over the past decade. To begin with, the PASCAL VOC dataset [Everingham et al., 2010] introduces 11 object categories, COCO dataset expands the categories to 80 common objects. Visual genome [Krishna et al., 2017] adopts coarse annotation strategies, and [Anderson et al., 2018] cleans up the dataset to include 1,600 common types of objects. LVIS [Gupta et al., 2019] collects more than 1,200 object categories with a long-tail distribution. With the recent proliferation of large language models (LM) [Vaswani et al., 2017, Devlin et al., 2019], there is a recent trend towards open vocabulary object detection [Zareian et al., 2021] where the large LM is used to match the detected object labels.

Object detection frameworks: Most object detection models can be categorized as single- or two-stage framework. In the single-stage framework [Redmon et al., 2016], the models learn to simultaneously predict whether the current location contains an object and the object label. The advantages of the single-stage framework are the end-to-end training nature and the detection speed. On the other hand, the two-stage framework [Ren et al., 2015b] additional employs a region-proposal network (RPN) that separately raises object candidates without knowing the detail categories, and then a box classification network predicts the categories.

2.2 Visual Question Answer (VQA)

Visual Question Answering (VQA) [Antol et al., 2015, Hudson and Manning, 2019, Singh et al., 2019, Marino et al., 2019] has emerged as a challenging task where artificial intelligence systems predict answers by jointly analyzing both natural language questions and visual content. The scope of the visual questions spans every aspect of daily life that requires many capabilities for the VQA models, from basic object recognition to commonsense reasoning and analyzing external knowledge beyond the visual content. In this section, we first present some common challenges in VQA and the datasets that try to address them. Then, we introduce some basic VQA models designed for different types of visual questions.

2.2.1 Visual Question Datasets and Challenges

To begin with, Antol et al. [2015] defines the free form and open-ended VQA task by collecting human-annotated visual questions and answers on COCO images. The dataset has two main limitations, existing language priors and the scope of the visual questions.

The language prior refers to the fact that question types and their answers are highly correlated. For instance, questions that begin with "How many" are usually answered by either two or three. These language priors allow VQA systems to take a shortcut when answering questions by only focusing on the questions without reasoning about the visual content. In order to prevent this shortcut, Goyal et al. [2017] balanced the answer distribution so that at least two similar images with different answers for each question exist in VQA v2. Recently, Agrawal et al. [2018]

introduced a diagnostic reconfiguration of the VQA v2 dataset called VQA-CP, where the distribution of the QA pairs in the training set is significantly different from those in the test set. For example, most utensils in the training set are forks, and knives are the most common answer in the test set.

For the question scope, questions in [Antol et al., 2015] mainly require basic visual recognition instead of visual reasoning, commonsense and factual knowledge. In order to stress the need for visual reasoning, VQA-X [Park et al., 2018] split out some portion of VQAv2 where the questions are judged to require children older than nine years to answer. VQA-X dataset also provides both visual and textual explanations for the visual questions. Visual Commonsense Reasoning (VCR) [Zellers et al., 2019] is another related topic that requires a machine learning system to choose the right commonsense rationales. Knowledge base visual questions [Wang et al., 2018, 2017] requires VQA systems to extract helpful question-relevant knowledge from existing knowledge bases. Finally, there is a recent trend towards outside knowledge visual question answering (OK-VQA) [Marino et al., 2019] where open domain external knowledge outside the image is necessary.

2.2.2 VQA Models

As VQA requires VQA models to understand which part of the image is question-relevant, a large amount of attention-based deep-learning methods have been proposed for VQA, including top-down [Ren et al., 2015a, Fukui et al., 2016, Wu et al., 2016a, Goyal et al., 2017, Li et al., 2018a] and bottom-up attention methods [Anderson et al., 2018, Li et al., 2018b]. Specifically, a typical model

first extracts image features using a pre-trained CNN and then trains an RNN to encode the question, using an attention mechanism to focus on specific features of the image. Finally, both question and attended image features are used to predict the final answer. With the proliferation of large transformer models, more recent VQA models [Lu et al., 2019, Tan and Bansal, 2019, Liu et al., 2019, Li et al., 2019, Yu et al., 2019, Li et al., 2020a, Zhou et al., 2020, Chen et al., 2020, Lu et al., 2020] feeds both visual and textual embeddings into a multi-modal transformer, which is pre-trained on auxiliary tasks using large-scale multi-modal datasets such as [Sharma et al., 2018, Hudson and Manning, 2019, Kazemzadeh et al., 2014].

To train a VQA system to be right for the right reason, recent research has collected human visual attention [Das et al., 2017, Gan et al., 2017] highlighting image regions that most contribute to the answer. Then, the VQA systems try to align either the VQA system's attention [Qiao et al., 2018, Zhang et al., 2019] or the gradient-based visual explanation [Selvaraju et al., 2019] to the human attention.

In order to answer outside-knowledge visual questions, VQA models [Marino et al., 2019, Gardères et al., 2020, Zhu et al., 2020, Li et al., 2020b, Narasimhan et al., 2018, Marino et al., 2021, Gui et al., 2021] incorporate a retriever-reader framework that first retrieves textual knowledge relevant to the question and image and then "reads" this text to predict the answer. As an online free encyclopedia, Wikipedia is often used as the knowledge source for OK-VQA.

We list a few VQA models that are used in this dissertation.

UpDn: This is the original Top-Down Bottom-Up VQA [Anderson et al., 2018]

model, which uses a single layer GRU to encode questions. The question vector is then used to compute single-stage attention over the detected objects to produce attended visual features. Finally, a two-layer feed-forward network computes answer probabilities given the joint features of the question and visual content.

LXMERT: In order to learn richer representations for both questions and visual content, LXMERT [Tan and Bansal, 2019] uses transformers [Vaswani et al., 2017, Devlin et al., 2019] that learn multiple layers of attention over the input. In particular, it first learns nine layers over the input question and five layers over detected objects, then finally, another five layers of attention across the two modalities to produce the final joint representation.

ViLBERT-multi-task: ViLBERT-multi-task model utilizes auxiliary pretraining tasks, such as VQA, VCR, and visual entailment, to help enrich multimodal representation. Similar to LXMERT, the ViLBERT model incorporates large transformers to encode the visual and textual inputs.

2.3 Image Captioning

Image captioning is the task of generating a textual description of the image. The descriptions should typically cover salient objects and scenes and describe the relationships among them. The COCO dataset [Chen et al., 2015] provides over a million human captions (5 per image) focusing on the 80 categories of common daily objects. Conceptual captions [Sharma et al., 2018] collect over 12 millions image-caption pairs from the web.

Automatically Evaluating generated image captions is challenging as it is hard to tell whether a caption is good or not without human judgment. Machine translation automatic evaluation metrics, for example, BLEU[Papineni et al., 2002], ROUGE [Lin, 2004], and METEOR [Banerjee and Lavie, 2005] are frequently used in the image captioning task. Besides, some metrics are designed specifically for image captioning to evaluate different aspects, such as object hallucination issues [Rohrbach et al., 2018]. SPICE metric [Anderson et al., 2016] is proposed to measure the graph similarity between the generated captions and a set of human reference captions.

Most image captioning models are attention-based deep-learning models [Donahue et al., 2015, Karpathy and Fei-Fei, 2015, Vinyals et al., 2015, Luo et al., Liu et al., 2018, Wu and Mooney, 2019b]. The output words' probabilities at each step are trained to maximize the human captions' log-likelihood or some end evaluation metric ($e.g.$ CIDEr) directly using REINFORCE. Most of them encode the image using a CNN and build an attentional RNN ($i.e.$ GRU [Cho et al., 2014], LSTM [Hochreiter and Schmidhuber, 1997]) on top of the image features as a language model to generate image captions. As image captions are relatively easy to collect, the captioning task is frequently used as a pretraining task for large multi-modal transformer models.

2.4 Explanation Generation

While image captions provide general descriptions about the image, explanations reveal the actual reasons for answering the visual questions. This section

discusses three types of explanations for VQA, including visual, textual, and multimodal explanations. First, we introduce human-annotated explanations and then discuss common approaches to generate these explanations for each category.

2.4.1 Visual Explanation

The VQA-HAT dataset [Das et al., 2017] is a visual explanation dataset that collects human attention maps by giving human experts blurred images and asking them to determine where to deblur to answer a given visual question.

Many approaches have been proposed to visually explain decisions made by vision systems by highlighting relevant image regions. For example, GRAD-CAM [Selvaraju et al., 2017] analyzes the gradient space to find visual regions that most affect the decision. Attention mechanisms in VQA models can also be directly used to determine highly-attended regions and generate visual explanations.

2.4.2 Textual Explanation

Visual explanations highlight key image regions behind the decision but do not explain the reasoning process and crucial relationships between the highlighted regions. There are two textual explanation datasets, VQA-E [Li et al., 2018b] and VQA-X [Park et al., 2018]. Explanations in VQA-E are automatically refined versions of the most relevant captions from the COCO dataset [Chen et al., 2015], which have a larger scale but are not strictly explanatory. In contrast, the VQA-X dataset collects textual explanations for the questions that are judged to require children older than nine years to answer by directly asking the crowdsourced human

workers.

Therefore, there has been some work on generating textual explanations for decisions made by visual classifiers [Hendricks et al., 2016]. These models typically adopt a sequence-to-sequence framework that learns to generate the explanation by taking the question and the visual representation as input.

2.4.3 Multimodal Explanation

Multimodal explanations link textual and visual explanations [Park et al., 2018] together to present the critical image region and the reasons for the answer. Besides textual explanations, the VQA-X dataset also annotates important image segments, serving as a multimodal explanation dataset.

On the model side, previous works [Hendricks et al., 2018] mainly adopt a post-hoc approach that first generates multiple textual explanations and then filters out those that could not be grounded in the image.

2.5 Passage Retrieval

While image captions and explanations provide commonsense knowledge, factual knowledge beyond the visual content are indispensable for some visual questions, where we need to retrieve passages from online corpus. We present passage retrieval in the section by introducing general sparse and dense retrievers and then discussing the retrieval models for VQA.

2.5.1 Sparse Retrieval

Before the recent proliferation of transformer-based dense passage retrieval models [Karpukhin et al., 2020], previous works mainly explored sparse retrievers, such as TF-IDF and BM25 [Robertson and Zaragoza, 2009], that measure the similarity between the search query and candidate passage utilize weighted term matching. These sparse retrievers require no training signals on the relevancy of the passage and show solid baseline performances. However, exact term matching prevents them from capturing synonyms and paraphrases and understanding the semantic meanings of the query and the passages.

2.5.2 Dense Retrieval

To better represent semantics, dense retrievers [Karpukhin et al., 2020, Chen et al., 2021b, Lewis et al., 2022, Lee et al., 2021] extract deep representations for the query and the candidate passages using large pretrained transformer models. Most dense retrievers are trained using a contrastive objective that encourages the representation of the query to be more similar to the relevant passages than other irrelevant passages. During training, the passage with a high sparse retrieval score containing the answer is often regarded as a positive sample for the question-answering task. However, these positive passages may not fit the question's context and only serve as **weak supervision**. Therefore most dense retrievers fail to explicitly discover and utilize critical entities for the question [Chen et al., 2021b]. This often leads to overly general knowledge without a specific focus.

Dense passage retrieval for VQA: Motivated by the trends toward dense retrievers,

previous work has also applied them to OK-VQA. Qu et al. [2021] utilize Wikipedia as a knowledge source. Luo et al. [2021] crawl Google search results on the training set as a knowledge source. However, the weak training signals of passage retrieval become more problematic for VQA as the visual context of the question makes it more complex. Therefore, the "positive passage" become less likely to fit the visual context and provide suitable supervision. In order to better incorporate visual content, Gui et al. [2021] adopt an image-based knowledge retriever that employs the CLIP model [Radford et al., 2021] pretrained on large-scale multi-modal pairs as the backbone. However, question relevancy is not considered, so the retriever has to retrieve knowledge on every aspect of the image for different possible questions.

Phrase-based dense passage retrieval: The most relevant work to ours is phrase-based dense passage retrieval. Chen et al. [2021b] employ a separate lexical model that is trained to mimic the performance of a sparse retriever better at matching phrases. Lee et al. [2021] propose a DensePhrase model that extracts each possible phrase feature in the passage and only uses the most relevant phrase to measure the similarity between the query and passage. However, the training signals are still from the exact matching of the ground truth answers, and the phrases are parsed from the candidate passage, limiting the search scope.

Chapter 3

Generating Image Captions for VQA

3.1 Motivation and Chapter Overview

In recent years, VQA [Antol et al., 2015] and image captioning [Donahue et al., 2015, Rennie et al., 2017] have been widely studied in both the computer vision and NLP communities. Most recent VQA research [Lu et al., 2017, Pedersoli et al., 2017, Anderson et al., 2018, Lu et al., 2018] concentrates on directly utilizing visual input features, including detected objects, attributes, and relations between pairs of objects.

However, little VQA research works on exploiting textual descriptions of the image which are able to tersely encode the necessary information to answer the questions. This information could be richer than the visual features in that the sentences have fewer structural constraints and can easily include the attributes of and relationships among multiple objects. In fact, we observe that appropriate captions can be very useful for many VQA questions. In particular, we trained a model to answer visual questions for the VQA v2 challenge [Antol et al., 2015] only using the human-annotated captions **without** images and achieved a score of 59.6%, outperforming a large number of VQA models that use image features. Existing work using captions for VQA has generated **question-agnostic** captions using a

Human Captions :
1) A man on a blue surfboard on top of some rough water.
2) A young surfer in a wetsuit surfs a small wave.
3) A young man rides a surf board on a small wave while a man swims in the background.
4) A young man is on his surf board with someone in the background.
5) A boy riding waves on his surf board in the ocean.

Question 1: Does this boy have a full wetsuit on?
Caption: A young man wearing **wetsuit** surfing on a wave.
Question 2: What color is the board?
Caption: A young man riding a wave on a **blue surfboard**.

Figure 3.1: Examples of our generated question-relevant captions. During the training phase, our model selects the most relevant human captions for each question (marked by the same color).

pretrained captioner [Li et al., 2018a]. This approach can provide additional general information; however, this information is not guaranteed to be relevant to the given VQA question.

Therefore, we explore a novel approach that generates **question-relevant** image descriptions, which contain information that is directly relevant to a particular VQA question. Figure 3.1 shows examples of our generated captions given different questions. Then, we integrate the generated question-relevant captions as additional inputs to aid VQA.

This chapter is based on the [Wu et al., 2019]. In the following sections, we

present the captioning and VQA models in detail and then discuss the experimental results.

3.2 Question-Relevant Image Captioning Model

In this section, we present the question-relevant image captioning model that takes as input the question and the image. We start with the question and image representations and then introduce the caption generation model with the question relevancy criterion. We use $f(x)$ to denote fully-connected layers, where $f(x)$ $=$ LReLU$(Wx + b)$ with input features x and ignore the notation of weights and biases for simplicity, where these fc layers do not share weights. LReLU denotes a Leaky ReLU [He et al., 2015].

Image and question embedding: We use object detection as bottom-up attention [Anderson et al., 2018], which provides salient image regions with clear boundaries. In particular, we use a Faster R-CNN head [Ren et al., 2015b] in conjunction with a ResNet-101 base network [He et al., 2016] as our detection module. The detection head is first pre-trained on the Visual Genome dataset [Krishna et al., 2017] and can detect $1,600$ objects categories and 400 attributes. To generate an output set of image features **V**, we take the final detection outputs and perform non-maximum suppression (NMS) for each object category using an IoU threshold of 0.7. Finally, a fixed number of 36 detected objects for each image are extracted as the image features (a $2,048$ dimensional vector for each object) as suggested by Teney et al. [2017].

For the question embedding, we use a standard GRU [Cho et al., 2014] with $1,280$ hidden units and extract the output of the hidden units at the final time step as the question features $\mathbf{q}$. Following Anderson et al. [2018], the question features $\mathbf{q}$ and image feature set $\mathbf{V}$ are further embedded together to produce a question-attended image feature set $\mathbf{V}^q$ via question visual-attention $\mathbf{A}^{qv}$ as illustrated in Figure 3.2.

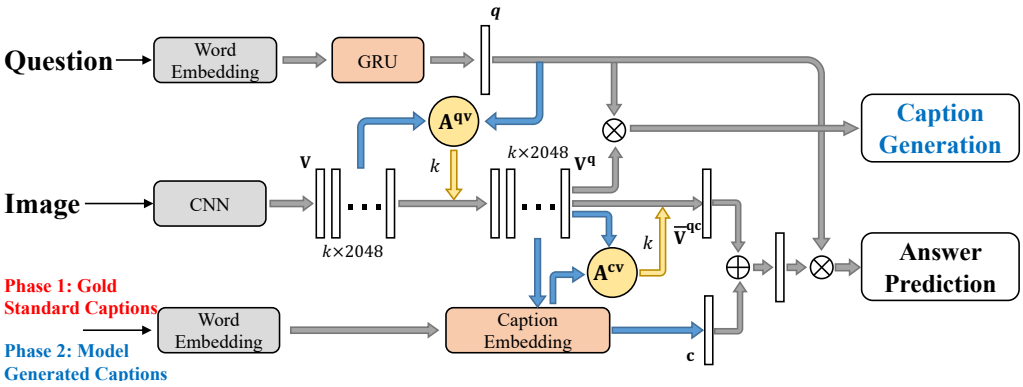


Figure 3.2: Overall structure of our model that generates question-relevant captions to aid VQA. Our model is first trained to generate question-relevant captions as determined in an online fashion in phase 1. Meantime, the human annotated captions are also used to pretrain the VQA part of the model. Then, the VQA model is fine-tuned with generated captions from the first phase to predict answers. $\otimes$ denotes element-wise multiplication and $\oplus$ denotes element-wise addition. Blue arrows denote fully-connected layers (fc) and yellow arrows denote attention embedding.

Caption generation model: We adopt an image captioning model similar to that of [Anderson et al., 2018]. The key difference between our module and theirs lies in the input features and the caption supervision. Specifically, we use the question-attended image features $\mathbf{V}^q$ as inputs and only use the most relevant caption, which is automatically determined in an online fashion (detailed below), for each question-image pair to train the captioning module. This ensures that only question-relevant

captions are generated.

Selecting relevant captions for Training: Previously, [Li et al., 2018b] selected relevant captions for VQA based on word similarities between captions and questions; however, their approach does not take into account the details of the VQA process. In contrast, during training, our approach dynamically determines for each problem the caption that will most improve VQA. We do this by updating with a shared descent direction [Wu et al., 2018] which decreases the loss for *both* captioning and VQA. This ensures a consistent target for both the image captioning module and the VQA module in the optimization process.

During training, we compute the cross-entropy loss for the i-th caption using Eq. 3.1, and back-propagate the gradients only from the most relevant caption determined by solving Eq. 3.2.

$$\mathcal{L}_i^c = -\sum_{t=1}^{T} \log(p(w_{i,t}^c | w_{i,t-1}^c)) \tag{3.1}$$

In particular, we require the inner product of the current gradient vectors from the predicted answer and the human captions to be greater than a positive constant ξ, and further select the caption that maximizes that inner product.

$$\begin{aligned} &\arg\max_i \sum_{k=0}^{K} \left(\frac{\partial \hat{s}_{\text{pred}}}{\partial \mathbf{v}_k^q} \right)^T \frac{\partial \log(p(\mathbf{W}_i^c))}{\partial \mathbf{v}_k^q} \\ &s.t. \sum_{k=0}^{K} \left(\frac{\partial \hat{s}_{\text{pred}}}{\partial \mathbf{v}_k^q} \right)^T \frac{\partial \log(p(\mathbf{W}_i^c))}{\partial \mathbf{v}_k^q} > \xi \end{aligned} \tag{3.2}$$

where the $\hat{s}_{\text{pred}}$ is the logit[1] for the predicted answer, $\mathbf{W}_i^c$ denotes the i-th human caption for the image and k traverses the K object features.

3.3 Integrating Captions in VQA

Now that we obtain the question-relevant captions, we present our approach to embed the generated captions in the VQA model.

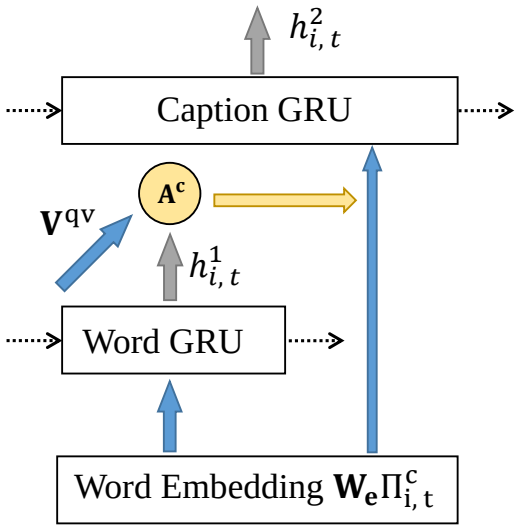


Figure 3.3: Overview of the caption embedding module. The Word GRU is used to generate attention to identify the relevant words in each caption, and the Caption GRU generates the final caption embedding. We use question-attended image features $\mathbf{V}^{qv}$ to compute the attention. Blue arrows denote fc layers and yellow arrows denote attention embedding.

Caption embedding: It takes as input the question-attended image feature set $\mathbf{V}^q$, question features $\mathbf{q}$, and C captions $\mathbf{W}_i^c = \{w_{i,1}^c, w_{i,2}^c, ..., w_{i,T}^c\}$, where T denotes the

[1]The input to the softmax function.

length of the captions and $i = 1, ..., C$ are the caption indices, and then produces the caption features $\mathbf{c}$.

The goals of the caption module are to serve as a knowledge supplement to aid VQA, and to provide additional clues to identify the relevant objects better and adjust the top-down attention weights. To achieve this, as illustrated in Figure 3.3, we use a two-layer GRU architecture. The first-layer GRU (called the Word GRU) sequentially encodes the words in a caption $\mathbf{W}_i^c$ at each time step as $h_{i,t}^1$.

$$h_{i,t}^1 = \text{GRU}(\mathbf{W}_e \Pi_{i,t}^c,\ h_{i,t-1}^1) \tag{3.3}$$

where $\mathbf{W}_e$ is the word embedding matrix, and $\Pi_{i,t}^c$ is the one-hot embedding for the word $w_{i,t}^c$.

Then, we design a caption attention module $\mathbf{A}^c$ which utilizes the question-attended feature set $\mathbf{V}^q$, question features $\mathbf{q}$, and $h_{i,t}^1$ to generate the attention weight on the current word in order to indicate its importance. Specifically, the Word GRU first encodes the words embedding $\Pi_{i,t}^c$ in Eq. 3.3, and then we feed the outputs $h_{i,t}^1$ and $\mathbf{V}^q$ to the attention module $\mathbf{A}^c$ as shown in Eq. 3.6.

$$\overline{\mathbf{v}}^q = \sum_{k=1}^{K} \mathbf{v}_k^q \tag{3.4}$$

$$a_{i,t}^c = h_{i,t}^1 \circ f(\overline{\mathbf{v}}^q) + h_{i,t}^1 \circ f(\mathbf{q}) \tag{3.5}$$

$$\alpha_{i,t}^c = \sigma(a_{i,t}^c) \tag{3.6}$$

where σ denotes the sigmoid function, and K is the number of objects in the bottom-up attention.

