Natural Language Driven Image Edits using a Semantic Image Manipulation Language (SIMPL)

Akrit Mohapatra

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Jia-Bin Huang, Co-Chair
Dhruv Batra, Co-Chair
A. Lynn Abbott

April 18, 2018
Blacksburg, Virginia

Keywords: Machine Learning, Natural language Processing, Computer Vision

Natural Language Driven Image Edits using a Semantic Image Manipulation Language (SIMPL)

Akrit Mohapatra

(ABSTRACT)

Language provides us with a powerful tool to articulate and express ourselves! Understanding and harnessing the expressions of natural language can open the doors to a vast array of creative applications. In this work we explore one such application - natural language based image editing. We propose a novel framework to go from free-form natural language commands to performing fine-grained image edits.

Recent progress in the field of deep learning has motivated solving most tasks using end-to-end deep convolutional frameworks. Such methods have shown to be very successful even achieving super-human performance in some cases. Although such progress has shown significant promise for the future we believe there is still progress to be made before their effective application to a task like fine-grained image editing. We approach the problem by dissecting the inputs (image and language query) and focusing on understanding the language input utilizing traditional natural language processing (NLP) techniques. We start by parsing the input query to identify the entities, attributes and relationships and generate a *command entity representation*. We define our own high-level image manipulation language that serves as an intermediate programming language connecting natural language requests that represent a creative intent over an image into the lower-level operations needed to execute them. The semantic command entity representations are mapped into this high-level language to carry out the intended execution.

Natural Language Driven Image Edits using a Semantic Image Manipulation Language (SIMPL)

Akrit Mohapatra

(GENERAL AUDIENCE ABSTRACT)

Image editing is a very challenging task that requires a specific skill set. Hence, Going from natural language to directly performing image edits thereby automating the entire procedure is a challenging problem as well as a potential application that could benefit widespread users. There are multiple stages involved in such a process starting with understanding the intent of a command provided in natural language, identifying the editing tasks represented by it and the different objects and properties of the image the command intends to act upon and finally performing the intended edit(s).

There has been significant progress in the field of natural language processing as well as computer vision in recent years. On the natural language front computers are now able to accurately parse sentences, analyze large amounts of text, classify sentiments and emotions and much more. Similarly on the computer vision side computers can accurately identify objects, localize them and even generate real life like images from random noise pixels.

In this work, we propose a novel framework that enables us to go from natural language commands to performing image edits. Our approach starts by parsing the language input, identifying the entities and relations in the image from the language followed by mapping it into a set of sequential executable commands in an intermediate programming language that we define to execute the edit.

Acknowledgments

I would like to sincerely thank Dr. Dhruv Batra for giving me an opportunity to work and be part of the Machine Learning & Perception Lab at Virginia Tech! I would also like to thank Dr. Devi Parikh for all the interactions and collaborations during my time at the lab. Working with them has been extremely inspiring and an amazing learning experience. My journey in the lab dates back to my sophomore days as an undergraduate at Virginia Tech. I'm also grateful to all other members of the lab both past and present. It's been an honor to be a part of such a lab and be surrounded by amazingly talented and hard-working people. The friendships I've made along the way will always be the best. I would like to thank Dr. Jia-Bin Huang and Dr. Lynn Abbott for being part of my thesis committee. Their inputs and advice have been very valuable for the completion of this work.

I would specially like to thank Matthew Fisher for providing me with the amazing opportunity to work on this project and being an awesome mentor. I learned a lot through our daily interactions during and post my internship at Adobe.

This journey would not have been possible without the love and support of all my friends especially the Virginia Tech Cricket team as well as my family who stood by me through all the highs and lows. Finally, no words can truly describe my gratitude towards my parents. Their love and support through my life has made me who I am today and I shall be forever indebted to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image editing is a challenging task that requires a specific set of skills with a steep learning curve. Modern image editing software are highly complicated and require an in-depth domain knowledge to effectively use them. Majority of users face difficulty and seek professional expertise for editing their images. The task could be made accessible to a wide user base if we could go from natural language to the final goal without having to deal with the complicated procedures. Natural language provides us with the tool to articulate and express ourselves while extending the boundaries of creativity.

Recent progress in the field of deep learning has pushed the research community to treat most problems as end-to-end tasks using convolutional networks while achieving super-human performance in few tasks. Although such progress has shown significant promise for the future we believe there is still progress to be made before it can be used effectively for the specific task of language-based image editing. We approach the problem by focusing on the language side of the problem utilizing traditional natural language processing (NLP) techniques. We start by parsing the language to identify the entities, attributes and relationships and representing it as a *command-entity representation*. We define our own high-level Semantic Image Manipulation Language (SIMPL) that serves as an intermediate programming language to connect natural language commands representing a creative intent over an image into the lower-level operations needed to execute this intention. The representations of the commands are then mapped into this intermediate language to finally execute the edit.

Language-based image editing is indeed a very interesting problem. Yet it poses a series of challenging sub-problems. Firstly, it requires parsing and understanding the intent of a given input, identifying the entities involved, their attributes and relationships. Next, it requires being able to ground the entities in the image and finally being able to accurately execute the intent of the command to produce the final output image.

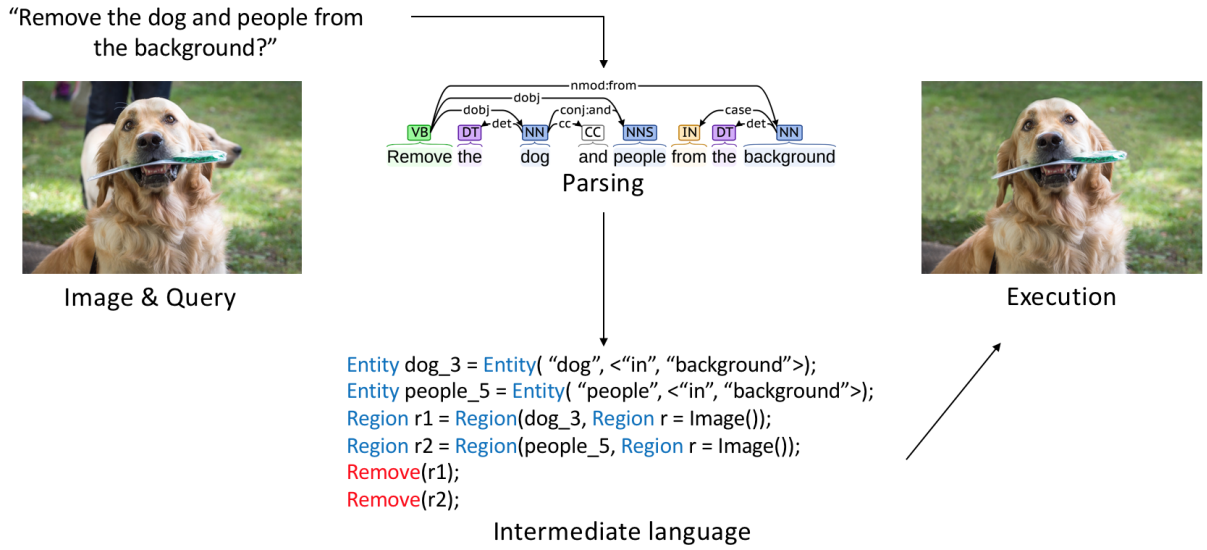

Figure 1.1: An example highlighting the pipeline to go from the original image and the natural language command to SIMPL commands and eventually the edited output image.

In this work we propose a novel framework to go from free-form natural language commands to performing fine-grained image edits. Our approach focuses on a sequential step by step execution pipeline starting with parsing the language input, followed by generating a *command-entity representation* and finally transforming these representations into our high-level Semantic Image Manipulation Language (SIMPL) commands for execution. We curate a dataset for our task by crawling the publicly available /r/PhotshopRequest subreddit. Each post on the subreddit contains an original image and a natural language image edit request. The requests are then carried out and posted by other users as comments on the original thread to be voted upon by the original poster. This provides a large variety of common editing requests from which we manually develop rules for language parsing.

1.1 Publications

1. Chapters 1 - 6 describe work done as part of an internship at Adobe Research in collaboration with Matthew Fisher and Dhruv Batra.

2. Chapter 7 talks about a previous publication:
   A. Mahendru, V. Prabhu, A. Mohapatra, D. Batra, and S. Lee. The Promise of Premise: Harnessing Question Premises in Visual Question Answering. This work was published at *EMNLP*, 2017 [27] and as part of A. Mahendru's thesis work [26].

Chapter 2

Related Work

Language-based Image Editing: Traditional image editing is a well studied area in the computer graphics community. However, language based image editing is yet to be extensively explored. Chen et al. [2018] [6] recently introduced the task of language based image editing using recurrent attentive models. Their focus is constrained to developing a generic modeling framework for image segmentation and colorization tasks. Our work focuses on being able to handle a wide-range of commonly used editing tasks. For this reason we specifically deal with the language side of the problem instead of formulating the problem as an end-to-end image and language combined task.

PixelTone: The most related work to ours is PixelTone by Laput et. al [2013] which presented a multimodal interface that combined speech and direct image manipulation [20]. The multimodal interface consists of a speech recognition component that converts users' voice to text. The output text is then mapped to an image processing request by combining natural language processing and keyword matching techniques. Finally, an execution engine carries out the image processing operation. Our approach deals with natural language text input (which can be extended to voice commands by adding a voice-to-text engine prior to the text processing) by mapping into a high-level intermediate language to carry out the low-level image edit execution.

Natural Language Processing: For the purpose of parsing and understanding the intent of a natural language command input we use the widely popular Stanford CoreNLP Natural Language Processing toolkit released by Manning et al. [2014] [30]. The CoreNLP toolkit provides extensive grammar analysis tools such as providing base forms of words, identifying parts of speech, named-entity recognition etc. We specifically use the Parts of Speech (PoS) tagger to tokenize and tag each word in a sentence (e.g. nouns, verbs, adjectives etc.). We then use CoreNLP's 'Enhanced++' dependency annotator to analyze the grammatical structure and relations between words in a sentence.

Scene Graphs: Scene graphs are extensively used in computer vision tasks as they enable unique representations of objects, attributes and the various relationships in a given image [15] [38]. For example, Fisher et. al [2011] [9] transform scenes into a scene graph for the task of 3D modeling. The nodes represent semantically meaningful objects, and edges represent different types of relationships between nodes. In this work we use the notion of scene graphs to build a semantic representation from language. The nodes represent the various entities and the edges present the different attributes and relationships between the entities.