Next, the attended words in the caption are used to produce the final caption representation in Eq. 3.7 via the Caption GRU. Since the goal is to gather more information, we perform element-wise max pooling across the representations of all of the input captions $\mathbf{c}_i$ in Eq. 3.9.

$$h^2_{i,t} = \text{GRU}(\alpha^c_{i,t}\mathbf{W}_e\Pi^c_{i,t},\ h^2_{i,t-1}) \tag{3.7}$$

$$\mathbf{c}_i = f(h^2_{i,T}) \tag{3.8}$$

$$\mathbf{c} = max(\mathbf{c}_i) \tag{3.9}$$

where max denotes the element-wise max pooling across all of caption representations $\mathbf{c}_i$ of the image.

The generated captions are usually capable of capturing relations among the question-relevant objects; however these relations are absent in the bottom-up attention. Therefore, our VQA module utilizes the caption embeddings $\mathbf{c}$ to adjust the top-down attention weights in VQA in order to produce the final caption-attended features $\overline{\mathbf{v}}^{qc}$ in Eq. 3.12:

$$a^{cv}_k = f(f(\mathbf{c}) \circ f(\overline{\mathbf{v}}^q_k)) \tag{3.10}$$

$$\alpha^{cv}_k = softmax(a^{cv}_{c,k}) \tag{3.11}$$

$$\overline{\mathbf{v}}^{qc} = \sum_k^K \mathbf{v}^q_k \alpha^{cv}_k \tag{3.12}$$

where k traverses the K objects features.

To better incorporate the information from the captions into the VQA process, we add the caption features $\mathbf{c}$ to the attended image features $\overline{\mathbf{v}}^{qc}$, and then element-wise

multiply by the question features as shown in Eq. 3.13:

$$\mathbf{h} = \mathbf{q} \circ (f(\overline{\mathbf{v}}^{qc}) + f(\mathbf{c})) \tag{3.13}$$

$$\hat{s} = \sigma(f(\mathbf{h})) \tag{3.14}$$

We frame the answer prediction task as a multi-label regression problem [Anderson et al., 2018]. In particular, we use the soft scores in the gold-standard VQA-v2 data (which are used in the evaluation metric), as labels to supervise the sigmoid-normalized predictions as shown in Eq. 6.3:

$$\mathcal{L}^{vqa} = -\sum_{j=1}^{N} s_j \log \hat{s}_j + (1 - s_j) \log(1 - \hat{s}_j) \tag{3.15}$$

where the index j runs over N candidate answers and s are the soft answer scores.

In case of multiple feasible answers, the soft scores capture the occasional uncertainty in the ground-truth annotations. As suggested by Teney et al. [2017], we collect the candidate answers that appear more than 8 times in the training set, which results in $3,129$ answer candidates. The overview of our caption aided VQA model is shown in Figure 3.2. We use a modified UpDn attention model as the suggested in [Singh et al., 2018].

3.4 Experimental Setup and Results

We perform extensive experiments and ablation studies to evaluate our joint model on VQA.

3.4.1 Datasets and evaluation metrics

VQA dataset: We use the VQA v2.0 dataset [Antol et al., 2015] for the evaluation of our proposed joint model, where the answers are balanced in order to minimize the effectiveness of learning dataset priors. This dataset is used in the VQA 2018 challenge and contains over 1.1M questions from the over 200K images in the MSCOCO 2015 dataset [Chen et al., 2015].

Following Anderson et al. [2018], we perform standard text pre-processing and tokenization. In particular, questions are first converted to lower case and then trimmed to a maximum of 14 words, and the words that appear less than 5 times are replaced with an "<unk>" token. To evaluate answer quality, we report accuracies using the official VQA metric using soft scores, which accounts for the occasional disagreement between annotators for the ground truth answers.

Image captioning dataset: We use the MSCOCO 2014 dataset [Chen et al., 2015] for the image caption module. To maintain consistency with the VQA tasks, we use the dataset's official configuration that includes $82,372$ images for training and $40,504$ for validation. Similar to the VQA question pre-processing, we first convert all sentences to lower case, tokenizing on white spaces, and filtering words that do not occur at least 5 times.

Training and implementation details: We train our joint model using the AdaMax optimizer [Kingma and Ba, 2015] with a batch size of 384 and a learning rate of 0.002 as suggested by Teney et al. [2017]. We use the validation set for VQA v2 to tune the initial learning rate and the number of epochs, yielding the highest

overall VQA score. We use $1,280$ hidden units in the question embedding and attention model in the VQA module with 36 object detection features for each image. For captioning models, the dimension of the LSTM hidden state, image feature embedding, and word embedding are all set to 512. We also use Glove vectors [Pennington et al., 2014] to initialize the word embedding matrix in the caption embedding module.

We initialize the training process with human annotated captions from the COCO dataset [Chen et al., 2015] to pre-train the VQA model and caption-generation modules for 20 epochs. After that, we generate question-relevant captions for all question-image pairs in the COCO train, validation, and test sets. In particular, we sample 5 captions per question-image pair. We fine-tune our model using the generated captions with 0.25 $\times$ learning rate for another 10 epochs.

3.4.2 VQA Results

We first report the experimental results on the VQA task and compare our results with the state-of-the-art methods in this section. After that, we perform ablation studies to verify the contribution of additional knowledge from the generated captions, and the effectiveness of using caption representations to adjust the top-down visual attention weights.

As demonstrated in Table 3.1, our single model outperforms other state-of-the-art single models by a clear margin, $i.e.$ 2.06%, which indicates the effectiveness of including caption features as additional inputs. In particular, we observe that our single model outperforms other methods, especially in the 'Num' and 'Other'

	Test-standard			
	Yes/No	Num	Other	All
Prior [Goyal et al., 2017]	61.20	0.36	1.17	25.98
Language-only [Goyal et al., 2017]	67.01	31.55	27.37	44.26
MCB [Fukui et al., 2016]	78.82	38.28	53.36	62.27
Up-Down [Anderson et al., 2018]	82.20	43.90	56.26	65.32
VQA-E [Li et al., 2018b]	83.22	43.58	56.79	66.31
Ours(single)	**84.69**	**46.75**	**59.30**	**68.37**
Ours(Ensemble-10)	**86.15**	**47.41**	**60.41**	**69.66**

Table 3.1: Comparison of our results on VQA with the state-of-the-art methods on the test-standard data. Accuracies in percentage (%) are reported.

categories. This is because the generated captions are capable of providing more numerical clues for answering the 'Num' questions, since the captions can describe the number of relevant objects and provide general knowledge for answering the 'Other' questions. Furthermore, an ensemble of 10 models with different initialization seeds results in a score of 69.7% for the test-standard set.

Figure 3.4 shows several examples of our generated question-relevant captions. These examples illustrate how different captions are generated for the same image when the question is changed. They also show how the objects in the image that are important to answering the question are described in the question-relevant captions.

Comparison between using generated and human captions: Next, we analyze the difference between using automatically generated captions and using those provided by human annotators. In particular, we train our model with generated question-agnostic captions using the Up-Down [Anderson et al., 2018] captioner,

Figure 3.4: Examples of our generated question-relevant captions. The influential objects with attention weights greater than 0.1 are indicated by bounding boxes (annotated with their visual attention weights in the blue box), and the gray-scale levels in the caption words indicate the word attentions from the caption embedding module.

question-relevant captions from our caption generation module, and human annotated captions from the COCO dataset.

As demonstrated in Table 3.2, our model gains about 4% improvement from using human captions and 2.5% improvement from our generated question-relevant captions on the validation set. This indicates the insufficiency of directly answering visual questions using a limited number of detection features, and the utility of incorporating additional information about the images. We also observe that our generated question-relevant captions trained with our caption selection strategy provide more helpful clues for the VQA process than the question-agnostic Up-Down captions,

	Validation
Up-Down [Anderson et al., 2018]	63.2
Ours with Up-Down captions	64.6
Ours with our generated captions	65.8
Ours with human captions	**67.1**

Table 3.2: Comparison of the performance using generated and human captions. Both of them provide significant improvements to the baseline model. However, there is still a reasonable gap between generated and human captions.

outperforming their captions by 1.2%.

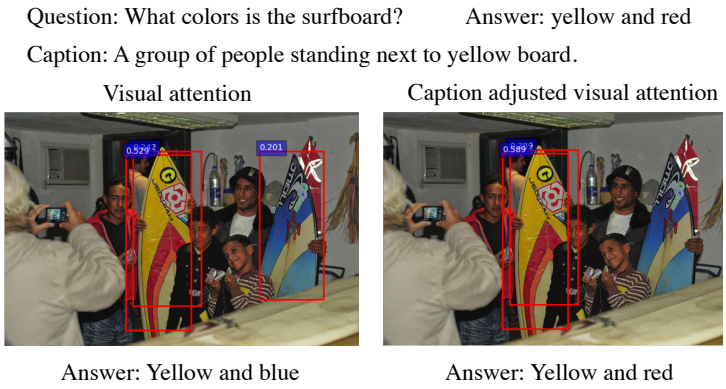


Figure 3.5: An example of caption attention adjustment. The question-relevant caption helps the VQA module adjust the visual attention from both the yellow board and the blue sail to the yellow board onl, which further leads to the right answer.

Effectiveness of adjusting top-down attention: In this section, we quantitatively analyze the effectiveness of utilizing captions to adjust the top-down attention weights, in addition to the advantage of providing additional information. In particular, we compare our model with a baseline version where the top-down attention-weight adjustment factor $\mathbf{A}^{cv}$ is manually set to 1.0 (resulting in no adjustment).

As demonstrated in Tables 3.3 and 3.4, we observe an improvement when using caption features to adjust the attention weights. This indicates that the caption features help the model to more robustly locate the objects that are helpful to the VQA process. We use $w\ \mathcal{CAA}$ to indicate with caption attention adjustment and $w/o\ \mathcal{CAA}$ to indicate without it. Figure 3.5 illustrates an example of caption attention adjustment. Without $\mathcal{CAA}$, the top-down visual attention focuses on both the yellow surfboard and the blue sail, generating the incorrect answer "yellow and blue.". However, with "yellow board" in the caption, the caption attention adjustment ($\mathcal{CAA}$) helps the VQA module focus attention just on the yellow surfboard, thereby generating the correct answer "yellow and red" (since there is some red coloring in the surfboard).

	Test-standard			
	All	Yes/No	Num	Other
Up-Down	65.3	82.2	43.9	56.3
Ours $w/o\ \mathcal{CAA}$	67.4	84.0	44.5	57.9
Ours $w\ \mathcal{CAA}$	68.4	84.7	46.8	59.3

Table 3.3: Evaluation of the effectiveness of caption-based attention adjustment ($\mathcal{CAA}$) on the test-standard data. Accuracies in percentage (%) are reported.

	Validation			
	All	Yes/No	Num	Other
Up-Down	63.2	80.3	42.8	55.8
Ours $w/o\ \mathcal{CAA}$	65.2	82.1	43.6	55.8
Ours $w\ \mathcal{CAA}$	65.8	82.6	43.9	56.4

Table 3.4: Evaluation of the effectiveness of $\mathcal{CAA}$ on the validation data. Accuracies in percentage (%) are reported.

3.5 Chapter Summary

In this chapter, we have explored how generating question-relevant image captions can improve VQA performance. In particular, we present a model which jointly generates question-related captions and uses them to provide additional information to aid VQA. This approach only utilizes existing image-caption datasets, automatically determining which captions are relevant to a given question. In particular, we design the training algorithm to only update the network parameters in the optimization process when the caption generation and VQA tasks agree on the direction of change. As a result, our single model joint system outperforms the state-of-the-art single model for VQA at the time of submission (May 2018).

Chapter 4

Self-Critical Reasoning for Debiasing VQA

4.1 Motivation and Chapter Overview

The state-of-the-art VQA systems [Fukui et al., 2016, Anderson et al., 2018, Agrawal et al., 2018, Andreas et al., 2016, Hu et al., 2018, Yang et al., 2016, Selvaraju et al., 2019, Wu et al., 2019, Jiang et al., 2018, Kim et al., 2018, Ramakrishnan et al., 2018] achieve high performance when the training and test question-answer (QA) pairs are sampled from the same distribution. However, most of these systems fail to generalize to test data with a substantially different QA distribution. In particular, their performance drops catastrophically on the recently introduced Visual Question Answering under Changing Priors (VQA-CP) [Agrawal et al., 2018] dataset. The strong language priors encourage systems to blindly capture superficial statistical correlations in the training QA pairs and simply output the most common answers, instead of reasoning about the relevant image regions on which a human would focus. For example, about 40% of questions that begin with "what sport" have the answer "tennis"; systems tend to learn to output "tennis" for these questions regardless of image content.

A number of recent VQA systems [Trott et al., 2018, Zhang et al., 2019, Selvaraju et al., 2019, Qiao et al., 2018] learn to not only predict correct answers but

also be "right for the right reasons" [Ross et al., 2017, Selvaraju et al., 2019]. These systems are trained to encourage the network to focus on regions in the image that humans have somehow annotated as important (which we will refer to as "important regions."). However, the network often focuses on these important regions even when it produces a wrong answer. Previous approaches do nothing to actively discourage this phenomenon, which we have found occurs quite frequently.[1] For example, as shown in Figure 4.1, we ask the VQA system, "What is the man eating?". Unfortunately, the baseline system predicts "hot dog" but focuses on the banana because the hot dog appears much more frequently in the training data. What's worse, this error is hard to detect when only analyzing the correct answer "banana" that has been successfully grounded in the image.

We present a "self-critical" approach that directly criticizes incorrect answers' sensitivity to the important regions to address this issue. First, for each QA, we determine the important region that most influences the network's prediction of the correct answer. We then penalize the network for focusing on this region when its predicted answer for this question is *wrong*.

This chapter is based on [Wu and Mooney, 2019c]. In the following sections, we present the approach to construct a set of potentially influential objects and then discuss the two new objectives that encourage the VQA model to focus on them.

[1] We exam these situations by designing a metric called false sensitivity rate ($\mathcal{FSR}$) in the experiment section in this chapter.

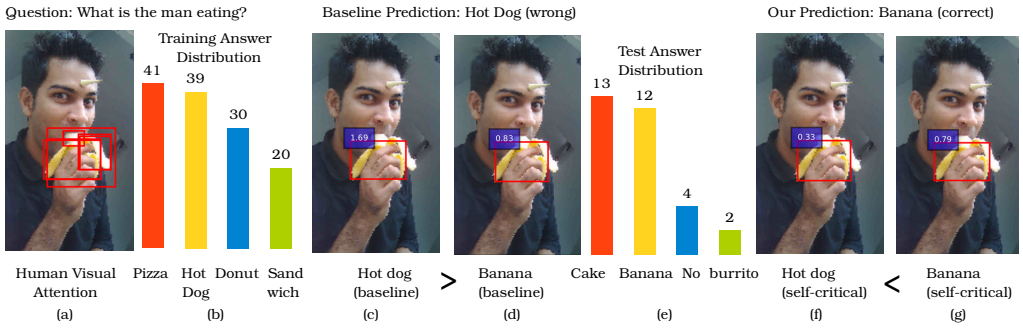


Figure 4.1: example of a common answer misleading the prediction even though the VQA system has the right reasons for the correct answer. Figure (a) shows the important regions extracted from human visual attention. Figure (b), (e) show the answers' distribution for the question "What is the man eating?" in the training and test dataset. Figure (c), (d) show the most influential region for the prediction "hot dog" and "banana" using the baseline UpDn VQA system and Figure (f), (g) show the influential region for the prediction "hot dog" and "banana" using the VQA system after being trained with our self-critical objective. The number on the bounding box shows the answer's sensitivity to the object.

4.2 Human Explanation Hints

Our approach ideally requires identifying important regions that a human considers most critical in answering the question. However, directly obtaining such a clear set of influential objects from either visual or textual explanations is hard, as the visual explanations also highlight the neighbor objects around the most influential one, and grounding textual explanations in images is still an active research field. We relax this requirement by identifying a proposed set of influential objects $\mathcal{I}$ for each QA pair. This set may be noisy and contain some irrelevant objects, but we assume that it at least includes the most relevant object. We explore three separate methods for constructing this proposal set, as described below:

Construction from visual explanations: Following HINT [Selvaraju et al., 2019], we use the VQA-HAT dataset [Das et al., 2017] as the visual explanation source. HAT maps contain a total of $59,457$ image-question pairs, corresponding to approximately 9% of the VQA-CP training and test set. We also inherit HINT's object scoring system that is based on the normalized human attention map energy inside the proposal box relative to the normalized energy outside the box. We score each detected object from the bottom-up attention and build the potential object set by selecting the top $|\mathcal{I}|$ objects.

Construction from textual explanations: Recently, [Park et al., 2018] introduced a textual explanation dataset that annotates $32,886$ image-question pairs, corresponding to 5% of the entire VQA-CP dataset. To extract the potential object set, we first assign part-of-speech (POS) tags to each word in the explanation using the spaCy POS tagger [Honnibal and Montani, 2017] and extract the nouns in the sentence. Then, we select the detected objects whose cosine similarity between the Glove embeddings [Pennington et al., 2014] of their category names and any of the extracted nouns' is greater than 0.6. Finally, we select the $|\mathcal{I}|$ objects with the highest similarity.

Construction from questions and answers: Since the above explanations may not be available in other datasets, we also consider a simple way to extract the proposal object set from just the training QA pairs alone. The method is quite similar to the way we construct the potential set from textual explanations. The only difference is that instead of parsing the explanations, we parse the QA pairs and extract nouns from them.

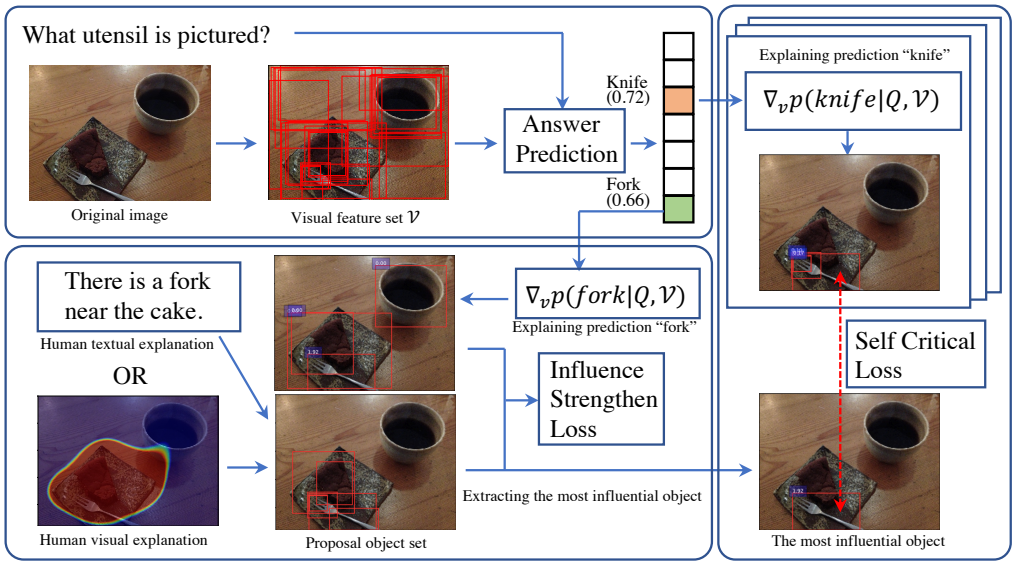


Figure 4.2: Model overview. In the left top block, the base UpDn VQA system first detects a set of objects and predicts an answer. We then analyze the correct answer's sensitivity (Fork) to the detected objects via visual explanation and extract the most influential one in the proposal object set as the most influential object, which is also further strengthened via the influence strengthen loss (left bottom block). Finally, we analyze the competitive incorrect answers' sensitivities (Knife) to the most influential object and criticize the sensitivity until the VQA system answers the question correctly (right block). The number on a bounding box is the answer's sensitivity to the given object.

4.3 Debiasing VQA Model

In this section, we present our self-critical approach to prevent the most common answer from dominating the correct answer given the proposal sets of influential objects. Figure 4.2 shows an overview of our approach. Besides the UpDn VQA system (left top block), our approach contains two other components, we first recognize and strengthen the most influential objects (left bottom block), and then we criticize incorrect answers that are more highly ranked than the correct answer

and try to make them less sensitive to these key objects (right block). As recent research suggests that gradient-based methods more faithfully represent a model's decision making process [Selvaraju et al., 2019, Zhang et al., Wu et al., 2018, Jain and Wallace, 2019], we use a modified GradCAM [Selvaraju et al., 2017] to compute the answer a's sensitivity to the i-th object features $\mathbf{v}_i$ as shown in Eq. 4.1.[2]

$$\mathcal{S}(a, \mathbf{v}_i) \coloneqq \left(\nabla_{\mathbf{v}_i} P(a|V, q)\right)^T \mathbf{1} \tag{4.1}$$

There are two modifications to GradCAM: (1) ReLU units are removed, (2) gradients are no longer weighted by their feature vectors. This is because negative gradients on the inputs to a ReLU are valuable evidence against the current prediction. Therefore, there is no need to zero them out with a ReLU. Also, before they are weighted by the feature vectors, the gradients indicate how small changes in any direction influence the final prediction. If weighted by the feature vectors, the output tends to reflect the influence caused *only* by *existing* attributes of the objects, thereby ignoring other potential attributes that may appear in the test data.

4.3.1 Recognizing and Strengthening Influential Objects

Given a proposal object set $\mathcal{I}$ and the entire detected object set $\mathcal{V}$, we identify the object that the correct answer is most sensitive to and further strengthen its sensitivity. We first introduce a sensitivity violation term $\mathcal{SV}(a, \mathbf{v}_i, \mathbf{v}_j)$ for answer a and the i-th and j-th object features $\mathbf{v}_i$ and $\mathbf{v}_j$ as the amount of sensitivity that $\mathbf{v}_j$

[2] $\mathbf{1}$ denotes a vector with all 1's.

surpasses $\mathbf{v}_i$, as shown in Eq. 4.2.

$$\mathcal{SV}(a, \mathbf{v}_i, \mathbf{v}_j) = \max\big(\mathcal{S}(a, \mathbf{v}_j) - \mathcal{S}(a, \mathbf{v}_i), 0\big) \tag{4.2}$$

Based on the assumption that the proposal set contains at least one influential object that a human would use to infer the answer, we impose the constraint that the most sensitive object in the proposal set should not be less sensitive than any object outside the proposal set. Therefore, we introduce the influence strengthen loss $\mathcal{L}_{infl}$ in Eq. 4.3:

$$\mathcal{L}_{infl} = \min_{\mathbf{v}_i \in \mathcal{I}} \Big(\sum_{\mathbf{v}_j \in \mathcal{V} \setminus \mathcal{I}} \mathcal{SV}(a_{gt}, \mathbf{v}_i, \mathbf{v}_j) \Big) \tag{4.3}$$

where the a_{gt} denotes the ground truth answer. The key differences between our influence strengthen loss and the ranking-based HINT loss are that (1) we relax the unnecessary constraint that the objects should follow the exact human ranking, and (2) it is easier to adapt to different types of explanation ($e.g.$ textual explanations) where such detailed rankings are not available.

4.3.2 Criticizing Incorrect Dominant Answers

Next, for the incorrect answers ranked higher than the correct answer, we attempt to decrease the sensitivity of the influential objects. For example, in VQA-CP, bedrooms are the most common room type. Therefore, during testing, systems frequently incorrectly classify bathrooms (which are rare in the training data) as bedrooms. Since humans identify a sink as an influential object when identifying bathrooms, we want to decrease the influence of sinks on concluding bedroom.

In order to address this issue, we design a self-critical objective to criticize the VQA systems' incorrect but competitive decisions based on the most influential object $\mathbf{v}^*$ to which the correct answer is most sensitive as defined in Eq. 4.4.

$$\mathbf{v}^* = \arg\min_{\mathbf{v}_i \in \mathcal{I}} \Big(\sum_{\mathbf{v}_j \in \mathcal{V} \setminus \mathcal{I}} \mathcal{SV}(a_{gt}, \mathbf{v}_i, \mathbf{v}_j) \Big) \tag{4.4}$$

Specifically, we extract a bucket of at most B predictions with higher confidence than the correct answer $\mathcal{B} = \{a_1, a_2, ..., a_{|\mathcal{B}|}\}$ and utilize the proposed self-critical loss $\mathcal{L}_{crit}$ to directly minimize the weighted sensitivities of the answers in the bucket $\mathcal{B}$ to the selected most influential object, as shown in Eq. 4.5.

$$\mathcal{L}_{crit} = \sum_{a \in \mathcal{B}} w(a)(\mathcal{S}(a, \mathbf{v}^*) - \mathcal{S}(a_{gt}, \mathbf{v}^*)) \tag{4.5}$$

where a_{gt} denotes the ground truth answer. Because several answer candidates could be similar ($e.g.$ cow and $cattle$), we weight the sensitivity gaps in Eq. 4.5 by the cosine distance between the answers' 300-d Glove embeddings [Pennington et al., 2014], $i.e.$ $w(a) = cosine\_dist(Glove(a_{gt}),\ Glove(a))$. In the multi-word answer case, the Glove embeddings of these answers are computed as the sum of the individual word's Glove embeddings.

4.4 Experimental Setup and Results

First, we present experiments on a simple synthetic dataset to illustrate basic aspects of our approach. We then present experimental results on the VQA-CP (Visual Question Answering with Changing Priors) [Agrawal et al., 2018] dataset where the QA pairs in the training data and test data have significantly different

distributions. We compare our self-critical system's VQA performance with the start-of-the-art systems via the standard evaluation metric. After that, we perform ablation studies to verify the contribution of strengthening the influential objects and criticizing competitive answers. Finally, we show some qualitative examples to illustrate the effectiveness of criticizing the incorrect answers' sensitivity.

4.4.1 Synthetic Task Results

We manually created a dataset where the inputs are drawn from a mixture of two Gaussians, *i.e.* $\mathcal{N}_1 = \mathcal{N}([-3, 3]^T, 2I_2)$ and $\mathcal{N}_2 = \mathcal{N}([3, 3]^T, 2I_2)$, where each distribution defines a category. In order to ensure the training and test data have different category distributions, we intentionally assign different weights to the two components. In particular, during training, the examples are drawn from $\mathcal{N}_1$ with probability p, and during test, the examples are drawn from $\mathcal{N}_1$ with probability $1 - p$. We examine the effectiveness of our self-critical approach varying p from 0.05 to 0.5 ($i.e.$ 0.05, 0.1, 0.2, 0.5) (0.5 means no train/test difference). In these experiments, we use the obvious human explanation that the first channel (x-axis) is important for all training examples. We use a 15-layer feed-forward neural network with 256 hidden units and 1000 examples for both training and test in all of our experiments. We use Adam to optimize our model with a learning rate of 1e-3 during pre-training (100 epochs) with binary cross-entropy loss, and 1e-5 during fine-tuning (50 epochs) with our self-critical approach. The influence strengthening loss weight and self-critical loss weight are set to 20 and 1000, respectively. The results in Fig. 4.3 shows that the self-critical approach helps shift the decision boundary towards

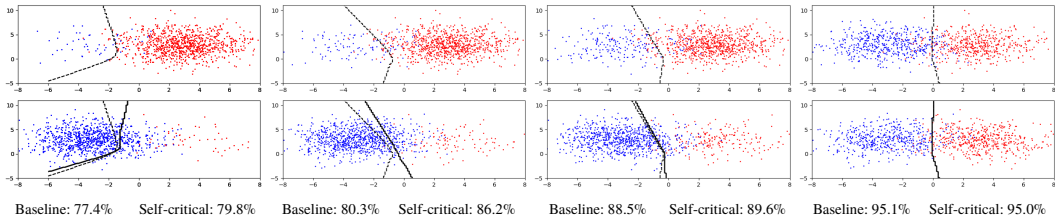


Figure 4.3: Decision boundaries and test set accuracies on synthetic data with various class ratios p, which is varied from 0.05, 0.1, 0.2, to 0.5 from left to right. The training data is shown in the top row, testing in the bottom. Red and blue colors denote different categories. Dashed lines and solid lines denote the boundaries of the pretrained and fine-tuned models, respectively.

the correct, unbiased position, increasing robustness and accuracy on the test data.

4.4.2 VQA Results

Implementation and Training Details:

Training Details: We first pre-train our base UpDn VQA system on the VQA-CP training set using standard VQA loss $\mathcal{L}_{vqa}$ (binary cross-entropy loss with soft scores as supervision) with the Adam optimizer [Kingma and Ba, 2015] for at most 20 epochs. As suggested in [Teney et al., 2017], the learning rate is fixed to 10e-3 with a batch size of 384 during the pre-training process, and we use $1,280$ hidden units in the base UpDn VQA system. We use a modified UpDn attention model as the suggested in [Singh et al., 2018]. Then, we fine-tune our system to recognize important objects using $\mathcal{L}_{vqa} + \lambda_{infl}\mathcal{L}_{infl}$ with a learning rate of 10e-5 for at most 15 epochs on the intersection of VQA-X and VQA-CP training set. We initialize the model with the best model from the pre-train stage. In this stage, we also find the best influence strengthening loss weight $\lambda^{\star}_{infl}$. Finally, we fine-tune the system

with the joint loss $\mathcal{L} = \mathcal{L}_{vqa} + \lambda^{\star}_{infl}\mathcal{L}_{infl} + \lambda_{crit}\mathcal{L}_{crit}$ for at most 15 epochs with a learning rate of 10e-5 on the intersection of VQA-X and VQA-CP training set. The bucket size $|B|$ of the competitive answers is set to 5 because we observed that the top-5 overall score of the pre-trained system on the VQA-CP dataset achieves 80.4%, and increasing the bucket size only marginally improves the score.

Implementation: We implemented our approach on top of the original UpDn system. The base system utilizes a Faster R-CNN head [Ren et al., 2015b] in conjunction with a ResNet-101 base network [He et al., 2016] as the object detection module. The detection head is pre-trained on the Visual Genome dataset [Krishna et al., 2017] and is capable of detecting $1,600$ objects categories and 400 attributes. UpDn takes the final detection outputs and performs non-maximum suppression (NMS) for each object category using an IoU threshold of 0.7. Then, the convolutional features for the top 36 objects are extracted for each image as the visual features, $i.e.$ a $2,048$ dimensional vector for each object. For question embedding, following [Anderson et al., 2018], we perform standard text pre-processing and tokenization. In particular, questions are first converted to lower case and then trimmed to a maximum of 14 words, and the words that appear less than 5 times are replaced with an "<unk>" token. A single layer GRU [Cho et al., 2014] is used to sequentially process the word vectors and produce a sentential representation for the pre-processed question. We also use Glove vectors [Pennington et al., 2014] to initialize the word embedding matrix when embedding the questions. The size of proposal object set is set to 6.

VQA performance on VQA-CP v2 datasets: Table 4.1 shows results on the VQA-CP generalization task, comparing our results with the state-of-the-art methods.

	Expl.	VQA-CP v2 test			
		All	Yes/No	Num	Other
GVQA[Agrawal et al., 2018]		31.3	58.0	13.7	22.1
UpDn [Anderson et al., 2018]		39.7	42.7	11.9	46.1
UpDn+AttAlign [Selvaraju et al., 2019]		38.5	42.5	11.4	43.8
UpDn+AdvReg. [Ramakrishnan et al., 2018]		41.2	65.5	**15.5**	35.5
UpDn+SCR (ours)	QA	**48.47**	**70.41**	10.42	**47.29**
UpDn+HINT [Selvaraju et al., 2019]	HAT	47.7	70.0	10.7	46.3
UpDn+SCR (ours)	HAT	49.17	71.55	10.72	47.49
UpDn+SCR (ours)	VQA-X	**49.45**	**72.36**	10.93	**48.02**

Table 4.1: Comparison of the results on VQA-CP test with the state-of-the-art systems. The upper part includes VQA systems without human explanations during training, and the VQA systems in the bottom part use either visual or textual human explanations. The "Expl." column shows the source of explanations for training the VQA systems. SCR is the short hand for our self-critical reasoning approach. The results with a precision of 2 decimal points denote the mean of three runs with different random initial seeds.

Our system significantly outperforms other state-of-the-art system (e.g., HINT [Selvaraju et al., 2019]) by 1.5% on the overall score for VQA-CP when using the same human visual explanations (VQA-HAT), which indicates the effectiveness of directly criticizing the competitive answers' sensitivity to the most influential objects. Using human textual explanations as supervision is even a bit more effective. With only about half the number of explanations compared to VQA-HAT, these textual explanations improve VQA performance by an additional 0.3% on the overall score, achieving a new state-of-the-art of 49.5%.

Without human explanations, our approach that only uses the QA proposal object set as supervision clearly outperforms all of the previous approaches, even those that use human explanations. We further analyzed the quality of the influential

object proposal sets extracted from the QA pairs by comparing them to those from the corresponding human explanations. On average, the QA proposal sets contain 57.1% and 54.3% of the objects in the VQA-X and VQA-HAT proposal object sets, respectively, indicating a significant but not perfect overlap.

Note that our self-critical objective particularly improves VQA performance in the 'Yes/No' and 'Other' question categories; however, it does not do as well in the 'Num' category. This is understandable because counting problems are generally harder than the other two types, and requires the VQA system to consider *all* of the objects jointly. Therefore, criticizing only the most sensitive ones does not improve the performance.

Effectiveness of criticizing false sensitivity: In this section, we quantitatively evaluate the effectiveness of the proposed self-critical objective. In particular, we evaluate the fraction of false sensitivity where the predicted incorrect answer's sensitivity to the influential object (to which the correct answer is most sensitive) is greater than the correct answer's sensitivity. We formally define the false sensitivity rate in Eq. 4.6:

$$\mathcal{FSR} = \frac{\sum_{Q,\mathcal{V}} 1[\mathcal{S}(a_{pred}, \mathbf{v}^*) - \mathcal{S}(a_{gt}, \mathbf{v}^*) > 0, score(a_{pred}) = 0]}{\sum_{Q,\mathcal{V}} 1} \tag{4.6}$$

where $1[\cdot]$ denote the function that returns 1 if the condition is satisfied and returns 0 otherwise.

For the original UpDn VQA system, we observe a false sensitivity rate of 35.5% among all the test QA pairs in the VQA-CP. After the self-critical training,

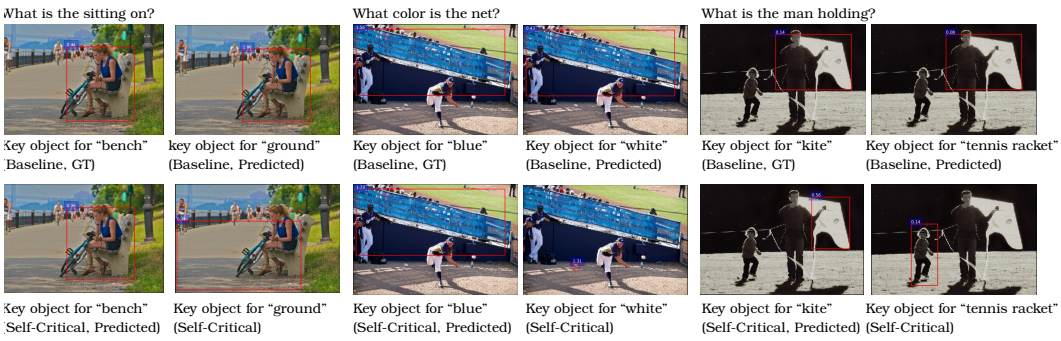


Figure 4.4: Positive examples are showing that our self-critical reasoning approach prevents the incorrectly predicted answer in the UpDn baseline system from being sensitive to the most influential object. For each example, the top two figures show the object to which the ground truth (left) and incorrectly predicted (right) answers are sensitive. The bottom two figures show the corresponding most influential object after our self-critical training. Note that the attention for the incorrect answer shifts to a more relevant part of the image for that answer. The number around the bounding box is the answer's sensitivity to the object.

	UpDn	UpDn + QA	UpDn + HAT	UpDn + VQA-X
$\mathcal{FSR}$	35.5%	22.6%	20.4%	19.6%

Table 4.2: False sensitivity rate ($\mathcal{FSR}$) comparison of using different types of human explanations.

the false sensitivity rate reduces to 20.4% using the VQA-HAT explanations, and to 19.6% using VQA-X explanations. This indicates that false sensitivity is a common problem in VQA systems and shows the utility of addressing it.

Some examples of how our self-critical approach mitigates false sensitivity are shown in Figure 4.4. Note that for the correct answer, our approach increases the influence of the most influential object, which we attribute to the influence strengthening part. More importantly, we observe that this object's influence on the

incorrect answer *decreases* and sometimes falls below other objects.

Ablation study on proposal influential object set size: Table 4.3 reports results with various set sizes indicating the two objectives are fairly robust. We use VQA-HAT visual explanations to construct the influential object sets and both losses to fine-tune our model.

$\lvert\mathcal{I}\rvert$	4	5	6	7	8	10
VQA-CP v2 test	48.8%	49.1%	49.2%	49.1%	48.7%	48.3%

Table 4.3: Ablation study on the size of the proposal influential object set when computing the two losses.

4.5 Chapter Summary

In this chapter, we have explores how to improve VQA performance by criticizing the sensitivity of incorrect answers to the most influential object for the correct answer. Our "self-critical" approach helps VQA systems generalize to test data where the distribution of question-answer pairs is significantly different from the training data. The influential objects are selected from a proposal set extracted from human visual or textual explanations, or simply from the mentioned objects in the questions and answers.

Chapter 5

Generating Multimodal Faithful Explanations

5.1 Motivation and Chapter Overview

Question: What sport is pictured? Answer: Surfing
Explanation: Because the man is riding a wave on a surfboard.

Figure 5.1: Example of our multimodal explanation. It highlights relevant image regions together with a textual explanation with corresponding words in the same color.

Most systems [Fukui et al., 2016, Anderson et al., 2018, Yang et al., 2016, Jiang et al., 2018] based on deep neural networks are difficult to comprehend because of many layers of abstraction and a large number of parameters. This makes it hard

to develop user trust. Partly due to the opacity of current deep models, there has been a recent resurgence of interest in *explainable AI*, systems that can explain their reasoning to human users. In particular, there has been some recent development of explainable VQA systems [Selvaraju et al., 2017, Park et al., 2018, Hendricks et al., 2016, 2018].

One approach to explainable VQA is to generate *visual explanations*, which highlight image regions that most contributed to the system's answer, as determined by attention mechanisms [Lu et al., 2016] or gradient analysis [Selvaraju et al., 2017]. However, such simple visualizations do not explain *how* these regions support the answer. An alternate approach is to generate a *textual explanation*, a natural-language sentence that provides reasons for the answer. Some recent work has generated textual explanations for VQA by training a recurrent neural network (RNN) to directly mimic examples of human explanations [Hendricks et al., 2016, Park et al., 2018]. A *multimodal* approach that integrates *both* a visual and textual explanation provides the advantages of both. Words and phrases in the text can point to relevant regions in the image. An illustrative explanation generated by our system is shown in Figure. 5.1.

Recent research on such multimodal VQA explanation is presented in [Park et al., 2018] that employs a form of "post hoc justification" that does not truly follow and reflect the system's actual processing. As suggested in [Bilgic and Mooney, 2005], we believe that explanations should more faithfully reflect the actual processing of the underlying system in order to provide users with a deeper understanding of the system, increasing trust for the right reasons, rather than trying

to simply convince them of the system's reliability. In order to be faithful, the textual explanation generator should focus on the set of objects that contribute to the predicted answers, and receive proper supervision from only the gold standard explanations that are consistent with the actual VQA reasoning process. Towards this end, our explanation module directly uses the VQA-attended features and is trained to only generate human explanations that can be traced back to the relevant object set using a gradient-based method called GradCAM [Selvaraju et al., 2017].

This chapter is based on [Wu et al., 2019]. In the following sections, we present the design and evaluation of explanation generation model.

5.2 Explanation Generation Model

Our goal is to generate more faithful multimodal explanations that specifically include the segmented objects in the image that are the focus of the VQA module. Figure 5.2 illustrates our model's pipeline in the training phase, consisting of the VQA module , and textual explanation module. We first segment the objects in the image and predict the answer using the VQA module, which has an attention mechanism over those objects. Next, the explanation module is trained to generate textual explanations conditioned on the question, answer, and VQA-attended features. To faithfully train the explanation module, we filter out human textual explanations whose gradient-based visual explanation is not consistent with that of the predicted answer. For example, in Figure 5.2 "Explanation 1" is accepted as the textual explanation since it is mainly focused on the surfer and "Explanation 2" is rejected.

As suggested in [Park et al., 2018], we encode questions and answers as

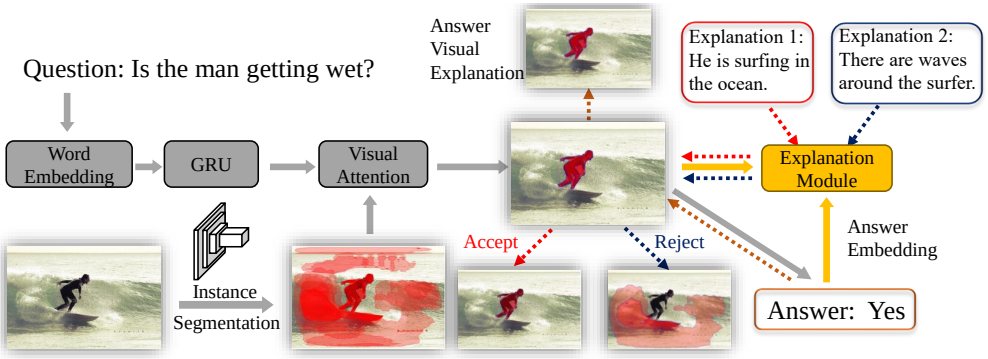


Figure 5.2: Model overview: We first segment the image and then predict the answer for the visual question with a pretrained VQA module. Then, we learn to embed the question, answer, and the VQA-attended features to generate textual explanations. During training, we only use the faithful human explanation whose gradient-based visual explanation is consistent with that of the predicted answer. In the example, our explanation module is only trained to generate "Explanation 1" and further enforces the consistency between this explanation and the predicted answer. "Explanation 2" is filtered out since its visual explanation is mainly focused on the waves and is not consistent with VQA module's focus on the surfer. Dashed arrows denote gradients, gray and yellow arrows denote fixed and trainable parameters, respectively. The three smaller images denote the gradient-based visual explanations for the predicted answer and the two textual explanations.

input features to the explanation module. In particular, we regard the normalized answer prediction output as a multinomial distribution, and sample one answer from this distribution at each time step. We re-embed it as a one-hot vector $\mathbf{a}_s = \text{one-hot}(\text{multinomial}(s))$.

$$\mathbf{u}_i = \mathbf{v}_i^q \odot f(\mathbf{a}_s) \odot f(\mathbf{q}) \tag{5.1}$$

Next, we element-wise multiply the embedding of $\mathbf{q}$ and $\mathbf{a}_s$ with $\mathbf{v}_i^q$ to compute the joint representation $\mathbf{u}_i$. Note that $\mathbf{u}$ faithfully represents the focus of

the VQA process, in that it is directly derived from the VQA-attended features. We use f to denote the fully-connected fc layers of the neural network, and these fc layers do not share parameters. We notate the sigmoid functions as σ. The subscript i indexes the elements of the segmented object sets from images. Bold letters denote vectors, overlining $\bar{\cdot}$ denotes averaging, and $[\cdot, \cdot]$ denotes concatenation.

The explanation module has a two-layer-LSTM architecture whose first layer produces an attention over the $\mathbf{u}_i$, and whose second layer learns a representation for predicting the next word using the first layer's features.

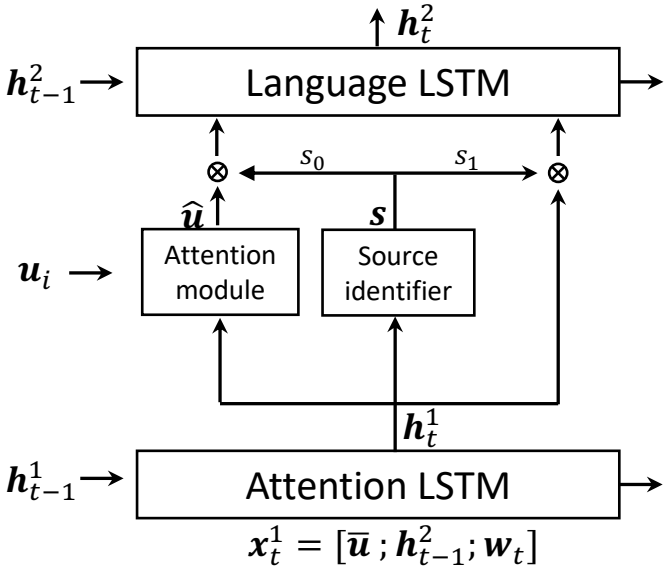


Figure 5.3: Overview of the explanation module that has a two-layer-LSTM architecture.

In particular, the first visual attention LSTM takes the concatenation $\mathbf{x}_t^1$ of the second language LSTM's previous output $\mathbf{h}_{t-1}^2$, the average pooling of $\mathbf{u}_i$, and the previous words' embedding as input and produces the hidden presentation $\mathbf{h}_t^1$.

Then, an attention mechanism re-weights the image feature $\mathbf{u}_i$ using the generated $\mathbf{h}_t^1$ as input shown in Eq. 5.2. Please refer to [Anderson et al., 2018] for the detailed structure.

$$a_{i,t} = f(tanh(f(\mathbf{u}_i) + f(\mathbf{h}_t^1))) \tag{5.2}$$

$$\boldsymbol{\alpha}_t = softmax(\boldsymbol{\alpha}_t) \tag{5.3}$$

For the purpose of faithfully grounding the generated explanation in the image, we argue that the generator should be able to determine if the next word should be based on image content attended to by the VQA system or on learned linguistic content. To achieve this, we introduce a "source identifier" to balance the total amount of attention paid to the visual features $\mathbf{u}_i$ and the recurrent hidden representation $\mathbf{h}_t^1$ at each time step. In particular, given the output $\mathbf{h}_t^1$ from the attention LSTM and the average pooling $\overline{\mathbf{u}}_i$ over $\mathbf{u}_i$, we train a fc layer to produce a 2-d output $\mathbf{s} = \sigma(f([\mathbf{h}_t^1,\ \overline{\mathbf{u}}_i])) = (s_0, s_1)$ that identifies which source the current generated word should be based on ($i.e.$ s_0 for the output of the attention LSTM[1] and s_1 for the attended image features).

$$\mathbf{s} = \sigma(f([\mathbf{h}_t^1,\ \overline{\mathbf{u}}_i])) \tag{5.4}$$

We use the following approach to obtain training labels $\hat{\mathbf{s}}$ for the source identifier. For each visual features $\mathbf{u}_i$, we assign label 1 (indicating the use of attended

[1]We tried to directly use the source weights s_0 in the language LSTM's hidden representation $\mathbf{h}_{t-1}^2$ and found that using $\mathbf{h}_t^1$ works better. The reason is that directly constraining $\mathbf{h}_{t-1}^2$ makes the language LSTM forget the previously encoded content and prevents it from learning long term dependencies.

visual information) when there exists a segmentation $\mathbf{u}_i$ whose cosine similarity between its category name's GloVe representation and the current generated word's GloVe representation is above 0.6. Given the labeled data, we train the source identifier using cross entropy loss $\mathcal{L}_s$ as shown in Eq. 5.5:

$$\mathcal{L}_s = -(\sum_{j=0}^{1} \hat{s}_j \log s_j + (1 - \hat{s}_j) \log(1 - s_j)) \tag{5.5}$$

where the $\hat{s}_0, \hat{s}_1$ are the aforementioned labels.