Domain-specific Language: Domain specific languages have been around since the evolution of programming languages. It refers to a computer programming language that is dedicated to a specific task or particular problem domain. We define a high-level intermediate language specific to the domain of image editing. The natural language input to our proposed framework is eventually transformed to this Semantic Image Manipulation Language (SIMPL) for executing the intended edit.

Chapter 3

Overview & Dataset

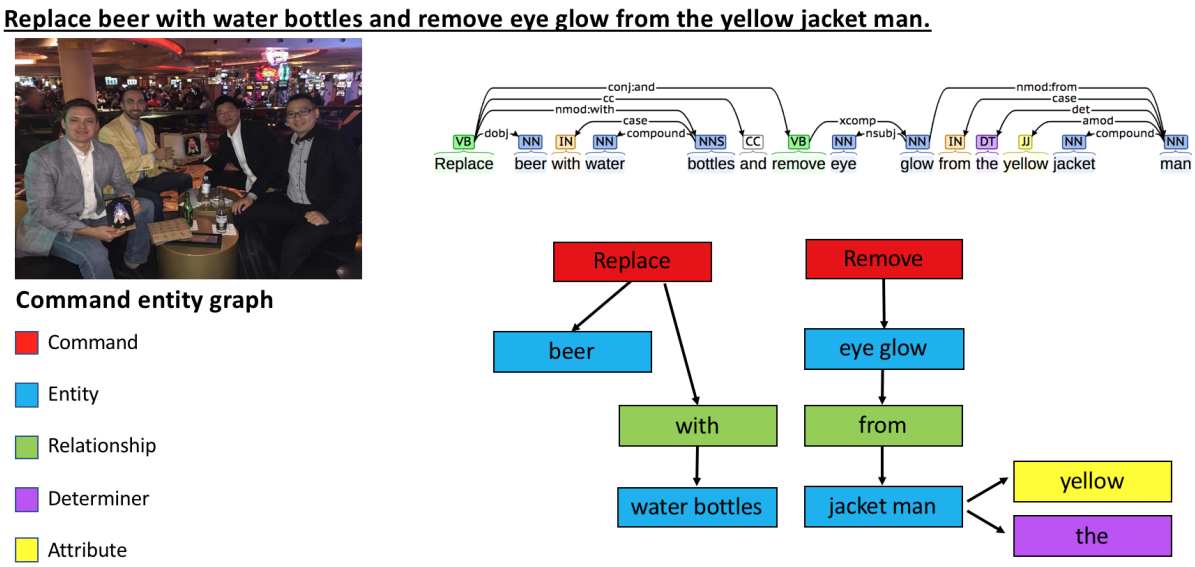

Figure 3.1: An illustration of the *command-entity representation* that is generated after the language and the corresponding dependencies are parsed.

Our proposed framework consists of several components as listed below:

- *Language parsing.* Our framework starts by passing the natural language input commands through an NLP parser, specifically the Stanford CoreNLP parser by Manning et. al [30]. Each input command must specify an image editing operation that acts either on the entire image (e.g "Change the background color to red.") or on specific entities grounded in the image (e.g "Remove the people next to the flag."). The language processing module transforms each sentence into a canonical command-entity representation using a set of manually crafted rules described later.

For this purpose, we first extract the command and associated entities (i.e. objects, their attributes, and their relationships) from the sentence using CoreNLP's Parts-of-Speech (PoS) tagger and Enhanced++ dependency annotator and convert the command and the respective entities into the *command-entity representation* representation.

- *Command-entity representation.* To be able to match between a text command and the corresponding SIMPL output we create a command-entity representation which contains objects or entities as nodes and their attributes and relationships as edges.

- *Descriptor Generation.* From each command-entity representation we output a series of *descriptors* which are essentially tuples of of the form (*verb*, *type*, *category*). Based on the dataset we manually create a series of templated descriptors which map to SIMPL commands. Each descriptor generated is matched to a specific template if found and the corresponding SIMPL output is generated using text replacement.

- *Mapping descriptors to SIMPL.* To map each templated descriptor we use a combination of hypernym and synonym matching from the WordNet [8] hierarchy. WordNet provides an exhaustive list of nouns, verbs, adjectives, and adverbs which are organized into sets of synonyms, each representing a lexicalized concept. These prove quite efficient to match each *category* in a newly generated descriptor to map to existing ones. Once a descriptor is matched, we use text replacement to replace the *category* with the entity's declaration in SIMPL and output the corresponding SIMPL commands.

3.1 Photoshop Request Dataset

In order to be able to analyze and build a framework we need a large dataset of commonly requested natural language queries covering a large variety of possible image editing actions. We source our data from the publicly available on-line forum called 'Photoshop Request' accessible at https://www.reddit.com/r/PhotoshopRequest/. The /r/PhotoshopRequest subreddit is a closely moderated forum where users of the website can create posts or threads presenting image content in the form of external hyper-links or directly pasted images and request edits to be made to them. Once posted, other users preferably graphic artists can execute the requested edits and post final edited image(s) as comments to the original posts. Edited posts can be approved by the original posters and are thereby marked as 'solved'. Each post is monitored for inappropriate content ensuring clean curated posts suitable for our purpose.

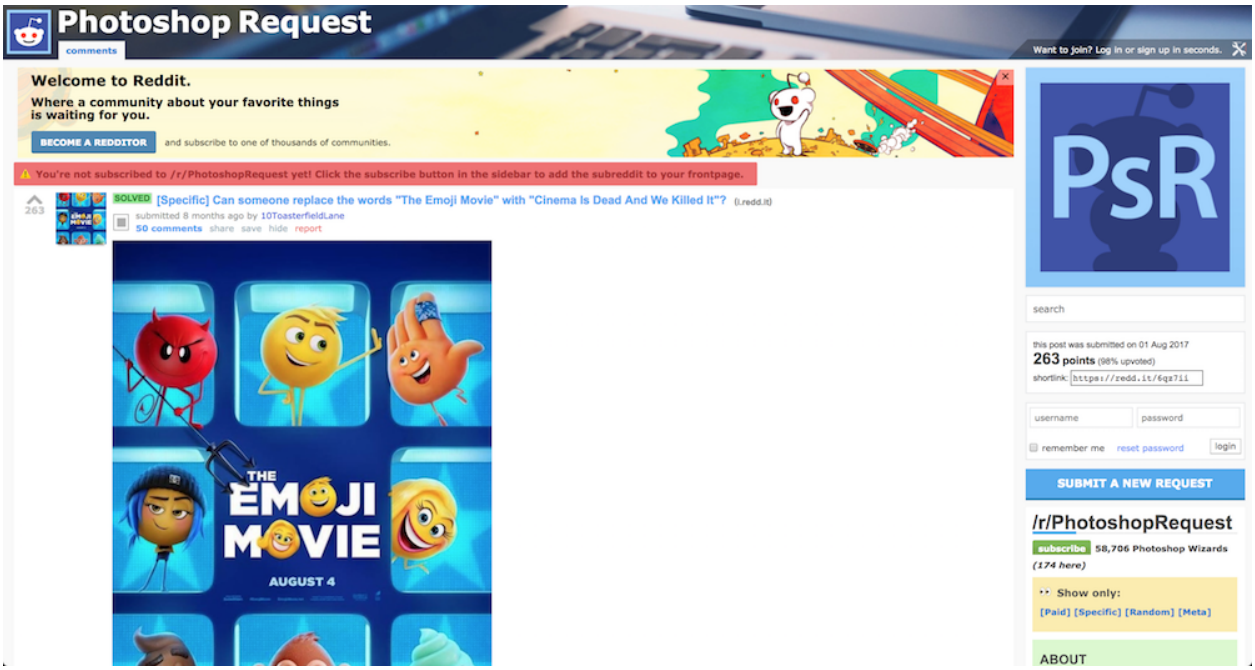

Figure 3.2: Screenshot of a post from the subreddit /r/PhotoshopRequest. Posts usually consist of a submission title and body with a free-form natural language editing request, original image(s) and comments with edited response image(s).

Users are encouraged to tag their posts as follows:

- [Specific] - When a user requests a specific edit.
  E.g. "[Specific] Can someone please remove the background?"
- [Random] - When a user leaves the request open-ended and encourages creativity.
  E.g. "[Random] Please do something funny with this pic"
- [Paid] - When a user is offering a reward for the edit.
  E.g. "[Paid] Can someone please fix this pic? $10 to the best submission"
- [Meta] - Used for sub-related threads.
  E.g. "[Meta] This sub is so cool!"

For building a dataset containing concrete editing requests we mainly focus on posts tagged as 'specific' and explore few 'random' posts for exposure to vague cases. Posts are crawled from the website thereby storing the submission title, original image(s) and edited image(s).

Chapter 4

Parsing

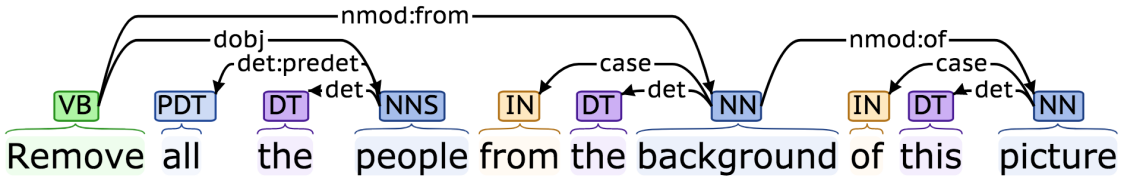

Figure 4.1: An example dependency parse using 'Enhanced++ Dependencies' from Stanford CoreNLP.
VB=verb, DT=determiner, NN(S)=(plural) noun, IN=preposition, PDT=predeterminer.