With the hidden states $\mathbf{h}_t^2$ in the Language LSTM, the output word's probability is computed using Eq. 5.6:

$$p(y_t|y_{1:t-1}) = softmax(f(\mathbf{h}_t^2)) \tag{5.6}$$

where y_t denotes the t-th word in the explanation $\mathbf{y}$ and $y_{1:t-1}$ denotes the first $t-1$ words.

Faithful explanation supervision. Directly collecting faithful textual explanations is infeasible because it would require an annotation process where workers provide explanations based on the attended VQA features. Instead, we design an online algorithm that automatically filters unfaithful explanations from the human ones in the VQA-X data [Park et al., 2018] based on the idea that a proper explanation should focus on the same set of objects as the VQA module and be locally faithful. As recent research suggested that gradient-based methods more faithfully present the models' decision making process [Zhang et al., Wu et al., 2018, Jain and Wallace, 2019], we define a faithfulness score $\mathcal{S}_f$ as the cosine similarity between the Grad-CAM [Selvaraju et al., 2017] visual explanation vectors of the textual explanation

and the predicted answer as shown in Eq. 5.7:

$$\mathcal{S}_f(\mathbf{y}) = cos(g(s^{vqa}_{pred}, \mathbf{v}^q), g(\log p(\mathbf{y}), \mathbf{v}^q)) \tag{5.7}$$

where g denotes the Grad-CAM operation and the result is a vector of length $|V|$ indicating the contribution of each segmented object. s^{vqa}_{pred} is the logit for the predicted answer.

Then, we filter out the explanations in the training set whose faithfulness scores are less than $\xi \max(0.02\ it, 1)$, where ξ is a threshold and the $\max(0.02\ it, 1)$ term is used to jump-start the randomly initialized explanation module. For example, during training, we only accept "Explanation 1" in Figure 5.2 because the visual explanations of the predicted answer and the textual explanation are consistent and reject "Explanation 2".

Since the VQA-X dataset only has explanations for the correct answers, we also discard the explanations when the predicted answers are wrong. With the remaining human explanations, we minimize the cross-entropy loss $\mathcal{L}_{XE}$ in Eq. 5.8:

$$\mathcal{L}_{XE} = \sum_{t=1}^{T} \log(p(y_t|y_{1:t-1})) \tag{5.8}$$

As a last step, we link words in the generated textual explanation to image segments in order to generate the final multimodal explanation. To determine which words to link, we extract all common nouns whose source identifier weight s_1 in Eq. 5.4 exceeds 0.5. We then link them to the segmented object with the highest attention weight $\boldsymbol{\alpha}_t$ in Eq. 5.2 when that corresponding output word y_t was generated, but

only if this weight is greater than 0.2.[2]

5.3 Experimental Setup and Results

In this section, we present the evaluations of our model on both textual and multimodal aspects. We pre-train the VQA module on the entire VQA v2 training set for 15 epochs using the Adam optimizer [Kingma and Ba, 2015] with a learning rate of 0.001. After that, the parameters in the VQA module are frozen. Our VQA module is capable of achieving 82.9% and 80.3% in the VQA-X val and test split respectively. and 63.5% in the VQA v2 validation set which is comparable to the baseline Up-Down model (63.2%) [Anderson et al., 2018]. Note that VQA performance is not the focus of this work, and our experimental evaluation focuses on the generated explanations. Finally, we train the explanation module using the human explanations in the VQA-X dataset [Park et al., 2018] filtered for faithfulness. VQA-X contains 29,459 question answer pairs and each pair is associated with a human explanation. We train to minimize the joint loss $\mathcal{L}$, and ξ is empirically set to 0.3. We ran the Adam optimizer for 25 epochs with a batch size of 128. The learning rate for training the explanation module is initialized to 5e-4 and decays by a factor of 0.8 every three epochs.

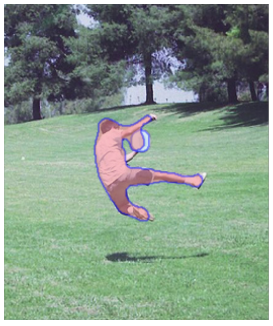
Automated evaluation: Similar to [Park et al., 2018], we first evaluate our textual explanations using automated metrics by comparing them to the gold-standard human explanations in the VQA-X test data using standard sentence-comparison

[2]Due to duplicated segments, we use a lower threshold.

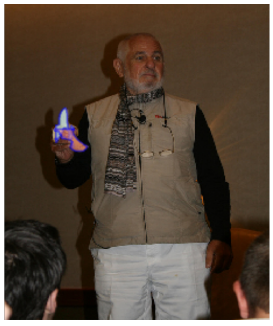
			Textual				
	$\mathcal{L}_s$	$\mathcal{F}$	B-4	M	R-L	C	S
PJ-X [Park et al., 2018]			19.5	18.2	43.7	71.3	15.1
Ours (Justification)	✓		24.1	18.6	46.2	83.4	16.2
Ours (Explanation)	✓	✓	24.7	19.2	47.0	85.1	16.6

Table 5.1: Explanation evaluation results. $\mathcal{F}$ denotes whether to filter out the unfaithful training explanations. $\mathcal{L}_s$ denotes the losses of the source identifier. B-4, M, R-L, C and S are short hand for BLEU-4, METEOR, ROUGE-L, CIDEr and SPICE, respectively.

metrics: BLEU-4 [Papineni et al., 2002], METEOR [Banerjee and Lavie, 2005], ROUGE-L [Lin, 2004], CIDEr [Vedantam et al., 2015] and SPICE [Anderson et al., 2016]. Table 5.1 reports our performance, including ablations.



Question: What sport is shown?
Answer: Frisbee
Explanation: The man is catching a frisbee.



Question: What is he eating?
Answer: Banana
Explanation: He is eating a yellow fruit with a peel.

Question: What sport is this?
Answer: Snowboarding
Explanation: The man is going down a snowy hill on single board.

Figure 5.4: Sample positively-rated explanations. The generated explanations reveal that important objects for answering the visual question in both the visual and the textual modalities.

Human evaluation: We also asked AMT workers to evaluate our final multimodal explanations that link words in the textual explanation directly to segments in the image. Specifically, we randomly selected 1,000 correctly answered question and

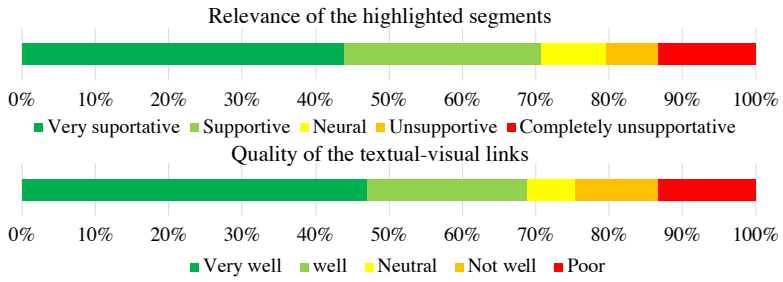


Figure 5.5: Human evaluation results. About 70% of the evaluations are positive and about 45% of them are strongly positive.

asked workers " How well do the highlighted image regions support the answer to the question?" and provided them a Likert-scale set of possible answers: "Very supportive", "Supportive", "Neutral", 'Unsupportive" and "Completely unsupportive". The second task was to evaluate the quality of the links between words and image regions in the explanations. We asked workers "How well do the colored image segments highlight the appropriate regions for the corresponding colored words in the explanation?" with the Like-scale choices: "Very Well", "Well", "Neutral", "Not Well", "Poorly". We assign five questions in each AMT HIT with one "validation" item to control the HIT's qualities.

As shown in Figure 5.5, in both cases, about 70% of the evaluations are positive and about 45% of them are strongly positive. This indicates that our multimodal explanations provide good connections among visual explanations, textual explanations, and the VQA process. Figure 5.4 presents some sample positively-rated multimodal explanations.

5.4 Chapter Summary

This chapter has presented a new approach to generating multimodal explanations for visual question answering systems that aims to more faithfully represent the reasoning of the underlying VQA system while maintaining the style of human explanations. The approach generates textual explanations with words linked to relevant image regions actually attended to by the underlying VQA system. Experimental evaluations of the explanations using both automated metrics and crowdsourced human judgments were presented that demonstrate the advantages of this approach compared to a previously-published competing method.

Chapter 6

Comparing Competing Explanations to Improve VQA

6.1 Motivation and Chapter Overview

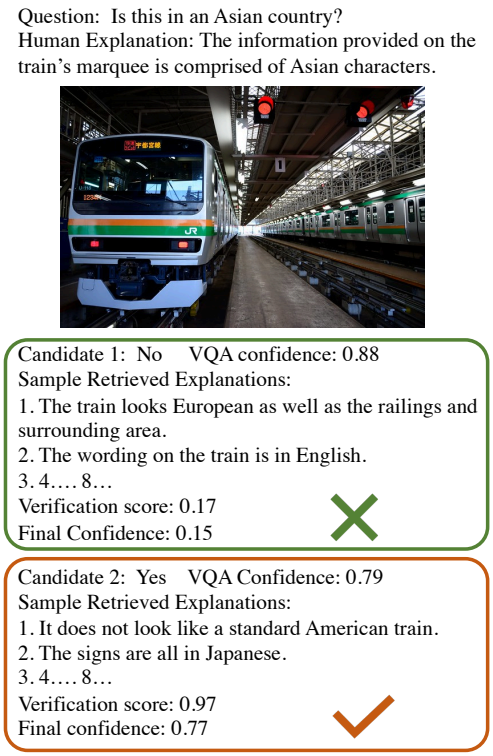


Figure 6.1: An example of utilizing retrieved explanations to correct the original VQA prediction. Though the original VQA confidence of the correct answer "Yes" is lower than that of the incorrect answer "No", the retrieved explanations for "Yes" that states the signs are all in Japanese support their answer better, resulting in a higher verification score and a final correct decision.

Most state-of-the-art VQA systems [Anderson et al., 2018, Kim et al., 2018, Ben-Younes et al., 2017, Jiang et al., 2018, Cadene et al., 2019, Lu et al., 2019, Liu et al., 2019, Tan and Bansal, 2019] are trained to fit the answer distribution using question and visual features and achieve high performance on simple visual questions. However, these systems often exhibit poor explanatory capabilities and take shortcuts by only focusing on simple visual concepts or question priors instead of finding the right answer for the right reasons [Ross et al., 2017, Selvaraju et al., 2019]. This problem becomes increasingly severe when the questions require more complex reasoning and commonsense knowledge.

For more complex questions, VQA systems need to be right for the right reasons in order to generalize well to test problems. Two ways to provide these reasons are to crowdsource human visual explanations [Das et al., 2017] or textual explanations [Park et al., 2018]. While visual explanations only annotate which parts of an image contribute most to the answer, textual explanations encode richer information such as detailed attributes, relationships, or commonsense knowledge that is not necessarily directly found in the image. Therefore, we adopt textual explanations to guide VQA systems.

Recent research utilizing textual explanations adopts a multi-task learning strategy that jointly trains an answer predictor and an explanation generator [Li et al., 2018b, Park et al., 2018]. However, this approach only considers explanations for the one chosen answer. Our approach considers explanations for multiple competing answers, comparing these explanations when choosing a final answer, as shown in Figure 6.1.

This chapter is based on [Wu et al., 2020]. There are a few ways to obtain such explanations, and we present a generation-based approach and a retrieval-based approach in the following section. After that, we present the VQA model that compares these explanations.

6.2 Retrieving Competing Explanations

This section presents our approach to retrieving the most supportive human textual explanation from the training set for each answer candidate. Ideally, we would dynamically retrieve explanations for each answer at each iteration. However, this would be very computational costly because the question and visual features have to be computed for each image from the training set. Therefore, we adopt the below relaxation for computational efficiency that only needs to compute the features once.

In particular, we first pretrain the VQA model, and extract the question and visual embeddings, denoted as **q** and **v**, for each example in the training set. We consider two VQA model in this scetion, i.e. UpDn and LXMERT. For UpDn, we use the attended visual features and the question GRU's last hidden state as the visual and question embeddings. For LXMERT, we use the last cross-modal attention layer's visual and question output as the embeddings.

Then, for each question, we only compute the top-10 answer candidates since the top-10 answers together achieve high recall. After that, for each answer candidate a, we extract explanations from the training set that have the same ground

truth answer [1] as the current candidate. We then sort these explanations by the L2 distance between the explanations' QV embeddings, $\mathbf{q} \odot \mathbf{v}$, and the example's and pick the closest 8 explanations as the competing explanations set denoted as $\mathcal{X}_a$.

6.3 Generating Competing Explanations

Next,the retrieved explanations for similar VQA examples from the training set are used to help generate even better explanations.

We adopt the explainer from Chapter 6, a two-layer LSTM network similar to the UpDn captioner [Anderson et al., 2018], as our baseline.[2] Since the current VQA systems are built upon detected objects, we use them as the visual inputs instead of segmentations.

The baseline explainer first computes a set of question-attended visual features, $\mathcal{U}$, and an average pooled version, $\bar{\mathbf{u}}$. The explainer then uses $\bar{\mathbf{u}}$ and $\mathcal{U}$ together with question and answer embeddings as inputs to produce explanations.

Our approach simply replaces the average pooled question-attended visual features $\bar{\mathbf{u}}$ with the retrieved explanations' features, $\mathbf{x}$. We use a single-layer GRU to encode all of the retrieved explanations for the correct answer, and then max pool the last hidden states among these explanations to compute $\mathbf{x}$. We sample 8 explanations for each answer candidate to construct the generated explanation set.

[1] More specifically, the soft score of the answer candidate in the retrieved explanation's example is over 0.6

[2] We replace segmentation features to detection features.

6.4 Comparing Competing Explanations for VQA

After obtaining textual explanations, we discuss how to use them to learn a verification score for each answer candidate and use it to re-rank these answer candidates. Precisely, as shown in Figure 6.2, after the base VQA system computes the top-k answers, our approach retrieves the most supportive explanations for each answer from the training set to construct the set of competing explanations. Then, these explanations are used to help generate explanations for the current question. Next, we learn to predict verification scores that indicate how well the retrieved or generated explanations support the predictions given the input question and visual content. The final answer is determined by jointly considering the original answer probabilities and these verification scores.

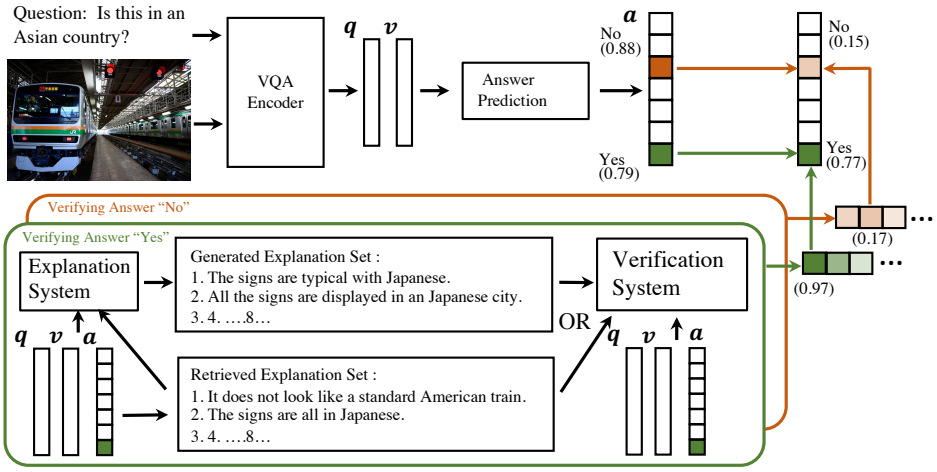


Figure 6.2: Our approach first predicts a set of answer candidates and retrieves explanations for each based on the answer, question, and visual content. These explanations are then used to generate improved explanations. Finally, either retrieved or generated explanations are employed to predict verification scores that are used to reweight the original predictions and compute the final answer.

6.4.1 Learning Verification Scores

A verification system is trained to score how well a generated or retrieved explanation supports a corresponding answer candidate given the question and visual content. The verification system takes four inputs: the visual, question, answer and its explanation features; and outputs the verification score, $i.e.$ $S(Q, \mathcal{V}, a, x) = \sigma(f_2(f(\mathbf{q}), f(\mathbf{v}), f(\mathbf{a}), f(\phi(x)))$. where **a** is the one-hot embedding of the answer, and $\phi(x)$ is the feature vector for the explanation, x, encoded using a GRU [Cho et al., 2014], ϕ. We use f_n to denote n consecutive feed-forward layers (for simplicity n is omitted when $n = 1$). We use σ to denote the sigmoid function. The verification system is similar to the answer predictor in architecture except for the number of outputs.

Given the VQA examples with their explanations in the VQA-X dataset, we use binary cross-entropy loss $\mathcal{L}_m$ to maximize the verification score for the matching human explanations, $i.e.$ $\mathcal{L}_m = -\log(S(Q, \mathcal{V}, a, x))$.

Intuitively, we want the verification score S to be high only when the explanation is matched to the VQA example, $i.e.$ replacement of any of the four input sources should lower the score. Therefore, we designed the five kinds of replacements below for constructing negative examples.

Replacement of visual and question features: Ideally, we would replace the visual and question features with the complementary features [Antol et al., 2015] that lead to the opposite answer. For example, for the question "Is this a vegetarian pizza?", with an image of a vegetarian pizza, we should replace the image with one of a meat

pizza, $i.e.$ counter-factual images. However, such replacement requires retrieving and computing the visual features **v** for the meat pizza, which is computationally inefficient. Therefore, we simply randomly choose a Q' or $\mathcal{V}'$ replacement from the current batch and minimize the binary cross entropy loss $\mathcal{L}_r^q$, $\mathcal{L}_r^q$ for the verification scores, $i.e.$ $\mathcal{L}_r^q = -\log(1 - S(Q', \mathcal{V}, a, x))$, $\mathcal{L}_r^v = -\log(1 - S(Q, \mathcal{V}', a, x))$.

Replacement of answer features: We sample the answer for replacement according to the current VQA's predicted incorrect probabilities. At each step, we try to minimize the expectated binary cross-entropy loss $\mathcal{L}_r^a$ for the incorrect predictions, $i.e.$ $\mathcal{L}_r^a = \mathbb{E}_{a' \sim p(a'|QV),\ s(a')<0.6}[-\log(1 - S(Q, \mathcal{V}, a', x))]$ where $s(a')$ denotes the human VQA soft score for answer a'. In practice, we only sample one incorrect answer during training.

Replacement of explanation features: We try to replace the matched human explanations with the most supportive explanations for the sampled incorrect answer and train our verification system to disprefer that explanation using the loss $\mathcal{L}_r^a = \max_{x' \in \mathcal{X}_{a'}}[-\log(1 - S(Q, \mathcal{V}, a, x'))]$. In particular, given the sampled incorrect answer a' from the previous section, we compute the verification score for each retrieved or generated explanation x' from the set $\mathcal{X}_{a'}$ for that wrong answer and regard the one with maximum verification score as the most supportive one.

Replacement of answer and explanation features: To further prevent the system from being falsely confident in the sampled incorrect answer a', we also minimize the verification score for its most supportive explanation for the incorrect answer a', $i.e.$ $\mathcal{L}_r^{ax} = -\log(\max_{x' \in \mathcal{X}_{a'}}(1 - S(Q, \mathcal{V}, a', x')))$.

Finally, the total verification loss is the sum of the aforementioned 6 losses as shown in Eq. 6.1:

$$\mathcal{L}_{verification} = \lambda \mathcal{L}_m + \mathcal{L}_r^q + \mathcal{L}_r^v + \mathcal{L}_r^a + \mathcal{L}_r^x + \mathcal{L}_r^{ax} \tag{6.1}$$

Since we have more negative examples (5 ways to form negative examples) and only one positive example, we assign a larger loss weight ($i.e.\ \lambda = 10$) for the only positive example.

6.4.2 Using Verification Scores

The original VQA system provides the answer probabilities conditioned on the question and visual content, $i.e.\ P(a|Q, \mathcal{V})$. The verification scores $S(Q, \mathcal{V}, a, x)$ are further used to reweight the original VQA predictions so that the final predictions $\tilde{P}(a|Q, \mathcal{V})$, shown in Eq. 6.2, can take the explanations into account.

$$\tilde{P}(a|Q, \mathcal{V}) = P(a|Q, \mathcal{V}) \max_{x \in \mathcal{X}_a} S(Q, \mathcal{V}, a, x) \tag{6.2}$$

where $\mathcal{X}_a$ denotes the generated or retrieved explanation set for the answer a.

Since we try to select the correct answer with its explanation, the prediction $\tilde{P}(a|Q, \mathcal{V})$ should only be high when the answer a is correct and the explanation x supports a, which is enforced using the loss in Eq. 6.3:

$$\mathcal{L}_{vqae} = -\log(P(a|Q, \mathcal{V})S(Q, \mathcal{V}, a, x_a)) - \log(1 - \tilde{P}(a'|Q, \mathcal{V})) \tag{6.3}$$

where x_a denotes the human explanation for the answer a.

During testing, we first extract the top 10 answer candidates $\mathcal{A}$, and then select the explanation for the answer candidate with the highest verification score.

Then, we compute the explanation-reweighted score for each answer candidate to determine the final answer $a^\star = \arg\max_{a \in \mathcal{A}} \tilde{P}(a|Q, \mathcal{V})$.

6.5 Experimental Setup and Results

Training Details: We first pre-train our base VQA system (UpDn or LXMERT) on either VQA v2 training set for 20 epochs or only the VQA-X training set for 30 epochs with the standard VQA loss (binary cross-entropy loss with soft scores as supervision) and the Adam optimizer [Kingma and Ba, 2015]. As the VQA-X validation and test set are both from the VQA v2 validation set that is covered in the LXMERT pretraining, we do not use the officially released LXMERT parameters. The learning rate is fixed to 5e-4 for UpDn and 5e-5 for LXMERT, with a batch size of 384 during the pre-training process. For answer prediction part, we use $1,280$ hidden units in UpDn and 768 hidden units in LXMERT, and for verification part, we use $1,280$ hidden units in both systems. We fine-tune our system using the verification loss and VQA loss $\mathcal{L}_{verification} + 0.1\mathcal{L}_{vqae}$ on the VQA-X training set for another 40 epochs. For the verification systems, the initial learning rate is set to 5e-4. The learning rate for every parameter is decayed by 0.8 every 5 epochs. During test, we consider the top-10 answer candidates for the VQA systems and use the reweighted prediction as the final answer.