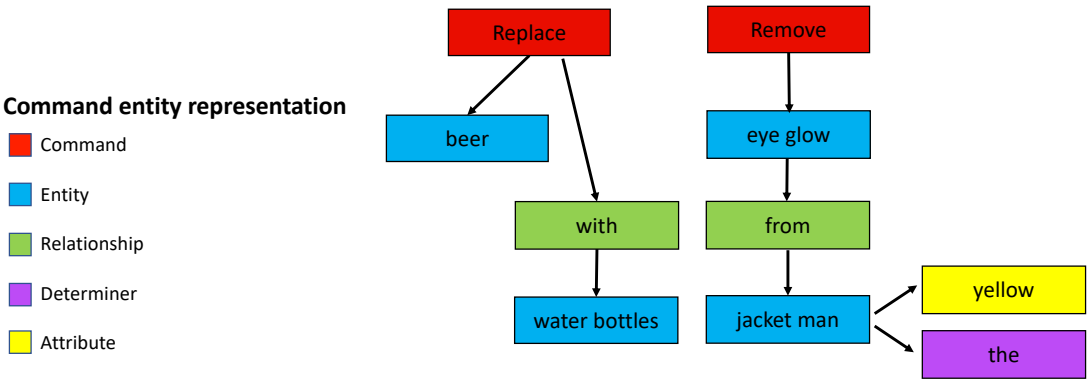

Figure 4.2: The sentence "Replace beer with water bottles and remove eye glow from the yellow jacket man." in our command entity representation.

As mentioned earlier we use the Stanford CoreNLP framework [30] to perform part-of-speech tagging and convert input statements into a dependency tree. This dependency tree assigns a parent token and annotation label to each token in the sentence; an example is shown in

the above figure. Parsing natural sentences is an inherently ambiguous process; reported on-corpus part-of-speech accuracy is 97% and dependency parsing accuracy is 91%.

In practice we found our parsing accuracy to be lower as our corpus is dissimilar from the newspaper articles the Stanford CoreNLP model was trained on.

4.1 Pre-processing

Each input statement goes through a series of pre-processing steps prior to the PoS tagging and dependency parsing. This step is necessary to reduce the possibility of mis-parses that frequently occur largely due to CoreNLP's training corpus being significantly different from our corpus. Mis-parses can lead to incorrect dependency parsings which in effect disrupt the entire pipeline. The different pre-processing steps are listed below:

- *Force verbs.* These are a list of frequently used words or phrases mostly in image editing parlance (e.g. 'change', 'clean', 'photoshop') that CoreNLP's PoS tagger fails to tag as verbs. We check for occurrences of these words early in the input statement and replace them with a temporary word that CoreNLP tags as a verb. This is required to ensure the appropriate dependency tree is generated. We curate a list of 15 words through our dataset.
- *Force nouns.* Similarly, these are a relatively small list of pronouns like 'me', 'they', 'him' etc. that are required to be tagged as nouns instead for accurate dependency parsing.
- *Stop verbs.* A list of words that occur as the first few words in a sentence are either incorrectly tagged as verbs or do not relate to image editing (e.g 'think', 'email', 'reward'). In such cases, we drop these words before they are passed onto the parser as input. A total of 55 stop verbs have been identified from the dataset.
- *Stop nouns.* These are words that are tagged as nouns but are rejected while creating the command entity representation to prevent them from being added as entities in the image (e.g. 'somebody', 'anybody').

4.2 Command-Entity representation

We seek to convert the low-level dependency shown in Fig. 4.1 into a list of entities annotated with attributes, relationships, determiners as well as a list of command verbs that operate over these entities. We refer to this as the *command-entity representation.* We use the term entity as opposed to object for cases where language refers to abstractions over group of objects, e.g. "the background".

Fig. 4.2 shows an example of our representation. We take all nouns in the sentences to be entities unless it is in a compound dependency with another noun (e.g. "photo" in "family photo") in which case the two are combined to represent a single entity or if the noun belongs in the stop noun list. We take all verbs in their base form to be commands unless they belong to the stop verb category.

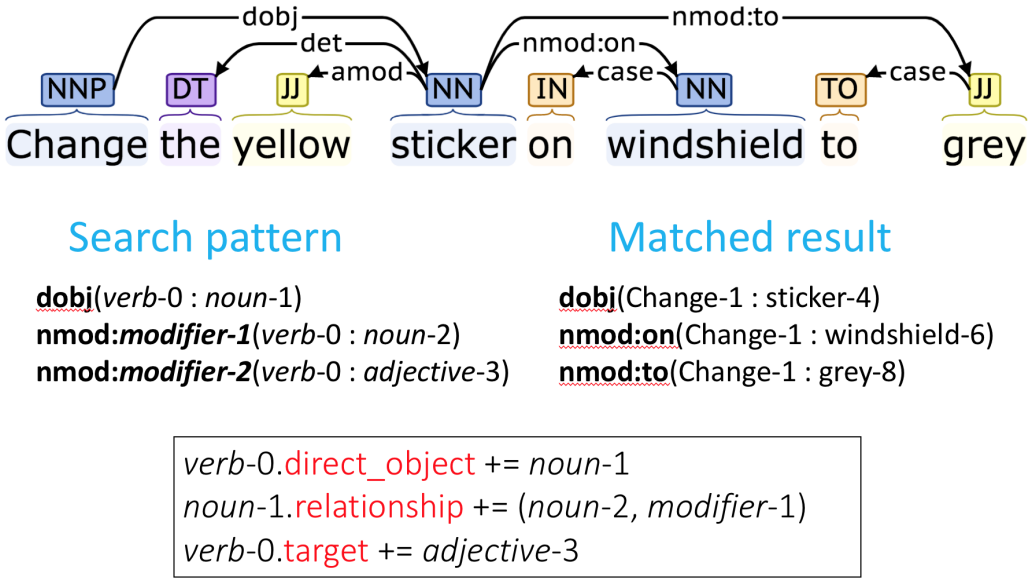

Figure 4.3: A pattern matching example. The entity (sticker) acted upon by the verb (Change) is added as a direct object along with relationships (on windshield) to other entities. The target adjective (yellow) is also added to the verb.