Implementation: We implemented our approach on top of the original UpDn and LXMERT. Both base systems utilize a Faster R-CNN head [Ren et al., 2015b] in conjunction with a ResNet-101 base network [He et al., 2016] as the object detection module. Convolutional features for the top 36 objects are then extracted for each

	VQA-X Pretrain		VQA v2 Pretrain	
	Gen. Expl.	Ret. Expl.	Gen. Expl.	Ret. Expl.
UpDn [Anderson et al., 2018]	74.2	74.2	83.6	83.6
UpDn+E (ours)	78.0	78.7	85.1	85.4
LXMERT [Tan and Bansal, 2019]	76.8	76.8	83.7	83.7
LXMERT+E (ours)	77.3	78.0	84.1	84.7

Table 6.1: Question answering accuracy on VQA-X using both UpDn and LXMERT as a base system,"+E" denotes using our competing explanations approach. "Gen. Expl." and "Ret. Expl." denote using generated and retrieved explanations, respectively.

image as the visual features, $i.e.$ a $2,048$ dimensional vector for each object. For question embedding, following [Anderson et al., 2018], we perform standard text pre-processing and tokenization for UpDn. In particular, questions are first converted to lower case, trimmed to a maximum of 14 words, and tokenized by white spaces. A single layer GRU [Cho et al., 2014] is used to sequentially process the word vectors and produce a sentential representation for the pre-processed question. We also use Glove vectors [Pennington et al., 2014] to initialize the word embedding matrix when embedding the questions. For LXMERT, we also follow the original BERT word-level sentence embedding strategy that first splits the sentence into words $w_1, ..., w_n$ with length of n by the same WordPiece tokenizer [Wu et al., 2016c] in [Devlin et al., 2019]. Next, the word and its index ($i.e.$ absolute position in the sentence) are projected to vectors by embedding sub-layers, and then added to the index-aware word embeddings. We use a single-layer GRU and three-layer GRU to encode the generated or retrieved explanation in the verification system when using UpDn and LXMERT as base system, respectively.

VQA main results: We present the VQA results on the VQA-X [Park et al., 2018] dataset. We combine the validation set (1,459 examples) and test set (1,968 examples) of the VQA-X dataset as our larger test set (3,427 examples) for more stable results since both are relatively small. We compare our system's VQA performance against two corresponding base systems using the standard protocol. In addition, we examine the quality of explanations by comparing our system against a baseline model as well as human explanations. Finally, we perform ablation studies to show that both improved feature representation and explanation reweighting are key aspects of the improvements.

Table 6.1 reports the results of our competing explanation approach. Our approach combined with UpDn pretrained on the entire VQA v2 dataset achieves the best results. When training only on the VQA-X training set, we improve the original UpDn and LXMERT by 4.5 % and 1.2 %, respectively. UpDn benefits more from using competing explanations than LXMERT, but both improve. By using transformers, LXMERT already creates better, but less flexible, representations which are harder to improve upon by using explanations. Because we do not use the official LXMERT model parameters pretrained on multiple large datasets (VQA-X test set is used as training set for the official released model) and only train the LXMERT on VQA v2 dataset, the performance of LXMERT is not better than UpDn.

Effect of using different explanations: Table 6.2 reports overall VQA scores using UpDn pretrained on the VQA-X train set. We include two baseline settings, "UpD" and "UpDn + VQA-E", where the model is trained to jointly predict the answer and

generate the explanation, using a two-layer attentional LSTM on top of the VQA shared features. This version models the approach used in [Li et al., 2018b].

In the first human explanation setting, denoted by $\mathcal{RR}$, we only $\mathcal{R}$eplace the retrieved explanation for the $\mathcal{R}$ight answer with the corresponding human ones, and still use the retrieved explanation for the incorrect answer. This setting shows how much retrieved explanations for correct answers impacts the results. The second human explanation setting, denoted by $\mathcal{RA}$, assumes that human explanations are used to $\mathcal{R}$eplace the retrieved explanations for $\mathcal{A}$ll the potential answer candidates. This setting provides an upper bound on our approach that uses textual explanations.

	VQA-X
UpDn [Anderson et al., 2018]	74.2
UpDn + VQA-E [Li et al., 2018b]	76.0
UpDn + generated explanations	78.0
UpDn + retrieved explanations	78.7
UpDn + human explanations ($\mathcal{RR}$)	79.3
UpDn + human explanations ($\mathcal{RA}$)	80.2

Table 6.2: VQA performance using different explanations. "VQA-E" denotes model trained to jointly predict the answer and generate the explanation.

The results indicate that using explanations even in a simple multi-task learning model [Li et al., 2018b] is helpful, providing 2% improvement on the overall score. However, our competing explanation approach with either generated or retrieved explanations significantly outperforms both baseline models.

Our system with retrieved explanations performs slightly better than the one with generated explanations. This is probably because there are no guarantees that the generated explanations will support the answers upon which they are conditioned.

Since all the explanation training examples are for correct answers, the explanation generation system tends to support the ground truth answer regardless of the answer candidate it is generated to support. Also, the generated explanations sometimes ignore or hallucinate [Rohrbach et al., 2018] visual content when explaining the answer. Therefore, although ideally, generated explanations could work better than retrieved ones, they are currently less helpful to the VQA performance due to their imperfections.

Not surprisingly, the human oracle explanations help the VQA system more than the retrieved ones. It indicates that our approach could achieve even better performance with more informative explanations, which could be achieved by either developing a better explanation generator, or enlarging the explanation training set from which human explanations are retrieved. Our system's results using retrieved explanations are only 0.6% lower than with human oracle explanations for the correct answers. This indicates that the retrieved explanations (for related questions) are a reasonable approximation to human explanations for the specific question.

Ablation study on each kind of replacement: Table 6.3 reports the results of ablating each one kind of replacement for raising negative examples during training the verification model. We use the UpDn model pretrained on VQA-X dataset. It verifies the value of each term in Eq. 6.1.

	Full	$w/o\ q$	$w/o\ v$	$w/o\ a$	$w/o\ x$	$w/o\ ax$
UpDn+Re.Expl. (ours)	78.7	77.2	77.4	77.9	78.4	78.3

Table 6.3: Ablation study on each kind of replacements when learning the verification score.

Evaluating representation improvement: This section presents an ablation investigating how our approach improves the learned representations. The "w/o reweighting" ablation still uses the fine-tuned representation trained using explanations, but it does not reweight the final predictions, therefore it tests the improvement solely due to better joint representations for the question and the visual content. The "fixed VQA" ablation uses reweighting, but does not fine-tune the VQA parameters during verification-score training ($i.e.$ only the verification parameters are trained).

	VQA-X
UpDn [Anderson et al., 2018]	74.2
UpDn + VQA-E [Li et al., 2018b]	76.0
UpDn+E (w/o reweighting)	77.8
UpDn+E (fixed VQA)	75.3
UpDn+E	78.7

Table 6.4: Ablation studies on representation improvements. We use the retrieved explanations.

Table 6.4 reports the results of the UpDn system pretrained on the VQA-X dataset. Using explanations as additional supervision helps the VQA systems build better representations for the question and answer, improving performance by 3.6%. This is because minimizing the verification loss $\mathcal{L}_{verification}$ prevents the VQA system from taking shortcuts. First, the $\mathcal{L}_m$ component forces the VQA system to produce visual and question features whose mapping can match the explanation features. Second, by minimizing $\mathcal{L}_r^v$ and $\mathcal{L}_r^v$, the system is forced not to solely focus on question and/or visual priors. Finally, our full system gains 1.1% improvement due to reweighting, and achieves our best results.

More qualitative examples: Figure 7.6 shows some qualitative examples with sampled explanations. The left column shows examples using generated explanations and the right column shows examples using retrieved explanations. For each example, we present the predictions of the original VQA system in the first line, and final predictions using competing explanations in the second line. The numbers in the parentheses are the ground-truth scores for the given answer, and the three scores after the parentheses are the original VQA prediction score, the verification score, and the final prediction score, respectively.

Figure 6.3: More qualitative examples with sample explanations. The first three rows show positive examples and the last row presents two failure cases.

The first three rows show examples where the verification system improves the performance. These examples show that the explanations help the VQA system clarify some commonsense knowledge (e.g. people wear gloves when cold, beer

is amber colored, etc.) and key features of the important objects for the visual questions, leading to better predictions.

The last row shows examples where the verification system actully degrades the original VQA systems' performance. However, the explanations for the incorrect answers are still reasonable and a better verification module should be able to further improve performance on examples like these.

6.6 Chapter Summary

In this work, we have explored how to exploiting explanations to improve VQA performance. We first present two sets of competing explanations, generated and retrieved explanations. Then, the VQA model exploiting the explanation supporting each answer candidate to learn a verification score to re-rank the answer confidence. Our approach also helps the system learn better visual and question representations. As a result, the VQA models avoid taking shortcuts and is able to handle difficult visual questions better, improving results on the challenging VQA-X dataset.

Chapter 7

Utilizing External Knowledge for OK-VQA

7.1 Motivation and Chapter Overview

In previous chapters, we discuss the several challenges in VQA. Despite challenging, a common assumption hold true that the required information could be found in the image. In this chapter, we will discuss a recent trend towards Outside Knowledge VQA (OK-VQA) [Wang et al., 2017, 2018, Marino et al., 2019] which requires information beyond the content of the images. Besides visual recognition, the model needs to perform logical reasoning and incorporate external knowledge about the world to answer these challenging questions correctly. These knowledge facts can be obtained from various sources, such as image search engines, encyclopedia articles, and knowledge bases about common concepts and their relations.

Figure 7.1 illustrates a few visual questions and the knowledge from different external sources that helps answer them. Each question needs a different type of external knowledge. For example, to identify the movie that featured a man telling his life story to strangers, we need to link the image content and question to some textual facts; Vegetarian food and eating vegetables are related to the concept of health; the retrieved images for a "golden retriever." are visually similar to the dog in the question image. *The challenge is to retrieve* and *correctly incorporate such*

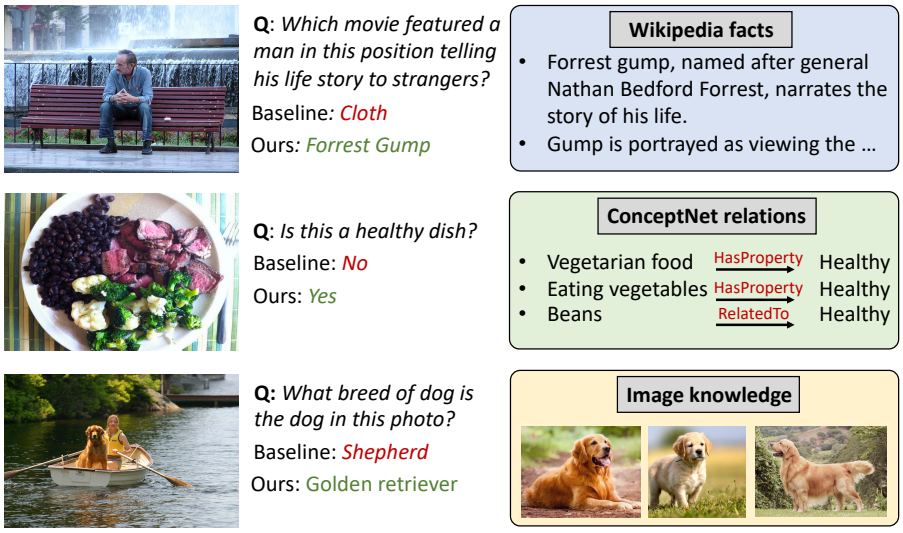


Figure 7.1: We address the problem of knowledge-based question answering. Retrieving relevant knowledge among diverse knowledge sources (visual knowledge, textual facts, concepts, etc.) is quite challenging because the broad scope of the visual questions.

external knowledge effectively in an open domain question answering framework. Following these two threads, we first present a multimodal answer validation framework (MAVEx) that utilizes retrieved knowledge from multiple sources to predict the mostly supported answer. Then, we explore a entity-focused retrieval (EnFoRe) model that retrieves knowledge specifically focused on critical entities.

7.2 Multi-Modal Answer Validation (MAVEx)

Most current knowledge-based VQA systems [Marino et al., 2019, Wang et al., 2018, Zhu et al., 2020, Marino et al., 2021] follow a two-stage framework, where a retriever first looks up knowledge relevant to the question and the image, and then a separate comprehension model predicts the answer. However, knowledge

retrieved directly for the question and image is often noisy and not helpful in predicting the correct answer. For example, as shown in Figure 7.2, the sentences retrieved using only the words in questions and objects in images (top) or a wrong answer (middle) are hardly helpful to answer the question. This increases the burden on the answer predictor, leading to only marginal improvements from the use of retrieved knowledge [Marino et al., 2019]. Interestingly, with the correct answer "Wimbledon" (bottom), the quality of the retrieved fact is significantly improved, making it suitable for answering the question. This observation motivates us to use retrieved knowledge for *answer validation* rather than for producing the answer.

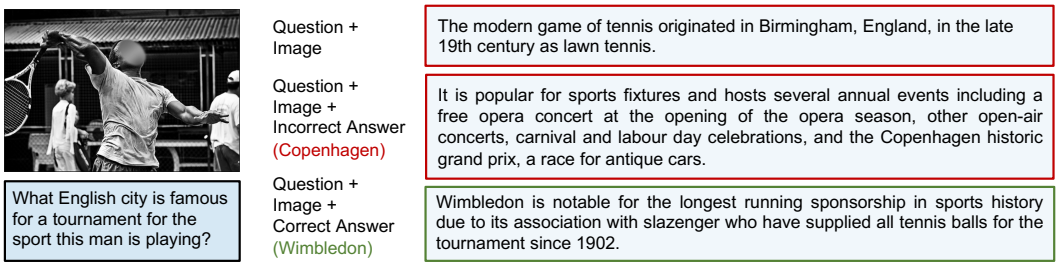


Figure 7.2: Examples of retrieved Wikipedia sentences using different sets of search words. The sentences retrieved using only the words in questions and objects in images (top) and the wrong answer (middle) are hardly helpful to answer the question. However, with the correct answer "Wimbledon" (bottom), the quality of the retrieved fact is significantly improved.

To address this challenge, we propose a new system called MAVEx or **M**ulti-modal **A**nswer **V**alidation using **Ex**ternal knowledge. We use a three-stage framework that first generates a set of promising answer candidates, retrieves knowledge guided by these answer candidates, and finally validates these answer candidates.

7.2.1 Answer Candidate Generation

In order to use answer candidates to inform knowledge retrieval, we use ViLBERT-multi-task system [Lu et al., 2019], a state-of-the-art VQA model, to generate answer candidates. In particular, we finetune a ViLBERT-multi-task model on the OK-VQA dataset that outputs a score for each answer collected from the training set. The highest-scoring answers are used as the candidates. Note that any VQA model or other approaches (for example, querying ontology knowledge bases) can be used for this purpose. However, as we will discuss in the experiments section, we found ViLBERT to be particularly effective at generating a small set of promising candidates.

7.2.2 Rule-based Answer Guided Knowledge Retrieval

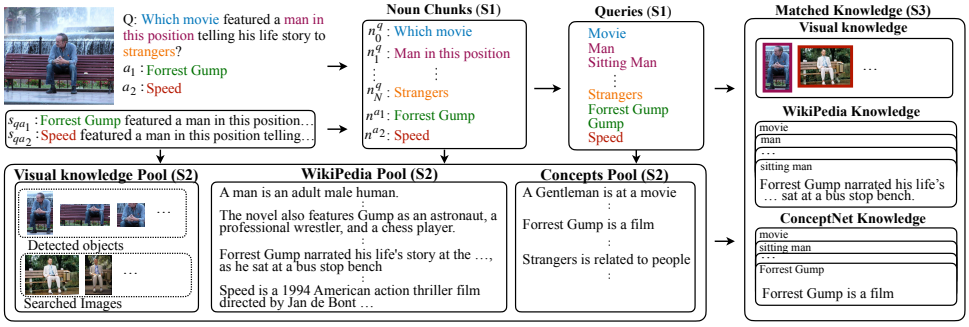


Figure 7.3: An example of the retrieval process for one question-answer pair. The numbers in parentheses denote the step number in Section 7.2.2. The noun phrase, its generated queries, and the matched visual knowledge are marked in the same color.

Given a question q about an image I and a set of answer candidates A, we retrieve external knowledge supporting A in three main steps. Figure 7.3 shows the

entire process for an example question and a candidate answer.

S1: Query extraction: We first collect short phrases in q, each answer candidate in A, and concepts represented in I as a starting point for retrieving external information. This involves the following sub-steps:

Extract noun phrases from question and answers: We parse the question and the candidate answers using a constituency parser to obtain the parse tree. Then, we extract all the nouns on the leaves of the parse tree together with the words that describe the nouns and belong to one of the types from `ADJP`, `ADVP`, `PP`, `SBAR`, `DT` or `JJ`. We extract three kinds of noun phrases for modeling: (1) The target noun phrase that contains 'wh' or 'how' word (e.g. *'which movie'*), denoted by n_0^q. (2) Question noun phrases from the rest of the question (e.g., *'man in this position'*), denoted by $n_i^q, i \in \{1, \ldots, N\}$. N is the number of noun phrases in the question. (3) Answer noun phrase for each answer a_i, denoted by n^{a_i} (e.g., *'Forrest Gump'*). These nouns help us link the mentioned objects to the images.

Link phrases to objects: As images usually contain plenty of question-irrelevant content, making the retrieval process hard, we propose narrowing the search to the objects referred to by the question or the answer candidates. In particular, we use a separate ViLBERT-multi-task [Lu et al., 2020] model as the object linker, where it takes as inputs a set of detected objects and a noun phrase from the question, and outputs a linking score for each detected object to indicate how likely the noun phrase refers to the object. We approve the linking when the score is higher than 0.5 and extract the linked objects.

Generate search query set: We further generate a set of search queries to search the external knowledge base. For each noun phrase, we first extract the head of a phrase by finding the innermost `NP` from the dependency tree. Then, we obtain the visual attributes of the head of the noun phrase by using a pre-trained object-with-attribute detector [Anderson et al., 2018] for the corresponding linked objects. For example, the visually grounded queries for *'man in this position'* are *'man'* and *'sitting man'* where *sitting* is inferred from visual attributes. We denote the set of queries as $r_i^n, i \in \{1, \ldots, K\}$, where n is the corresponding noun phrase, K is the maximum number of queries per noun phrase.

S2: Answer guided knowledge pool construction: We now use the visually grounded queries from step S1 to construct knowledge pool as follows:

Conversion to a natural language statement: In order to use the answer candidate a to inform the retrieval step, we convert q and $a \in A$ into a natural language statement s_{qa} using a T5 model [Raffel et al., 2020] finetuned on the QA-NLI dataset [Demszky et al., 2018]. Such conversion is effective as statements occur much more frequently than questions in textual knowledge sources [Khot et al., 2017]. These statements are later used to compute the relevance of the retrieved facts as described below.

Retrieval of textual facts and concepts: We search each query in the query set generated from the last sub-step in **S1** in Wikipedia and ConceptNet. We compute the BERTScore [Zhang et al., 2020a] between each sentence from the retrieved article and each statement s_{qa}. For each statement, the top-15 sentences (according to the BERTScore) from each retrieved article are pushed to the sentence pool. Then, we

decontextualize [Choi et al., 2021] each sentence in the Wikipedia pool for better knowledge quality.

Retrieval of visual knowledge: Pure textual knowledge is often insufficient due to two main reasons: (1) textual knowledge might be too generic and not specific to the question image, (2) it might be hard to describe some concepts using text, and an image might be more informative (e.g., the third question in Figure 7.1). Hence, visual knowledge can complement textual information, further enriching the external knowledge space. We consider both internal and external visual knowledge. For the given image, we utilize a MaskRCNN [He et al., 2017] object detector to detect common objects as internal knowledge. We use Google image search to retrieve the top-5 images using the statement s_{qa} as the query for each answer candidate a as the external visual knowledge.

S3: Matching knowledge pool to queries: Instead of simply using each query's top retrieved sentences as the query's knowledge, we propose matching the sentences from the entire pool to each query. The intuition is that most queries cannot directly retrieve helpful facts; however, they can help retrieve important aspects that should be contained in the external knowledge.

Matching Textual Knowledge: For each query, the sentences from both Wikipedia and ConceptNet pool with a mean recall greater than 0.6 are considered the retrieved results. Mean recall is the average cosine similarity between the Glove embedding of the words in the query and their most similar word in the sentence. To ensure knowledge relevance, we remove sentences that are matched to only a single query. For each query r_i^n, according to the maximum BERTScore

between the sentence and all of the statements S_q, we extract at most m sentences from both Wikipedia and ConceptNet pools, denoted by $W(r_i^n)$ and $C(r_i^n)$.

Matching Visual Knowledge: For each noun phrase in the question, we can directly use the results from the object linker defined in S1. Specifically, we find the top-3 referenced objects in the image for each question noun phrase, denoted $M(n)$.

For each answer noun phrase n^{a_i}, we use Google image search to retrieve the top-5 images, denoted $M(n^{a_i})$.

7.2.3 Answer Candidate Validation

The answer validation module takes as input an answer candidate a_i and the supporting knowledge, and outputs a scalar score indicating how well the knowledge supports a_i. As we will discuss, in order to better aggregate the knowledge, we first compute the knowledge embedding for each query. Then, we compute an embedding for each noun phrase that aggregates the embedding for the queries generated from the noun phrase[1]. Finally, the embedding for the entire question aggregates the embedding computed for all noun phrases.

We build MAVEx on top of the ViLBERT system. Given a question q and an image I, ViLBERT provides textual features $\boldsymbol{U} \in \mathbb{R}^{|q| \times d}$, visual features $\boldsymbol{V} \in \mathbb{R}^{|V| \times d}$ from the last layer, where $|q|$ is the number of tokens in q, d is the feature dimension, $|V|$ is the number of objects in the image plus one for the representation for the entire image, and a joint visual-textual representation $\boldsymbol{z} \in \mathbb{R}^d$. For each sentence

[1]Recall that our queries r_i^n are created based on noun phrase n.

in the retrieved textual knowledge $W(r_i^n)$ and $C(r_i^n)$, we use TinyBERT (T-BERT) model [Turc et al., 2019] to extract the corresponding features. We further average the sentence features for each query r_i^n, resulting $\boldsymbol{w}_i^n$ and $\boldsymbol{c}_i^n$.

For each image in the retrieved visual knowledge $M(n_a)$, we use MaskRCNN [He et al., 2017] to extract object features. Then, we average the object features of visual detection results as the image features and denote them as $\boldsymbol{m}_i^n$. Note that we directly use the object features for the linked objects. Figure 7.4 shows the overview of the model.

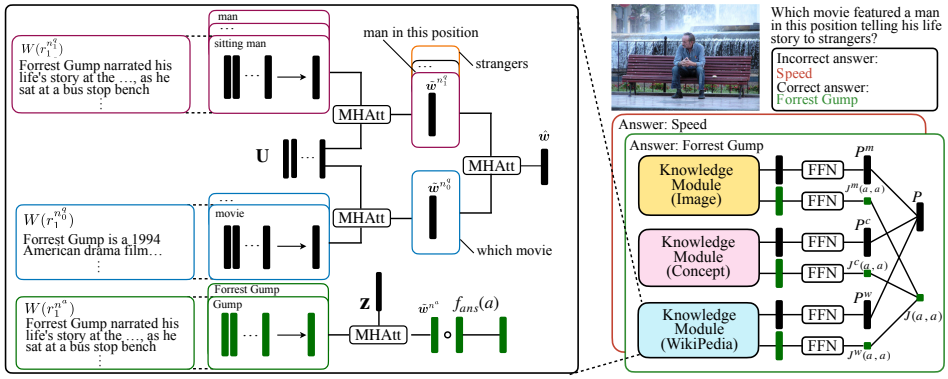


Figure 7.4: Model overview for validating two candidate answers. We explore three sources of external knowledge, $i.e.$ Wikipedia, ConceptNet, and Google Images presented by the three parallel knowledge embedding modules. The black blocks denote features shared by all answer candidates, and the green blocks denote answer-specific features. Different colors denotes the features for different noun phrases and their queries.

Multi-granular knowledge embedding module: In order to better aggregate the retrieved knowledge, we employ a multi-granular knowledge embedding module that learns to recognize the critical queries for each noun phrase, and then the critical

noun phrases for answering the question.

Note that our knowledge embedding module is identical for each knowledge source but with different learnable parameters. We only show knowledge from Wikipedia for brevity. Given the knowledge embeddings $\boldsymbol{w}_i^n$ for each query r_i^n in the question, we compute the knowledge embedding $\tilde{\boldsymbol{w}}^n$ for each noun phrase in question as follows:

$$\tilde{\boldsymbol{w}}^n = \texttt{MHAtt}(\boldsymbol{u}^n, \{\boldsymbol{w}_i^n\}_{i\in\{1,\ldots,K\}}, \{\boldsymbol{w}_i^n\}_{i\in\{1,\ldots,K\}}), \tag{7.1}$$

where $\texttt{MHAtt}(\texttt{query}, \texttt{key}, \texttt{value})$ is the multi-head attention operator. $\boldsymbol{u}^n$ is the attentive pooled [Lee et al., 2017] ViLBERT features according to the span $\{s, e\}$ of the phrase n. We use $\boldsymbol{u}^n$ as the `query` in `MHAtt` module to aggregate the retrieved knowledge, where the corresponding knowledge embeddings $\{\boldsymbol{w}_i^n\}_{i\in\{1,\ldots,K\}}$ serve as `key` and `value`.

Similarly, for each answer a[2], we compute the knowledge embedding $\boldsymbol{w}^a$ using a `MHAtt` module over the knowledge features $\boldsymbol{w}_i^{n_a}$ as follows:

$$\boldsymbol{w}^a = \texttt{MHAtt}(\boldsymbol{z}, \{\boldsymbol{w}_i^{n_a}\}_{i\in\{1,\ldots,K\}}, \{\boldsymbol{w}_i^{n_a}\}_{i\in\{1,\ldots,K\}}), \tag{7.2}$$

where the joint visual-textual embeddings $\boldsymbol{z}$ from ViLBERT system is used as the keys.

Then, another `MHAtt` module is used to gather the knowledge from each noun phrase $n \in \{n_1^q, \ldots, n_N^q\}$. Specifically, given the knowledge embedding for

[2]For simplicity we omit the subscript index of the answer in this section when there is only one answer involved in the current step.

each noun phrase, the knowledge embeddings $\boldsymbol{w}$ is computed as follows:

$$\hat{\boldsymbol{w}} = \texttt{MHAtt}(\tilde{\boldsymbol{w}}^{n_0^q}, \{\tilde{\boldsymbol{w}}^{n_i^q}\}_{i\in 1,\ldots,N}, \{\tilde{\boldsymbol{w}}^{n_i^q}\}_{i\in 1,\ldots,N}) \tag{7.3}$$

Answer prediction and validation module: Given the knowledge embedding $\boldsymbol{k} \in \{\hat{\boldsymbol{w}},\ \hat{\boldsymbol{c}},\ \hat{\boldsymbol{m}}\}$ from each one of the three knowledge sources, MAVEx predicts the answers' probability as $P^{\boldsymbol{k}} = \texttt{FFN}(\boldsymbol{k} + \boldsymbol{z})$, where FFN denotes a feed-forward layer. The final prediction P is the answer that has the maximum confidence over the three knowledge sources for each answer, $i.e.\ P = \max_{\boldsymbol{k}}\{P^{\boldsymbol{k}}\}$.

The validation module takes as inputs the answer candidate a and the knowledge features $\boldsymbol{k}^{a'} \in \{\boldsymbol{w}^{a'}, \boldsymbol{c}^{a'}, \boldsymbol{m}^{a'}\}$ from the three sources to learn how well the knowledge supports the answer candidate. We first embed the answer candidate using the summation of the BERT features of the corresponding statement and the glove features of the answer itself, $i.e.\ f_{ans}(a) = (\text{BERT}(s_{qa}) + \text{glove}(a))$. Then, the validation score $J(a, a')$ for answer candidate a using the knowledge retrieved for a' (a different candidate) is computed as $J^{\boldsymbol{k}}(a, a') = \text{FFN}(f_{ans}(a) \circ \boldsymbol{k}^{a'})$, where the $\circ$ means element-wise multiplication. The final validation score is the maximum validation confidence over the three knowledge sources, $i.e.\ J(a, a') = \max_{\boldsymbol{k}}\{J^{\boldsymbol{k}}(a, a')\}$.

Consistency criteria: The intuition behind our consistency criteria is that for the correct answer a, the knowledge retrieved for a from the most confident source (the one with the highest supportiveness score J for a) should support a more than it supports other answer candidates, and it should also support a more than knowledge retrieved for other answer candidates. Specifically, we approve the answer validation score $J(a, a)$ only if it is higher than the scores computed using this knowledge

for all other answers as well as the score for a when using knowledge retrieved for other answers. We also eliminate the case where the top-1 prediction from P is not in the answer candidate set. Mathematically, the consistency criteria checks that $J(a, a) > J(a', a)$ and $J(a, a) > J(a, a')$ for all $a' \neq a$. If the above condition is not met, we output the answer with the maximum VQA prediction score $P(a)$; otherwise, we output the answer with the maximum VQA-weighted validation score $J(a, a)P(a)$.

Training and Implementation Details:

Implementation. We implemented our approach on top of ViLBERT-multi-task [Lu et al., 2019], which utilizes a Mask-RCNN head [He et al., 2017] in conjunction with a ResNet-152 base network [He et al., 2016] as the object detection module. Convolutional features for at most 100 objects are then extracted for each image as the visual features, $i.e.$ a 2,048-dimensional vector for each object. We used the constituent parser from AllenNLP to extract the nouns phrases in the question. For linking the mentioned objects, we adopt a separate ViLBERT-multi-task system. For converting the question and answer, we finetuned a T5-base model [Raffel et al., 2020] on the QA-NLI dataset [Demszky et al., 2018] for 4 epochs. We detected 100 objects using Mask-RCNN to encode the retrieved Google images. For question embedding, following [Devlin et al., 2019], we use a BERT tokenizer on the question and use the first 23 tokens as the question tokens. We encode at most 4 sentences per query, 3 queries per noun phrase. The number of hidden units in the multi-head attention modules is set to 512. We use Pytorch 1.4 on a single TITAN V GPU with 12M memory for each run, and it generally costs 22 hours to train a single model.

Training. The OK-VQA test images are a subset of COCO validation images that are used to pre-train most transformer-based vision and language models [Lu et al., 2019, Tan and Bansal, 2019, Li et al., 2019]. Although the test questions never appear in the pre-training process, other questions on the test images may help the system understand the image better, leading to higher performance. Besides, there is also data contamination from extra object annotations from Visual Genome (VG) dataset, which also contains some OK-VQA test images. As the VG dataset is used to pre-train the object detector, those test images can access the ground truth object annotations. Therefore, we carefully remove all OK-VQA test images from the pre-training and re-train the ViLBERT-multi-task model and the object detector from scratch using the default configurations.

We finetune the ViLBERT-multi-task model on OK-VQA using the default configuration for 150 epochs for answer candidate generation. Binary cross-entropy loss and VQA soft score are employed to optimize the system. OK-VQA provides five annotations for each question. Soft scores are 0, 0.6, and 1 corresponding to 0, 1, more than 1 matching answer annotations. We use the finetuned model to extract the top 5 answers for each question in the training and test set. We follow the default settings of ViLBERT and apply the BertAdam optimizer [Devlin et al., 2019] with a linear warmup learning rate.

For the training of the answer validation module, we optimize the validation score $J(a, a')$ using the loss in Eq. 7.4 for the three knowledge sources, where $s(a)$ denotes the VQA soft scores for answer a. We also add the standard VQA losses on the predictions from the three external sources. We train the system for 75 epochs

using a learning rate of $2\text{e-}5$ for the ViLBERT parameters and $5\text{e-}5$ for the additional parameters introduced in the validation module. We freeze the first 10 layers of the ViLBERT base network. We use $\mathcal{L}_{bce}$ to denote binary cross-entropy loss.

$$\begin{aligned}\mathcal{L}_{\text{MAVEx}} = \mathcal{L}_{bce}\Big(\max_{\substack{a \\ s.t.\ a \neq a'}} J(a, a'),\ \mathbf{0}\Big) \\ + \mathcal{L}_{bce}\Big(\max_{\substack{a' \\ s.t.\ a \neq a'}} J(a, a'),\ \mathbf{0}\Big) \\ + \mathcal{L}_{bce}\Big(J(a, a), s(a)\Big) \end{aligned} \tag{7.4}$$

7.3 Entity-Focused Passage Retrieval for OK-VQA (EnFoRe)

Passage retrieval under a multi-modal setting is a critical prerequisite for applications such as outside-knowledge visual question answering (OK-VQA) [Marino et al., 2019], which requires effectively utilizing knowledge external to the image. Recent dense passage retrievers with deep semantic representations powered by large transformer models show appealing performance over traditional sparse retrievers such as BM25 [Robertson and Zaragoza, 2009] and TF-IDF under both textual [Karpukhin et al., 2020, Chen et al., 2021b, Lewis et al., 2022] and multi-modal settings [Luo et al., 2021, Qu et al., 2021, Gui et al., 2021]. In this work, we investigate two main drawbacks of recent dense retrievers [Karpukhin et al., 2020, Chen et al., 2021b, Lewis et al., 2022, Luo et al., 2021, Qu et al., 2021, Gui et al., 2021], which are typically trained to produce similar representations for input queries and passages containing ground-truth answers.

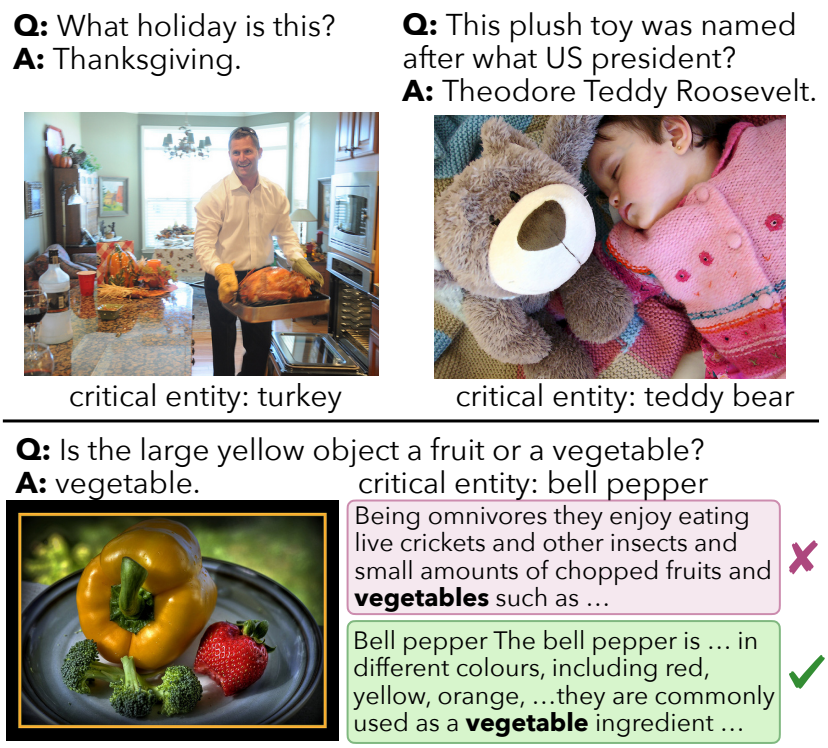


Figure 7.5: Top: Examples of critical entities upon which retrieval models should focus; Bottom: Example of improved supervision for passage retrieval using critical entities.

First, as most retrieval models encode the query and passages as a whole, they fail to explicitly discover entities critical to answering the question [Chen et al., 2021b]. This frequently leads to retrieving overly-general knowledge lacking a specific focus. Ideally, a retrieval model should identify the critical entities for the query and then retrieve question-relevant knowledge specifically about them. For example, as shown in the top half of Figure 7.5, retrieval models should realize that the entities "turkey" and "teddy bear" are critical.

Second, on the supervision side, the positive signals are often passages containing the right answers with top sparse retrieval scores such as BM 25 [Robertson and Zaragoza, 2009] and TF-IDF. However, this criterion is inadequate to guarantee question relevancy, and good positive passages should reveal facts that support the correct answer using the critical entities shown in the image. For example, as shown in the bottom of Figure 7.1, both passages mention the correct answer "vegetable" but only the second one which focuses on the critical entity "bell pepper" is question-relevant.

In order to address these shortcomings, we propose an **En**tity-**Fo**cused **Re**trieval (EnFoRe) model that improves the quality of the positive passages for stronger supervision. EnFoRe automatically identifies critical entities for the question and then retrieves knowledge focused on them. We recognize the entities that help improve a sparse retriever's performance if emphasized during retrieval as critical entities. We use the top passages containing *both* critical entities and the correct answer as positive signals. Then, our EnFoRe model learns two scores to indicate (1) the importance of each entity given the question and the image and (2) the score of each entity fitting in the context of each candidate passage.

7.3.1 Entity Set Construction

Our EnFoRe model is empowered by a comprehensive set of extracted entities that serves as a unified modality. It is worth noting that the entities are flexible and not limited to the phrases from the question and passages as in [Lee et al., 2021]. We collect entities from the following sources.

Question-based entities:

Entities from questions: First, the noun phrases in questions usually reveal critical entities in answering the visual question. Similar to MAVEx retrieval in Sec. 7.2.2, we parse the question using a constituency parser [Gardner et al., 2018] and extract noun phrases at the leaves of the parse tree. Then, we link each phrase to the image and extract the referred object with its attributes. We use a pretrained VilBERT model [Lu et al., 2020] as the object linker.

Entities from sub-questions: OK-VQA often requires systems to solve visual reference problems as general visual questions and comprehend relevant outside knowledge. Therefore, we employ a general VQA model to find answers to the visual aspect of the question. In particular, we collect a set of sub-questions by appending each noun phrase in the parse tree to the common question phrases "What is..." and "How is..." Then, we utilize the answer from a pretrained VilBERT model [Lu et al., 2020] when the VQA confidence exceeds 0.5 and add the approved answers to the entity set.

Entities from answer candidates: The answer candidate provides guidance that helps the retriever retrieve noisy facts, and state-of-the-art VQA models are surprisingly effective at generating a small set of promising answer candidates. We finetune a ViLBERT model [Lu et al., 2019] on the OK-VQA data set and extract the top 5 answer candidates for the training and test set.

Image-based entities:

Question-based entities are high precision and narrow down the search space

for knowledge retrievers. To complement this, we also collect image-based entities to help achieve higher recall.

Entities from azure tagging: Following Yang et al. [2022], we use Azure OCR and brand tagging to annotate the detected objects in the images using a Mask R-CNN detector [He et al., 2017].

Entities from Wikidata: As suggested by Gui et al. [2021], common image and object tags can be generic with a limited vocabulary, leading to noise or irrelevant knowledge. Therefore, we also leverage recent advanced visual-semantic matching approaches, i.e. CLIP [Radford et al., 2021], to extract image-relevant entities from Wikidata. In particular, the entities with their description in Wikidata and sliding windows of the images are used as inputs. Then, at most 18 entities with top maximum CLIP scores over these sliding windows are preserved.

Entities from captions: Captions provide a natural noting of salient objects in the image and do not suffer from the limited vocabulary issue of object detection. Similar to extracting entities from the question, we parse captions and extract noun phrases at the parse tree's leaves. During training, we use the human captions provided by the COCO dataset to provide richer entities, and during testing, we use generated captions from the OFA captioning model [Wang et al., 2022].

7.3.2 Oracle Critical Entity Detection

Given the comprehensive set of entities $\mathcal{E}$ covering different aspects of the question and image, we introduce an approach to automatically find critical entities and the passages containing them. Then, those entities and passages are used during

training to provide more substantial supervision. The intuition is that a good passage that fits the visual question's context should mention *both* the key entities and the correct answer. Also, emphasizing critical entities should improve retrieval performance.

Given a question q, we use BM25[3] [Robertson and Zaragoza, 2009] as the sparse retriever to retrieve a set of initial passages $\mathcal{P}_{init} = \{p_1, ..., p_K\}$. We calculate a baseline score $\texttt{SRR}_{init}$ for these K passages using summed reciprocal ranking (SRR) as shown in Eq. 7.5.

$$\texttt{SRR}(\mathcal{P}) = \sum_{i=1}^{K} \frac{1[\texttt{ans} \in p_i]}{i} \tag{7.5}$$

We adopt summed reciprocal ranking instead of reciprocal ranking. It provides more stable scores for evaluating the set of retrieved passages and is not over-ruled by ranking the first appeared passage.

Then, for each entity $e \in \mathcal{E}$, we retrieve another set of passages $\mathcal{P}_e$ using an entity-emphasizing query where the entity is appended to the end of the question. Note that the BM25 retriever does not take the order of the words in the question input account, and simply appending entities will not lead to undesired results due to the linguistic fluency of the query.

The scores for the entity $S(e)$ is computed as the different between the SRR of these two sets of retrieved passages, i.e. $S(e) = \texttt{SRR}(\mathcal{P}_e) - \texttt{SRR}(\mathcal{P}_{init})$. We regard entities with $S(e)$ over a threshold θ as critical entities, i.e. $\mathcal{E}_{oracle} = \{e \in \mathcal{E} | S(e) > \theta\}$.

[3]https://github.com/castorini/pyserini.git

Qu et al. [2021] extract the top-k passages containing the correct answer from $\mathcal{P}_{init}$ to construct the positive passage set $\mathcal{P}^{+}_{init}$. As we have identified oracle entities, the passage that contains both the right answer and the oracle entity is more likely to fit in the context of the question. Therefore, we augmented the positive passage set to include those passages for each oracle entity, $i.e.$ $\mathcal{P}^{+}_{\mathcal{E}} = \bigcup_{e \in \mathcal{E}_{oracle}} (\{p^{+}_{e}\})$, where p^{+}_{e} denotes the first passage that contains both the right answer and the oracle entity.

7.3.3 Entity-Focused Retrieval

We introduce our **En**tity-**Fo**cused **Re**trieval (EnFoRe) model that automatically recognizes critical entities and retrieves question-relevant knowledge specifically focused on them. "`proj`" denotes a projection function that consists of an MLP layer.

Encoders:

Query encoder: As observed by Qu et al. [2021], Luo et al. [2021], multi-modal transformers can better encode questions and visual content than uni-modal transformers, so we adopt LXMERT [Tan and Bansal, 2019] as our query encoder. In particular, we project the "`pooled_output`" at the last layer from LXMERT as the feature vector $f_q \in R^d$ given the query q that contains a visual question Q and the set of detected objects $\mathcal{V}$ in the image as shown in Eq. 7.6. See the LXMERT paper for further details.

$$f_q = \texttt{proj}(\texttt{LXMERT}(Q, \mathcal{V})) \tag{7.6}$$

Passage encoder: Following Qu et al. [2021], we use BERT[Devlin et al.,

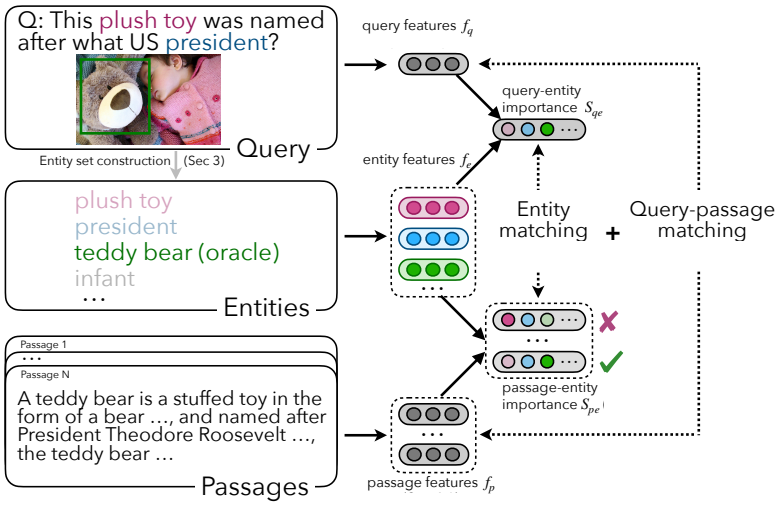


Figure 7.6: EnFoRe model overview. We first extract a set of entities from the query consisting of a question and an image. Those entities serve as an intermediate modalities for the visual and textual inputs. Then, the EnFoRe model computes the features for the query, the entities, and the passages. Query features and passage features, together with entity features, are used to compute a query-entity score and a passage-entity score to indicate the importance of the entities given the query and the passages, respectively. These two importance scores are combined to produce an entity-matching score, and the features of the query and the passages are used to predict a query-passage matching score.

2019] as the passage encoder and project the "`[CLS]`" representation to compute the vector features for each passage p.

$$f_p = \texttt{proj}(\texttt{BERT}(p)) \tag{7.7}$$

Entity encoder: In order to provide query context for each entity, we append the question and a generated image caption [Wang et al., 2022] after each entity. The input to the Entity encoder is "`[CLS] entity [SEP] question [SEP] caption`". Similar to the passage encoder, we use BERT [Devlin et al., 2019] as

the entity encoder and project the "[CLS]" representation to compute the features for each entity.

$$f_e = \texttt{proj}(\texttt{BERT}(e)) \tag{7.8}$$

Retrieval scores: Our EnFoRe model aims to retrieve question-relevant knowledge that explicitly focuses on critical entities. Therefore, the similarity metric consists of two parts: a question relevancy term and an entity focus term.

Modeling question relevancy: We model the question relevancy term S_{qp} as the inner-product of the query and passage features, i.e. $S_{qp}(q,p) = f_q^T f_p$. During inference, as the query and passage features are decomposable, maximum inner product search (MIPS) can be applied to efficiently retrieve top passages for the query.