4.3 Pattern Matching

Once we determine the appropriate verbs and entities from the language input, we use pattern matching over the generated dependency parse tree to assign all other properties of the command-entity representation. Fig. 4.3 shows an example of the pattern matching process. We manually analyze close to 500 sentences and develop a set of 16 pattern matching rules such that we were able to correctly annotate all sentences. In some cases CoreNLP fails to produce reasonable dependency parse trees. We simply skip over such sentences while curating rules.

4.4 Descriptors

Based on the generated command-entity representation, we extract a series of *descriptors*. We define a *descriptor* as a tuple of the form (*verb*, *type*, *category*). *verb* refers to the command verb in it's base form, *type* refers to the connection between the verb and the category (e.g. 'direct object', 'target' etc.) and *category* is the specific entity being acted upon. For example, the sentence "Replace the dog in the image with a cat" produces the set of descriptors "(replace, direct_object, dog), (replace, target, cat)".

4.5 Template Matching

The final step in the pipeline requires mapping the extracted descriptors to executable commands in SIMPL. For this purpose we manually create an existing set of templates that map particular descriptors to their respective set of executable commands defined in SIMPL. The particulars of the defined commands are described in the next chapter.

Each descriptor in a template contains an associated *verb* like "Remove, Replace, Photoshop" as well as a *category* in the form of a high-level Synset taken from WordNet [8]. Synsets are unordered grouping of synonyms. Each *category* essentially represents an abstraction of particular entities. For example, in the case of "dog, cat, bear etc." the abstracted category in the templated descriptor is "animal". We devise an algorithm that performs a combination of synonym (exact or another similar word), hyponym (more specific meaning of a general term) and hypernym (broader meaning of a specific word) matching on descriptor sets to find matches to existing tempaltes. A specific descriptor like "(replace, direct_object, dog), (replace, target, cat)" would map to "(replace, direct_object, animal), (replace, target, animal)" Similarly, we create such descriptors for a wide variety of frequently occurring entities. Based on our preliminary observations we cover an extensive list of commonly used categories.

Chapter 5

Semantic Image Manipulation Language (SIMPL)

In this chapter we describe the details of our high-level Semantic Image Manipulation Language (SIMPL) for the task of image editing. We design the language to be fairly high-level (CISC) commands. The initial design came from a back-end representation for natural language requests. It is currently not intended to represent specific quantitative requests such as increase the brightness by 5%, focusing on qualitative representations.

5.1 Language Description

SIMPL is a strongly typed language that is represented in C++ style and is easy to read. A high-level description (with examples) of the types in the language and example constructors are listed below:

- Region - A weighted subset of pixels (mask) in a single image. It can represent the entire input image, an asset image from a database, or be derived by selection from an entity and image.

  ```
  Region r = Region(Entity("boy"), Image())
  ```

- Attribute - An attribute on an entity. Typically a single string, but sometimes can include modifiers.

  ```
  // "Add some very small coins to the chest"
  Attribute att = Attribute("small", modifiers={"very"})
  ```

- Relationship - A relationship from a base entity to a target entity, specified by a string representing the type of the relationship and the targeted entity.

- Entity - Entities represent nouns along with associated information such as attributes or relationships to other entities.

```
// "Delete the boy behind the blue bleachers"
Entity bleachers_7 = Entity("bleachers", determiner="the",
    attributes={"blue"})
Entity boy_3 = Entity("boy", determiner="the",relationships= {
    Relationship("behind", bleachers_7)})
```

- Color - A distribution over RGB colors, derived either from natural language or the color of another entity.

```
// "Make the girl have the same color shirt as the sign"
Color signColor = Color(Region(Entity("sign"), Image()))
```

- Font - A font, typically copied from existing text in the image.

5.2 Functions

5.2.1 Type Constructors

- `Attribute(string baseAttribute, list<string> modifiers)`

  The *baseAttribute* may be an adverb, adjective or any other property of an entity. *modifiers* are adjectival modifiers over the *baseAttribute* (e.g. "very", "more").

- `Relationship(string type, Entity targetEntity)`

  type is the string describing the relationship and *targetEntity* refers to the entity being referenced.

- `Entity(string nounPhrase, string determiner, list<Attribute> attributes, list<Relationship> relationships)`

  nounPhrase refers to the base noun bring referenced, *determiner* is the determiner used to reference the *nounPhrase*, *attributes* are all attributes applied to the entity and *relationships* reference all specified relationships to other entities.

- `Region(Entity e, Region r = Image())`

  Selects the given region in the image.

- `Color(string colorName)`

  Describes the color distribution implied by the color.

- `Color(Region r)`

  Extracts the color distribution of the selected region.

- `Font(Region r)`

  Captures the font specified in the given region, including typeface, color and style.

5.2.2 General-purpose Functions

- `Image()`

  Returns the whole image being edited in cases where there is only a single input image.

- `Image0() ... ImageN()`

  Returns the Nth image when there are multiple images as input.