Modeling entity focus: The entity focus term consists of two parts, where query features are used to identify critical entities from the set of entities in Sec. 3, and passage features are used to determine whether it contains these key entities. For each entity, we compute the query-entity score $S_{qe}(q,e)$ as the inner-product of the projected query and entity feature, i.e. $S_{qe}(q,e) = \texttt{proj}(f_q)^T\texttt{proj}(f_e)$, and we compute the passage-entity score as $S_{pe}(p,e) = \texttt{proj}(f_p)^T\texttt{proj}(f_e)$. Then, we combine all of the entities and compute the entity-focused score S_{qpe} per Eq. 7.9:

$$S_{qpe}(q,p,\mathcal{E}) = \frac{\sum_{e\in\mathcal{E}} \sigma(S_{qe}(q,e)) \times S_{pe}(p,e)}{\sum_{e\in\mathcal{E}} \sigma(S_{qe}(q,e))} \tag{7.9}$$

where σ denotes the sigmoid function. The final score $S(q,p)$ for the query q and passage p linearly combines both terms, i.e. $S(q,p) = S_{qp}(q,p) + \lambda S_{qpe}(q,p,\mathcal{E})$, where the weight λ controls the balance between the these two terms.

Training: We train our EnFoRe model with a set of training instances consisting of a query containing the visual question with an image, a positive passage, a retrieved negative passage, and the set of entities. We adopt the "R-Neg+IB-All" setting introduced by Qu et al. [2021] that regards the retrieved negatives, along with all other in-batch passages, as negative samplings. Following previous work [Karpukhin et al., 2020], we use cross-entropy loss to maximize the relevancy score $S_{qp}(q, p)$ and the entity focusing score $S_{qpe}(q, p, \mathcal{E})$ of the positive passage given the negatives identified above. In addition, we regard the oracle entities, defined in Sec. 7.3.2, as positive entities and others as negative entities. We use binary cross-entropy loss to supervise the importance score $S_{qe}(q, e)$. We use AdamW [Loshchilov and Hutter, 2018] with a learning rate of 1e-5 to train the EnFoRe model for 8 epochs where 10% of the iterations are used to warm up the model linearly. The batch size is set to 6 per GPU, and we use 4 GPUs (Tesla V100) for each experiment. The training process takes about 45 hours for each model. We save the parameters every 5000 steps and present the best results (MRR@5) on the validation set. The hidden states size is set to 768 following Qu et al. [2021] for fair comparison. The threshold θ for recognizing critical entities is set to 0.8. As our model consists of two BERT encoder and an LXMERT encoder, resulting in 430M parameters in total.

Inference: As the question relevancy term is decomposable, we again adopt MIPS to retrieve the top-80 passages. Then, we evaluate the entity focus term for each passage and use the combined score $S(q, p)$ to rerank the retrieved passages.

Reader: We employ the current state-of-the-art KAT model [Gui et al., 2021] as our VQA reader. The KAT model is a generation-based reader that learns to generate the

answer given the retrieved knowledge. It adopts an FiD [Izacard and Grave, 2021] architecture to incorporate both implicit knowledge, generated by a frozen GPT-3 model, and explicit knowledge. For implicit GPT-3 knowledge, the input format is "`question:ques? candidate:cand. evidence:expl.`", where the `ques`, `cand` and `expl.` denotes the question, answer and its explanation generated by the GPT-3 model [Brown et al., 2020]. For the explicit knowledge, the input format is "`question:ques? entity:ent. description:desc.`", where `ent`, `desc` denote the retrieved entity and its description. We refer our readers to the original paper [Gui et al., 2021] for further details.

We change the original explicit knowledge to the knowledge retrieved by our EnFoRe model. As the retrieved passage contains multiple sentences, and usually, not all are relevant, we select the most relevant sentence for each passage. Specifically, we convert the question and the candidate answers to a set of statements. Then, we decontextualize each sentence for each passage and compute the BertScore [Zhang et al., 2020a] between the decontextualized sentences and each statement. The sentence with the highest BertScore across these statements is extracted for each passage. The input format for us is "`question:ques?entity:ents. description:desc.`", where the `ents`, `desc` denote the top-10 entities judged by the query-entity importance score $S_{qe}(q, e)$ and the extracted sentence.

Following Gui et al. [2021], we perform experiments for two KAT settings: (1) "KAT-base + EnFoRe" setting is a single model that employs T5-base [Raffel et al., 2020] as the backbone encoder and decoder. (2) "KAT-full + EnFoRe" is an ensemble model, where each model employs T5-large as the backbone encoder

and decoder. As our knowledge is question-aware, we only encode top-10 retrieved sentences in contrast to the 40 sentences in the original KAT model. We adopt the same training scheme as the KAT model.

7.4 Experimental Setup and Results

We evaluate our framework on the OK-VQA dataset. We first briefly describe the dataset and then present our results, comparing with the current state-of-the-art systems.

OK-VQA dataset: [Marino et al., 2019] is the largest knowledge-based VQA dataset at present. The questions are crowdsourced from Amazon Mechanical Turkers, leading to two main advantages: (1) the questions indeed require outside knowledge beyond images; (2) there are no existing knowledge bases that cover all the questions, thus requiring systems to explore open-domain resources. The dataset contains 14,031 images and 14,055 questions covering a variety of knowledge categories (i.e. 9,009 for training and 5046 for test). The metric is the VQA soft score. For the experiments of this paper, we used version 1.1 of the dataset. For knowledge retrieval, we adopt the same data configuration as Qu et al. [2021] that evenly splits the test set of the OK-VQA dataset into a validation set and a test set, and we refer to these as RetVal and RetTest, respectively.

7.4.1 MAVEx Results

Answer candidate accuracy: Our answer candidate generation module, based on the finetuned ViLBERT-multi-task model, outputs its top-5 answers as the candidates.

We found that the best answer in this small candidate set achieves a VQA soft score of 59.7 on the test set, substantially higher than other state-of-the-art systems without data contamination. We also evaluate the score achieved by slightly larger candidate sets, consisting of the top 6, 8, and 10 candidates. These achieve VQA soft scores of 62.1, 65.1, and 67.1, respectively. Since our answer validation framework needs to retrieve and encode answer-specific knowledge, we use only top-5 answer candidates as a reasonable trade-off between efficiency, answer coverage, and overall accuracy. Note that our method cannot produce answers not in the candidate set.

Method	**Knowledge Resources**	**VQA**
ArticleNet (AN) [Marino et al., 2019]	Wikipedia	5.3
Q-only [Marino et al., 2019]	—	14.9
MLP [Marino et al., 2019]	—	20.7
BAN [Kim et al., 2018]	—	25.2
+ AN [Marino et al., 2019]	Wikipedia	25.6
+ KG-AUG [Li et al., 2020b]	Wikipedia + ConceptNet	26.7
MUTAN [Ben-Younes et al., 2017]	—	26.4
+ AN [Marino et al., 2019]	Wikipedia	27.8
Mucko [Zhu et al., 2020]	Dense Caption	29.2
ConceptBert [Gardères et al., 2020]	ConceptNet	33.7
KRISP [Marino et al., 2021]	Wikipedia + ConceptNet	38.9*
RVL† [Shevchenko et al., 2021]	Wikipedia + ConceptNet	39.0†
MAVEx (ours)	Wikipedia + ConceptNet	39.45*
MAVEx (ours)	Wikipedia + ConceptNet + Google Images	**40.28***
MAVEx (ours) (Ensemble 3)	Wikipedia + ConceptNet + Google Images	**41.37***

Table 7.1: MAVEx outperforms current state-of-the-art approaches on OK-VQA. The middle column lists the external knowledge sources, if any, used in each system. † indicates that the system uses a pretrained model contaminated by OK-VQA test images. * indicates that the results have been reported on version 1.1 of the dataset.

MAVEx main results: Table 7.1 shows that MAVEx consistently outperforms prior approaches by a clear margin. For example, MAVEx single model outperforms recent state-of-the-art models, KRISP [Marino et al., 2021] by 1.4 points, respectively. An

ensemble of three MAVEx models with different initializations provides 2.47 points improvement compared to KRISP. All of our results, except for the ensemble model, are averaged across 3 different initialization seeds.

Figure 7.7: Examples where the VQA model is wrong but MAVEx with the three external knowledge sources (Wikipedia, Conceptnet and Google images) answers correctly. The correct answer is in the green box and the incorrect answer is shown in the red box. The grey box shows the question. Sample retrieved knowledge content is shown in the boxes under the predicted answers.

Ablation studies on knowledge sources: We report the performance of using the different combinations of knowledge sources in Table 7.2. We see that the three sources (WikiPedia, ConceptNet, and Images) improve the performance by 3.4, 3.3,

and 3.1, respectively, compared to the base ViLBERT system. This indicates the effectiveness and value of all three sources of external information. The decontextualization technique [Choi et al., 2021] improves the performance compared to using only the Wikipedia source by 0.4%. The decontextualization partially helps address the co-reference issue since the retrieved sentences provide more information from their paragraph. Combining the three sources achieves a net performance gain of 5% over the ViLBERT baseline, supporting the intuition that the three sources together provide complementary pieces of knowledge.

We show some qualitative examples in Figure 7.7, where the VQA model (ViLBERT) is wrong but provides good answer candidates. Our MAVEx gathers the external knowledge from the three sources and predicts the correct answers.

Knowledge Source	System Score
ViLBERT	35.20
Wikipedia (*w/o* decontextualization)	38.21
Wikipedia	38.63
ConceptNet	38.56
Images	38.30
Wikipedia + ConceptNet	39.45
ConceptNet + Images	39.60
Wikipedia + Images	39.37
Wikipedia + ConceptNet + Images	40.28
Wikipedia + ConceptNet + Images (Oracle)	47.76

Table 7.2: Ablation study using different combinations of knowledge sources.

Answer validation step: We consider a MAVEx baseline model that uses the retrieved knowledge ($\tilde{\boldsymbol{w}}$, $\tilde{\boldsymbol{c}}$, $\tilde{\boldsymbol{m}}$) as additional inputs but without answer validation.

This model achieves an overall score of 39.2, 4% higher than the ViLBERT base model and 1.1% lower than the full model, indicating that using answer-guided retrieved knowledge is helpful, and answer validation further improves performance.

Oracle source selector: We report an oracle score obtained by manually choosing the best verification score $J^{\boldsymbol{k}}(a, a)$ from the three sources $\boldsymbol{k} \in \{\tilde{\boldsymbol{w}},\ \tilde{\boldsymbol{c}},\ \tilde{\boldsymbol{m}}\}$ to weigh the prediction P. As a result, our answer validation framework achieves an oracle score of 47.76 as reported in Table 7.2. This indicates that the three knowledge sources provide complementary features, leaving further potential to improve the system.

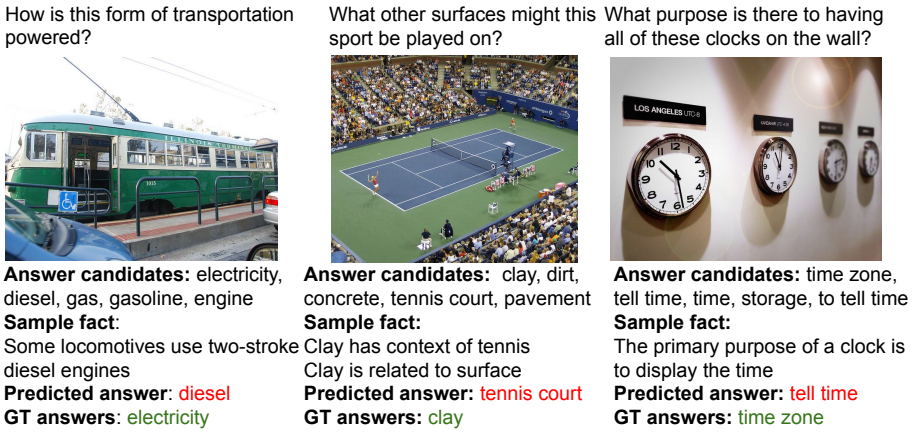


Figure 7.8: Some typical failure cases of our model have been shown. In these examples, the model falsely focuses on the retrieved fact (left), visual content (middle), or does not generate proper search word for knowledge retrieval (right).

Failures cases analysis: Figure 7.8 shows some common types of failure examples. In the left example, the model over-relies on the retrieved fact "some locomotives use diesel engines" and ignores the key visual clue in the image (the wires above

the train). In the middle example, the model relies on the visual content "tennis court" and does not use the retrieved knowledge. In the example shown on the right, the model fails to realize that the key clue is the *difference in displayed time on the clocks*.

Category	Sources			
	w	i	c	wci
VT	**35.6**	33.2	34.6	36.1
GHL	34.8	36.2	**36.6**	35.8
BCP	**30.5**	26.3	29.1	30.0
PEL	**34.7**	33.4	34.3	35.4
OMC	35.5	34.8	**35.6**	36.9
PA	38.4	**40.0**	39.3	40.2
SR	**46.4**	45.4	46.3	48.9
ST	34.1	**36.7**	32.6	35.4
CF	41.9	**42.6**	41.7	43.7
WC	50.3	49.5	**52.3**	52.6
Other	37.8	**38.1**	37.6	40.5
All	38.6	38.3	38.6	40.3

Table 7.3: Per category performance. The categories are Vehicles and Transportation (VT), Brands, Companies and Products (BCP), Objects, Material and Clothing (OMC), Sports and Recreation (SR), Cooking and Food (CF), Geography, History, Language and Culture(GHL), People and Everyday life (PEL), Plants and Animals (PA), Science and Technology (ST), and Weather and Climate (WC).

Per category performance: We report per category performance for the three external sources in Table 7.3. The detailed question categories are presented in the caption. The best category performance for the single-source models are shown in bold. We observe that different knowledge sources help the system achieve higher scores for different types of question, indicating the complementarity of those retrieved knowledge.

Methods	Val		Test	
	MRR@5	P@5	MRR@5	P@5
BM25-Obj	0.3772	0.2667	0.3686	0.2541
BM25-Cap	0.4727	0.3483	0.4622	0.3367
BM25 w. entities	0.3620	0.2558	0.3732	0.2620
BM25 w. oracle entities	0.6591	0.4548	0.6401	0.4345
DPR-LXMERT [Qu et al., 2021]	0.4704	0.3364	0.4526	0.3329
EnFoRe-LXMERT	**0.4881**	**0.3488**	**0.4800**	**0.3444**
EnFoRe-LXMERT w. oracle entities	0.4898	0.3533	0.4853	0.3451

Table 7.4: MRR and precision retreival results on OK-VQA. The first four rows present sparse retrieval results and the others are dense retrieval results.

7.4.2 Passage Retrieval Results

We present our passage retriever results in Table 7.4, comparing them with the current state-of-the-art systems. We adopt MRR and Precision at a cut of 5 as our automatic evaluation metric. The first four rows present sparse retrieval results. The BM25 approach using our oracle entities achieves an MRR@5 of 0.6401, and a precision@5 of 0.4345 on the OK-VQA RetTest set, indicating the comprehensiveness and the potential helpfulness of the extracted entities. With the help of these entities, EnFoRe-LXMERT outperforms the previous SOTA DPR-LXMERT (with the same architecture for visual and textual embedding) by 2.74% MRR@5 and 1.15% precision@5.

Ablation study on entity sources: We also performed an ablation study on entity-based re-ranking shown in Table 7.5. The EnFoRe backbone without re-ranking achieves an MRR of 0.4632, outperforming DPR [Qu et al., 2021] by 1.06%. This indicates that using our entities during training helps the retriever build better rep-

	Image-based	Question-based	MRR@5	P@5
DPR-LXMERT			0.4526	0.3329
EnFoRe (Backbone)			0.4632	0.3317
EnFoRe (Image)	✓		0.4688	0.3351
EnFoRe (Question)		✓	0.4750	0.3409
EnFoRe (Full)	✓	✓	0.4800	0.3444

Table 7.5: Ablation study on the entity sources used during re-ranking.

resentations. It is because (1) we add additional supervision that tells the retriever which entities are more likely to lead to the correct answers, and (2) we add additional training passages that contain both the oracle entities and the right answers. Image-based and Question-based entities help our EnFoRe model achieve MRR of 0.4688 and 0.4750, respectively. Our full model, taking advantage of both image- and question-based entities, achieves an MRR of 0.4800, showing that these two types of entities are complementary.

We present a further study on individual entity sources in Table 7.6. We introduce a particularly challenging "RetTest Hard" split that collects all of the examples in "RetTest" where none of the correct answers is in the entity set. Our EnFoRe model consistently achieves better retrieval performance (i.e. MRR@5 and P@5) by incorporating entities extracted from each source. On the normal RetTest set, removing entities from candidate answers yields the largest decrease in MRR@5. This is due to the fact that the candidate answers cover plenty of correct answers in the OK-VQA test split and therefore provide direct hints to the desired content. On the RetTest Hard set, image-based entities generally help improve the retrieval performance more, indicating the need for explicitly discovering critical visual clues.

Sources	RetTest		RetTest Hard	
	MRR@5	P@5	MRR@5	P@5
None	0.4632	0.3329	0.2525	0.1553
Image-based entities	0.4688	0.3351	**0.2709**	**0.1637**
Question-based entities	0.4750	0.3409	0.2594	0.1612
Full	**0.4800**	0.3444	0.2643	0.1632
w/o. Tags	0.4788	0.3410	0.2624	0.1606
w/o. Wikidata	0.4775	0.3429	0.2617	0.1574
w/o. Caption	0.4794	**0.3449**	0.2626	0.1611
w/o. Question	0.4786	0.3442	0.2647	0.1627
w/o. Sub-Question	0.4784	0.3411	0.2625	0.1605
w/o. Candidate	0.4693	0.3332	0.2664	0.1622

Table 7.6: Ablation study on entity sources.

Utilizing retrieved knowledge for VQA: We present the VQA performance of incorporating our EnFoRe knowledge in the state-of-the-art KAT model in Table 7.7. While a plain KAT-base model, which uses GPT-3 and CLIP [Radford et al., 2021] to retrieve image-based knowledge, achieves a score of (50.58)[4], switching to our EnFoRe knowledge brings a 1.7 point improvements, achieving a score of 51.34 (52.24). Our ensemble model (KAT-full + EnFoRe) achieves a new SOTA score of 54.35 (55.23).

Qualitative results: We present sample results in Figure 7.9 where (a)–(d) show cases where our EnFoRe model correctly identifies the critical entities ($i.e.$ the orange, the kite, the calico cat, and the teddy bear) and retrieved question-relevant knowledge focused on them. Case (e) shows an example where the retrieved sentence misleads the reader, because the reader currently only receives the textual input,

[4]The additional result shown in parentheses is computed by an unofficial evaluation metric that takes the max over 1.0 and number of annotators agreements divided by 3.

Method	Knowledge Resources	VQA Scores
Q-only [Marino et al., 2019]	—	14.9
BAN [Kim et al., 2018]	—	25.2
MUTAN [Ben-Younes et al., 2017]	—	26.4
Mucko [Zhu et al., 2020]	Dense Caption	29.2
ConceptBert [Gardères et al., 2020]	ConceptNet	33.7
KRISP [Marino et al., 2021]	Wikipedia + ConceptNet	38.9
RVL [Shevchenko et al., 2021]	Wikipedia + ConceptNet	39.0
PICa [Yang et al., 2022]	Frozen GPT-3	48.0
KAT-base	Frozen GPT3 + Wikidata	(50.58)
KAT-base + EnFoRe	Frozen GPT3 + Wikipedia	51.34 (52.24)
KAT-full	Frozen GPT3 + Wikidata	(54.41)
KAT-full + EnFoRe	Frozen GPT3 + Wikipedia	54.35 (55.23)

Table 7.7: EnFoRe knowledge boosts the current state-of-the-art approaches on OK-VQA. The middle column lists the external knowledge sources if any, used in each system. The additional result shown in parentheses is computed by an unofficial evaluation metric that takes the max over 1.0 and number of annotators agreements divided by 3.

and it fails to verify whether the pizza actually has a thin crust. Case (f) shows an example where the retriever properly focuses on the critical entity "NORWOOD" but fails to understand that this is the destination for the bus.

Human evaluation: We also conducted a human evaluation on AMT of the retrieved entities and sentences to demonstrate that EnFoRe retrieved knowledge better supports the correct answers. We first randomly sampled 1,000 test questions that are correctly answered by both the original KAT-base model and our "KAT-base + EnFoRe" model. Next, we extracted the top-3 sentences with the highest attention score averaged over all attention heads from the last decoder layer for both models. We also extracted the top-3 visual entities. For EnFoRe, the top-3 entities with the

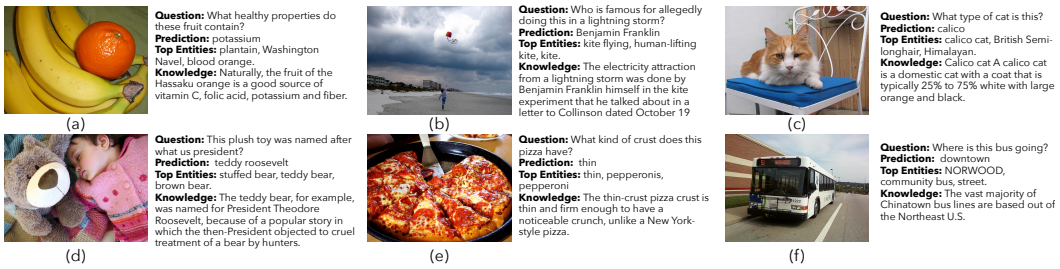


Figure 7.9: Qualitative results on EnFoRe; (a)-(d) present cases where EnFoRe correctly identifies the critical entities and retrieved question-relevant knowledge properly focuses on them; (e) and (f) present two failure cases.

highest attention scores in the input prompts are selected. For the original KAT model, we use the three entities from the three top retrieved sentences. Next, we show AMT workers the question, the predicted answer, the image with bounding boxes for the top entities, and the three retrieved sentences, for both systems randomly ordered. Finally, workers are asked to judge which system's set of highlighted entities and sentences best supports the given answer. Experimental results show that judges pick our EnFoRe knowledge 61.8% of the time, indicating a clear preference over the original KAT knowledge. Such information can be considered an explanation or rationale for the system's answer, and improved explanations can engender greater trust and acceptance from users and provide additional transparency of the system's operation. Figure 7.10 shows a sample question from a HIT. We eliminate data where the quality control is not passed, but pay the workers 80 cents for finishing the HIT regardless of passing the quality control example. The average time workers spent on each HIT is 2 min and 33 sec.

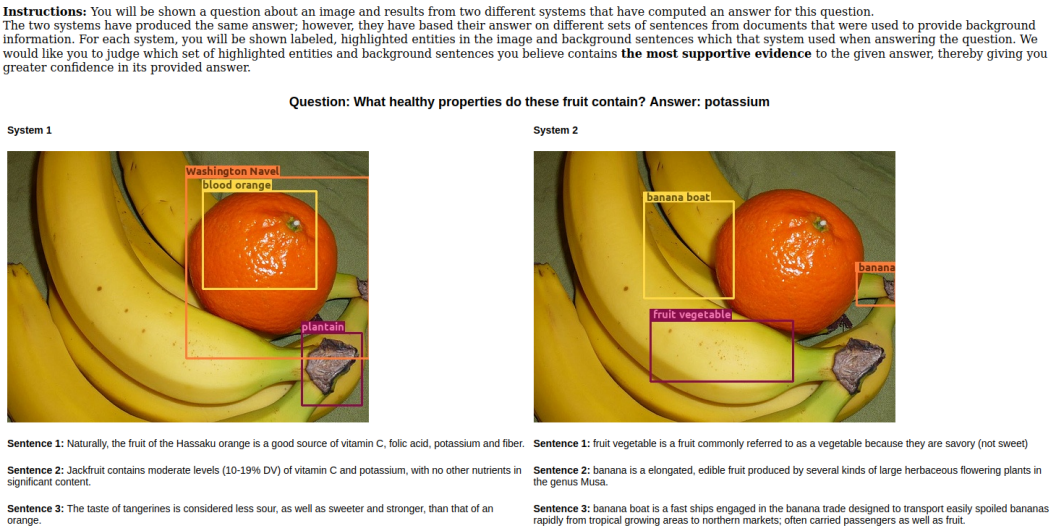


Figure 7.10: Sample question for the human evaluation. We ask the turkers to judge which system's set of highlighted entities and sentences best supports the given answer.