- `FindAsset(Entity e)`

  Search an image corpus for the given entity. It may return a subset of the image as appropriate.

- `Foreground(Region r) / Background(Region r)`

  Selects the foreground or background of a given region.

- `Remove(Region r)`

  Delete the region and fill with a semantically-valid replacement (e.g content-aware fill or stock fill).

- `MakeTransparent(Region r)`

  Make the given region transparent.

- `ReColor(Region r, Color c)`

  Recolor the given region to a new specified color.

- Composite(Region dest, Region newContent)

  Adds the given content onto an appropriate scale and location on the destination region.

- Replace(Region start, Region newContent)

  Replace the given region with another region, potentially from another image, and deal with the content-fill and color harmony tasks.

- DrawFont(Font font, string text, Region location)

  Draws the given text with the given font at the region.

- ModifyLuminance(Region r, Attribute modifier)

  Modify the luminance of the given region using the given attribute. E.g. "bright", "dark". The provided attribute maybe relative or absolute.

- Modify(Region r, Attribute modifier)

  Modify the given region to have the given attribute.

- Fix(Region r)

  This particular function is used to handle sufficiently vague requests that cannot otherwise be specifically handled. It will ideally be dispatched based on the type of content being handled.

5.2.3 Specialized Tools

This is a non-exhaustive list of functions that map to specialized tools in an image editing tool.

- ColorEnhanceTool(Region r)

  Used to generate color enhancements to an image. E.g. "Please touch up the color on the manś shirt"

- EyeTool(Region r)

  For specialized enhancements to eyes such as red-eye correction, shut eye correction etc. E.g. "Please remove the flash from my eyes"

- FaceTool(Region r)

  For modifying facial features. E.g. "Please smooth out the acne on the face."

Chapter 6

Proof of Concept

6.1 Example Executions

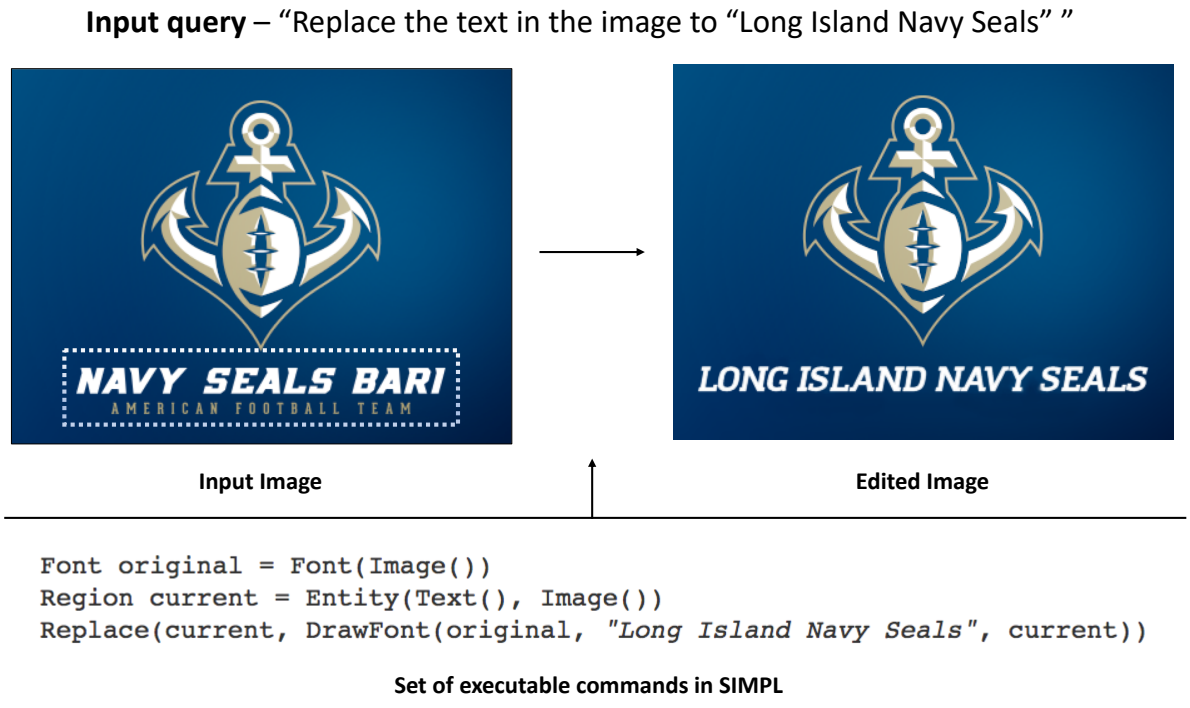

Figure 6.1: A sample execution of text replacement in an image while preserving the same font across the original and the final image.