7.5 Chapter Summary

In this chapter, we present the work on utilizing external knowledge for outside-knowledge VQA. We present an answer validation (MAVEx) framework that incorporate multimodal knowledge supporting a set of promising answer candidates. As the quality of the retrieved knowledge is crucial to the OK-VQA performance, we further developed a entity-focused retrieval (EnFoRe) model that retrieves question-relevant knowledge for the entities that are critical to answering the question.

Chapter 8

Future Directions

In this chapter, we discuss a few future directions motivated by existing works in this dissertation, including building explainable VQA models and retrieving helpful information for VQA.

8.1 Explainable VQA Model

While we present our explanation generation model in Chapter 6, it suffers from the unfaithful issue of simply generating the justification instead of revealing what the model learns. We try to alleviate this issue by forcing the generated explanation, and the VQA model focuses on the same set of objects. However, it is not enough to explain the logic and reasoning behind the model decision.

We present a framework for explaining VQA models by introducing a "interpretable space" that contains a set of human interpretable units. The explanation model examines each item in the interpretable space and generates a final explanation. The previous explanation model in Chapter 5 sets the interpretable space as the set of detected objects and attributes and uses a gradient-based method as the examination approach. We discuss the future work below concerning the span of the interpretable space, the examination methods, and the final explanation generation

approach.

Span of the interpretable space: The span of the interpretable space is of great importance in this framework. It not only influences the choice of the examination and the explanation generation approaches but also determines the maximum capacity of the faithfulness that this framework can achieve. The vanilla Grad-CAM and visual attention treat the image grid features as interpretable space and achieve pixel-level faithfulness. Our explanation model ensures the generated explanation and the VQA model focus on the same set of objects and achieves object-level faithfulness. In the future, we would like to expand the space to include commonsense rationale triplets to allow the explanation generation model to elaborate on the relations between objects and scenes using commonsense.

Examination of the VQA model: Given the interpretable space, the examination methods find out which item influences the VQA model most. Previous works [Zhou et al., 2018] mainly model these items individually that simultaneously predict a weight for each of them. As the interpretable space has been expanded, we would like to explore graph structure examination to determine which sub-graph in the interpretable space influences the VQA model most.

Explanation generation: With identified important items in the interpretable space, the explanation generation model aims to generate the final explanation in a human interpretable format. To illustrate the reasoning process, textual explanations are widely adopted as the output format. In the future, we would like to explore generating faithful textual explanations with the sub-graph in the interpretable space that are identified as influential. In order to achieve faithfulness, it is not feasible

for the explanation generation model to directly learn from the human annotations using the subgraph as inputs. Instead, we would like to incorporate an auto-encoder approach that introduces an explanation grounding model to project the human explanation to the interpretable space and then generate the explanation back during training. We can use the generation taking the subgraph features as inputs during the test.

8.2 Information Retrieval

Retrieving question helpful information is the fundamental prerequisite for answering visual questions. In this dissertation, we have explored using image captions, explanations, Wikipedia articles, commonsense rationales, and google images as information resources. However, this retrieved knowledge is still inadequate because of the imperfection of the retrieval model and the insufficiency of the covered material. We list a few future directions following these two threads below.

Retrieval model: As presented in chapter 7, knowledge retrieval literature has witnessed a shift from the sparse retriever to the dense retrievers in order to better capture the semantic of the query and the candidate passage. However, most question-answer models adopt the two-stage framework where the knowledge retriever is not aware of the actual reasoning process in the answer predictor. It forces the retriever to "guess" the right answer given the query and retrieve query-relevant knowledge containing the guessed answer. In order to alleviate this issue, we would like to explore logic-aware knowledge retrieval approaches. We first break down the question into a sequence of sub-questions and use each step as an intermediate

criterion to measure the query's relevancy and the candidate passage.

Information source: While we have explored retrieving knowledge from Wikipedia and Conceptnet, there is much richer knowledge from the internet articles. We have witnessed a trend toward web knowledge retrieval and would like to pursue this direction in the future. Unlike articles in Wikipedia, web articles reveal more commonsense and event knowledge in time. However, a few issues limit the performance of the current retrieval model. (1) As the passage candidate pool is extremely large, the retrieval model needs to filter out better irrelevant passages. This requires the retrieval model to understand the questions better and the logic of answering the question. We propose to use sub-questions as a hint to reveal that logic. (2) As the web articles are uncertified, that does not guarantee the correctness of the knowledge; the retrieval model also needs to judge the confidence that the knowledge is correct.

8.3 Incorporating Open Knowledge for Other Tasks

Multimodal pretraining: Nowadays, large multimodal transformers are pretrained using general caption datasets, e.g. COCO [Chen et al., 2015], Conceptual Captions [Sharma et al., 2018]. This large volume of caption data equips the large transformers with some commonsense knowledge. However, it is still inadequate because real world applications often require specific knowledge beyond commonsense. In this case, external open knowledge is necessary and we believe injecting this open knowledge will be beneficial to a number of downstream tasks including VQA, visual dialog, VCR, information retrieval, etc. In future, we would like to work on this topic.

Object detection: We notice a recent trend toward open vocabulary object detection, where captions [Zareian et al., 2021] are used as the source of vocabulary. However, most captions describe objects and scenes in a casual way that only present common objects (e.g. cats and dogs) or common names for specific objects (e.g. clock tower for the "Big Ben"). There is less work on utilizing external knowledge to construct the attributes for fine-grained object detection. In the future, we would like to explore in the direction toward building a knowledge-augmented fine-grained object detection.

Chapter 9

Conclusion

This dissertation presents our work on incorporating external information for VQA. Specifically, we consider image captions, visual question explanations, external factual, commonsense, and visual knowledge as the complementing information. We study the VQA task on both the performance and the explainability sides. We list a summary of individual contributions below.

In Chapter 3, we proposed framework that generates and integrates question-relevant captions for improving VQA performance. The relevancy criterion is fully automated and motivated by the assumption that question-relevant captions describe the objects on which the VQA model focuses.

In Chapter 4, human explanations are used to tell the VQA model where to focus so that the VQA model can be right for the right reason even during changing answer distribution.

In Chapter 5 and Chapter 6, we generate object-level faithful explanations for VQA. Then, we present a verification scheme that compares the explanations (retrieved or generated) for competing answer candidates. This new scheme helps both the VQA performance and its interpretability.

In Chapter 7, we focused on outside-knowledge VQA and proposed an

answer validation framework and an entity-focused retrieval model.

Finally, we discuss future directions towards (1) an explainable VQA model;(2) incorporating richer web information and (3) utilizing open knowledge for other tasks such as object detection and multimodal pretraining.

Bibliography

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the Behavior of Visual Question Answering Models. In *EMNLP*, 2016.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering. In *CVPR*, 2018.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: Semantic Propositional Image Caption Evaluation. In *ECCV*, 2016.

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 VQA. In *CVPR*, 2018.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural Module Networks. In *CVPR*, 2016.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *ICCV*, 2015.

Satanjeev Banerjee and Alon Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings*

of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, 2005.

Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. MUTAN: Multimodal Tucker Fusion for Visual Question Answering. In *ICCV*, 2017.

Mustafa Bilgic and Raymond Mooney. Explaining Recommendations: Satisfaction vs. Promotion. In *Proceedings of Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research at the 2005 International Conference on Intelligent User Interfaces*, 2005. URL `http://www.cs.iit.edu/~ml/pdfs/bilgic-iui05-wkshp.pdf`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models are Few-Shot Learners. In *NeurIPS*, 2020.

Remi Cadene, Hedi Ben-Younes, Matthieu Cord, and Nicolas Thome. Murel: Multimodal relational reasoning for visual question answering. In *CVPR*, 2019.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *ACL*, 2017.

Kezhen Chen, Qiuyuan Huang, Yonatan Bisk, Daniel McDuff, and Jianfeng Gao. KB-VLP: Knowledge Based Vision and Language Pretraining. In *Proceedings of the 38 th International Conference on Machine Learning, PMLR 139, 2021. ICML, workshop, 2021*, July 2021a.

Xilun Chen, Kushal Lakhotia, Barlas Oğuz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. Salient Phrase Aware Dense Retrieval: Can a Dense Retriever Imitate a Sparse One? *arXiv preprint arXiv:2110.06918*, 2021b.

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.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*, 2014.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. Decontextualization: Making Sentences Stand-Alone. *TACL*, 2021.

Peter Clark, Oren Etzioni, Daniel Khashabi, Tushar Khot, Bhavana D. Mishra, Kyle Richardson, A. Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, Sumithra Bhakthavatsalam, Dirk Groeneveld, Michal Guerquin, and Michael Schmitz. From 'F' to 'A' on the N.Y. Regents Science Exams: An Overview of the Aristo Project. *AI Magazine*, 2020.

Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions? In *Computer Vision and Image Understanding*, 2017.

Dorottya Demszky, Kelvin Guu, and Percy Liang. Transforming question answering datasets into natural language inference datasets. *arXiv preprint arXiv:1809.02922*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In *CVPR*, pages 2625–2634, 2015.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (voc) Challenge. *International journal of computer vision*, 88(2):303–338, 2010.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. *EMNLP*, 2016.

Chuang Gan, Yandong Li, Haoxiang Li, Chen Sun, and Boqing Gong. VQS: Linking

Segmentations to Questions and Answers for Supervised Attention in VQA and Question-Focused Semantic Segmentation. In *ICCV*, 2017.

François Gardères, Maryam Ziaeefard, Baptiste Abeloos, and Freddy Lecue. ConceptBert: Concept-Aware Representation for Visual Question Answering. In *EMNLP*, 2020.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, 2018.

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 *CVPR*, 2017.

Liangke Gui, Borui Wang, Qiuyuan Huang, Alex Hauptmann, Yonatan Bisk, and Jianfeng Gao. KAT: A Knowledge Augmented Transformer for Vision-and-Language. *arXiv preprint arXiv:2112.08614*, 2021.

Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz Grand Challenge: Answering Visual Questions from Blind People. In *ICCV*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification. In *ICCV*, pages 1026–1034, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating Visual Explanations. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding Visual Explanations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 264–279, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780, 1997.

Matthew Honnibal and Ines Montani. spaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing. 2017.

Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. Explainable Neural Computation via Stack Neural Module Networks. In *ECCV*, 2018.

Drew A Hudson and Christopher D Manning. GQA: a New Dataset for Compositional Question Answering over Real-World Images. In *CVPR*, 2019.

Gautier Izacard and Edouard Grave. Distilling Knowledge from Reader to Retriever for Question Answering. In *ICLR*, 2021.

Allan Jabri, Armand Joulin, and Laurens van der Maaten. Revisiting Visual Question Answering Baselines. In *European conference on computer vision*, pages 727–739. Springer, 2016.

Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0. 1: the Winning Entry to the VQA Challenge 2018. *arXiv*, 2018.

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.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense Passage Retrieval for Open-Domain Question Answering. In *EMNLP*, 2020.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to Objects in Photographs of Natural Scenes. In *EMNLP*, 2014.

Tushar Khot, Ashish Sabharwal, and Peter Clark. Answering Complex Questions Using Open Information Extraction. In *ACL*, 2017.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear Attention Networks. In *NeurIPS*, 2018.

Junyeong Kim, Minuk Ma, Kyungsu Kim, Sungjin Kim, and Chang D. Yoo. Progressive Attention Memory Network for Movie Story Question Answering. In *CVPR*, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

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. *IJCV*, 2017.

Jinhyuk Lee, Alexander Wettig, and Danqi Chen. Phrase Retrieval Learns Passage Retrieval, Too. In *EMNLP*, 2021.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end Neural Coreference Resolution. In *EMNLP*, 2017.

Patrick Lewis, Barlas Oğuz, Wenhan Xiong, Fabio Petroni, Wen-tau Yih, and Sebastian Riedel. Boosted Dense Retriever. In *NAACL*, 2022.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. Unicoder-VL: A Universal Encoder for Vision and Language by Cross-Modal Pre-Training. In *AAAI*, 2020a.

Guohao Li, Xin Wang, and Wenwu Zhu. Boosting Visual Question Answering with Context-aware Knowledge Aggregation. In *ACMMM*, 2020b.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

Qing Li, Jianlong Fu, Dongfei Yu, Tao Mei, and Jiebo Luo. Tell-and-Answer: Towards Explainable Visual Question Answering using Attributes and Captions. *arXiv preprint arXiv:1801.09041*, 2018a.

Qing Li, Qingyi Tao, Shafiq Joty, Jianfei Cai, and Jiebo Luo. VQA-E: Explaining, Elaborating, and Enhancing Your Answers for Visual Questions. *ECCV*, 2018b.

Chin-Yew Lin. Rouge: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out*, 2004.

Bei Liu, Zhicheng Huang, Zhaoyang Zeng, Zheyu Chen, and Jianlong Fu. Learning Rich Image Region Representation for Visual Question Answering. *arXiv preprint arXiv:1910.13077*, 2019.

Xihui Liu, Hongsheng Li, Jing Shao, Dapeng Chen, and Xiaogang Wang. Show, Tell and Discriminate: Image Captioning by Self-retrieval with Partially Labeled Data. *ECCV*, 2018.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2018.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical Question-Image Co-attention for Visual Question Answering. In *NeurIPS*, 2016.

Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6, 2017.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*, 2019.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-Task Vision and Language Representation Learning. In *CVPR*, 2020.

Pan Lu, Lei Ji, Wei Zhang, Nan Duan, Ming Zhou, and Jianyong Wang. R-VQA: learning visual relation facts with semantic attention for visual question answering. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1880–1889. ACM, 2018.

Man Luo, Yankai Zeng, Pratyay Banerjee, and Chitta Baral. Weakly-Supervised Visual-Retriever-Reader for Knowledge-based Question Answering. In *EMNLP*, 2021.

Ruotian Luo, Brian Price, Scott Cohen, and Gregory Shakhnarovich. Discriminability Objective for Training Descriptive Captions. In *CVPR*, June .

Yuning Mao et al. Generation-Augmented Retrieval for Open-Domain Question Answering. In *ACL*, 2021.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A Visual Question Answering Benchmark Requiring External Knowledge. In *CVPR*, 2019.

Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. KRISP: Integrating Implicit and Symbolic Knowledge for Open-Domain Knowledge-Based VQA. In *CVPR*, 2021.

Todor Mihaylov et al. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *EMNLP*, 2018.

Medhini Narasimhan and Alexander G Schwing. Straight to the facts: Learning knowledge base retrieval for factual visual question answering. In *ECCV*, 2018.

Medhini Narasimhan, Svetlana Lazebnik, and Alexander Schwing. Out-of-The-Box: Reasoning with Graph Convolution Nets for Factual Visual Question Answering. In *NeurIPS*, 2018.

Duy-Kien Nguyen, Vedanuj Goswami, and Xinlei Chen. Movie: Revisiting Modulated Convolutions for Visual Counting and Beyond. In *ICLR*, 2021.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *ACL*, 2002.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal Explanations: Justifying Decisions and Pointing to the Evidence. In *CVPR*, 2018.

Badri Patro, Shivansh Patel, and Vinay Namboodiri. Robust Explanations for Visual Question Answering. In *WACV*, 2020.

Marco Pedersoli, Thomas Lucas, Cordelia Schmid, and Jakob Verbeek. Areas of Attention for Image Captioning. In *ICCV-International Conference on Computer Vision*, 2017.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *EMNLP*, 2014.

Tingting Qiao, Jianfeng Dong, and Duanqing Xu. Exploring Human-Like Attention Supervision in Visual Question Answering. In *AAAI*, 2018.

Chen Qu, Hamed Zamani, Liu Yang, W Bruce Croft, and Erik Learned-Miller. Passage Retrieval for Outside-Knowledge Visual Question Answering. In *SIGIR*, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models from Natural Language Supervision. In *ICML*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR*, 2020.

Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming Language Priors in Visual Question Answering with Adversarial Regularization. In *NeurIPS*, 2018.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring Models and Data for Image Question Answering. In *NIPS*, pages 2953–2961, 2015a.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In *NIPS*, volume 28, 2015b.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical Sequence Training for Image Captioning. In *CVPR*, volume 1, page 3, 2017.

Stephen Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Information Retrieval*, 2009.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object Hallucination in Image Captioning. In *EMNLP*, 2018.

Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the Right Reasons: Training Differentiable Models by Constraining Their Explanations. In *IJCAI*, 2017.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. In *ICCV*, 2000.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *ICCV*, 2017.

Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Dhruv Batra, and Devi Parikh. Taking a HINT: Leveraging Explanations to Make Vision and Language Models More Grounded. In *ICCV*, 2019.

Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-Consistency for Robust Visual Question Answering. In *CVPR*, 2019.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

Violetta Shevchenko, Damien Teney, Anthony Dick, and Anton van den Hengel. Reasoning over Vision and Language: Exploring the Benefits of Supplemental Knowledge. In *Proceedings of the Third Workshop on Beyond Vision and LANguage: inTEgrating Real-world kNowledge (LANTERN)*, 2021.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA Models That Can Rread. In *CVPR*, 2019.

Jasdeep Singh, Vincent Ying, and Alex Nutkiewicz. Attention on Attention: Architectures for Visual Question Answering (VQA). *arXiv preprint arXiv:1803.07724*, 2018.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. PullNet: Open Domain Question Answering with Iterative Retrieval on Knowledge Bases and Text. In *EMNLP*, 2019.

Hao Tan and Mohit Bansal. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *EMNLP*, 2019.

Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. Tips and Tricks for Visual Question Answering: Learnings from the 2017 Challenge. *arXiv preprint arXiv:1708.02711*, 2017.

Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. In *NeurIPS*, 2014.

Alexander Trott, Caiming Xiong, and Richard Socher. Interpretable Counting for Visual Question Answering. 2018.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. *arXiv preprint arXiv:1908.08962*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-Based Image Description Evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575, 2015.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models. *arXiv preprint arXiv:1610.02424*, 2016.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and Tell: A Neural Image Caption Generator. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3156–3164. IEEE, 2015.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663, 2017.

Peng Wang, Qi Wu, Chunhua Shen, Anton van den Hengel, and Anthony Dick.

Explicit Knowledge-based Reasoning for Visual Question Answering. In *IJCAI*, 2017.

Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Fvqa: Fact-based Visual Question Answering. *TPAMI*, 2018.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Unifying Architectures, Tasks, and Modalities through a Simple Sequence-to-Sequence Learning Framework. *arXiv preprint arXiv:2202.03052*, 2022.

Jialin Wu and Raymond J Mooney. Faithful Multimodal Explanation for Visual Question Answering. In *ACL BlackboxNLP Workshop*, 2019a.

Jialin Wu and Raymond J Mooney. Hidden State Guidance: Improving Image Captioning using An Image Conditioned Autoencoder. *arXiv preprint arXiv:1910.14208*, 2019b.

Jialin Wu and Raymond J Mooney. Self-Critical Reasoning for Robust Visual Question Answering. *arXiv*, 2019c.

Jialin Wu, Gu Wang, Wukui Yang, and Xiangyang Ji. Action Recognition with Joint Attention on Multi-level Deep Features. *arXiv preprint arXiv:1607.02556*, 2016a.

Jialin Wu, Dai Li, Yu Yang, Chandrajit Bajaj, and Xiangyang Ji. Dynamic Filtering with Large Sampling Field for ConvNets. *ECCV*, 2018.

Jialin Wu, Zeyuan Hu, and Raymond J Mooney. Generating Question Relevant Captions to Aid Visual Question Answering. In *ACL*, 2019.

Jialin Wu, Liyan Chen, and Raymond J Mooney. Improving VQA and its Explanations by Comparing Competing Explanations. *arXiv preprint arXiv:2006.15631*, 2020.

Jialin Wu, Jiasen Lu, Ashish Sabharwal, and Roozbeh Mottaghi. Multi-Modal Answer Validation for Knowledge-Based VQA. In *AAAI*, 2022.

Qi Wu, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Ask Me Anything: Free-Form Visual Question Answering Based on Knowledge From External Sources. In *CVPR*, 2016b.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016c.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual Entailment: A Novel Task for Fine-Grained Image Understanding. *arXiv preprint arXiv:1901.06706*, 2019.

Huijuan Xu and Kate Saenko. Ask, Attend and Answer: Exploring Question-guided Spatial Attention for Visual Question Answering. In *ECCV*, pages 451–466. Springer, 2016.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An Empirical Study of GPT-3 for Few-Shot Knowledge-Based VQA. In *AAAI*, 2022.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked Attention Networks for Image Question Answering. In *CVPR*, pages 21–29, 2016.

Zhou Yu, Jun Yu, Chenchao Xiang, Jianping Fan, and Dacheng Tao. Beyond Bilinear: Generalized Multimodal Factorized High-Order Pooling for Visual Question Answering. *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *CVPR*, 2019.

Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14393–14402, 2021.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From Recognition to Cognition: Visual Commonsense Reasoning. In *CVPR*, 2019.

Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-Down Neural Attention by Excitation Backprop. *IJCV*.

Tianyi Zhang, Varsha Kishore, Felix Wu Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. In *ICLR*, 2020a.

Wenxuan Zhang, Yang Deng, Jing Ma, and Wai Lam. AnswerFact: Fact Checking in Product Question Answering. In *EMNLP*, 2020b.

Yundong Zhang, Juan Carlos Niebles, and Alvaro Soto. Interpretable Visual Question Answering by Visual Grounding from Attention Supervision Mining. In *WACV*, 2019.

Zeyu Zhang, Thuy Vu, and Alessandro Moschitti. Joint Models for Answer Verification in Question Answering Systems. In *ACL-IJCNLP*, August 2021.

Wenbo Zheng, Lan Yan, Chao Gou, and Fei-Yue Wang. Knowledge is Power: Hierarchical-Knowledge Embedded Meta-Learning for Visual Reasoning in Artistic Domains. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2360–2368, 2021.

Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified Vision-Language Pre-Training for Image Captioning and VQA. In *AAAI*, 2020.

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as Points. *arXiv preprint arXiv:1904.07850*, 2019.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded Question Answering in Images. In *CVPR*, 2016.

Zihao Zhu, Jing Yu, Yujing Wang, Yajing Sun, Yue Hu, and Qi Wu. Mucko: Multi-Layer Cross-Modal Knowledge Reasoning for Fact-based Visual Question Answering. In *IJCAI*, 2020.

Vita

Jialin Wu was born and brought up in Beijing, China. He obtained his bachelor degree in engineering and economics from the department of automation at Tsinghua University. He enrolled in the PhD program in the Department of Computer Science at the University of Texas at Austin in August 2017. His research focuses on commonsense reasoning and knowledge-based model in vision-language tasks, such as open-knowledge retrieval, question answering and explanation generation. His work has been published at top venues including ECCV 2018, ACL 2019, NeurIPS 2019, COLING 2020, AAAI2021, AAAI 2022. During his PhD, he has also completed summer internships at Google, New York, and AI2 remotely.

Permanent address: jialinwu@utexas.edu

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.