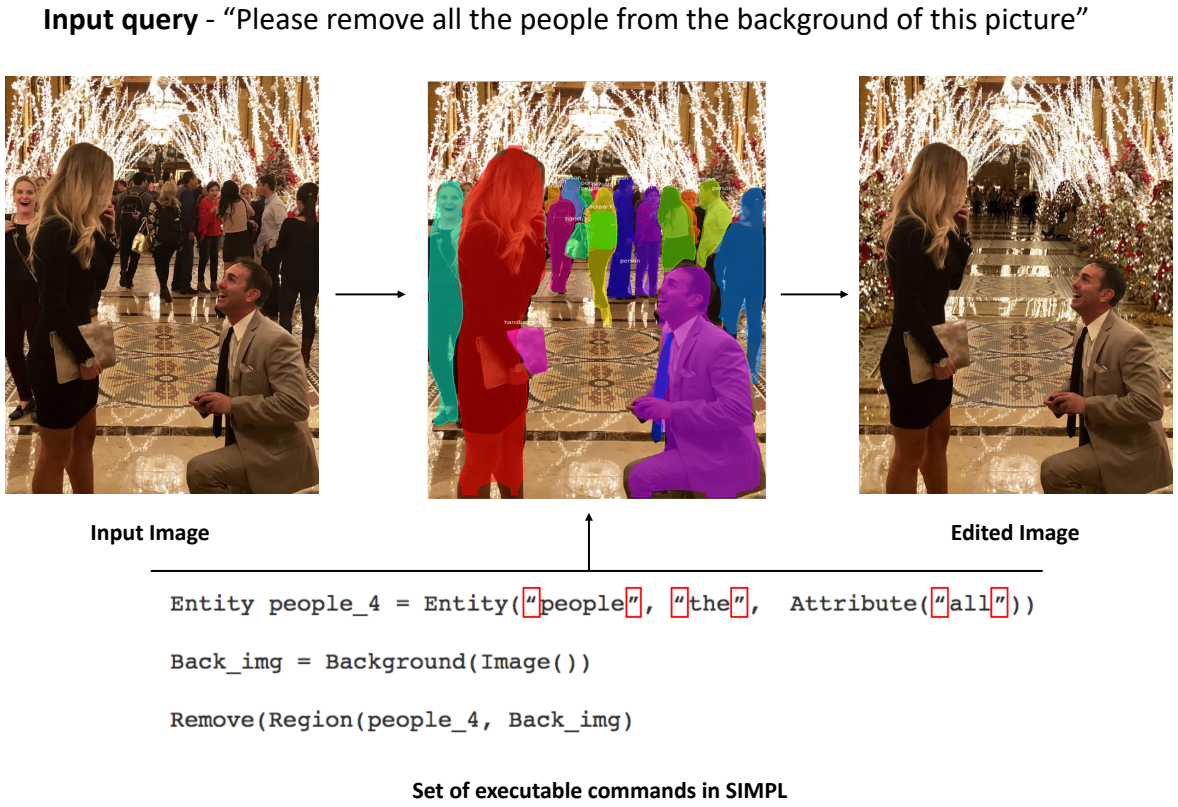

Figure 6.2: Sample execution of a complete edit execution using SIMPL commands. The masks in the center image were generated using an implementation of Mask R-CNN available online at https://github.com/matterport/Mask_RCNN.

Chapter 7

Role of Premises in Visual Question Answering

In this work, we make a simple but important observation – questions about images often contain *premises* – objects and relationships implied by the question – and that reasoning about premises can help Visual Question Answering (VQA) models respond more intelligently to irrelevant or previously unseen questions.

When presented with a question that is irrelevant to an image, state-of-the-art VQA models will still answer based purely on learned language biases, resulting in nonsensical or even misleading answers. We note that a visual question is irrelevant to an image if at least one of its premises is false (*i.e.* not depicted in the image). We leverage this observation to construct a dataset for Question Relevance Prediction and Explanation (QRPE) by searching for false premises. We train novel irrelevant question detection models and show that models that reason about premises consistently outperform models that do not.

We also find that forcing standard VQA models to reason about premises during training can lead to improvements on tasks requiring compositional reasoning.

7.1 Introduction

The task of providing natural language answers to free-form questions about an image – *i.e.* Visual Question Answering (VQA) – has received substantial attention in the past few years [28, 4, 29, 44, 17, 42, 24, 3, 23] and has quickly become a popular problem area. Despite significant progress on VQA benchmarks [4], current models still present a number of unintelligent and problematic characteristics.

When faced with questions that are irrelevant or not applicable for an image, current 'forced choice' models will still produce an answer. For example, given an image of a dog and the

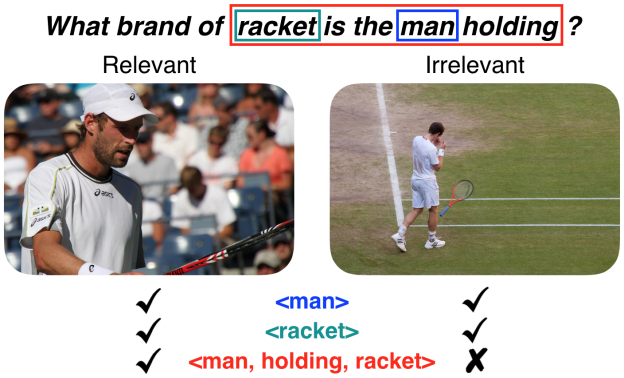

Figure 7.1: Questions asked about images often contains *'premises'* that imply visual semantics. From the above question, we can infer that a relevant image must contain a man, a racket, and that the man must be holding the racket. We extract these premises from visually grounded questions and use them to construct a new dataset and models for question relevance prediction. We also find that augmenting standard VQA training with simple premise-based questions yields improved performance on tasks requiring compositional reasoning.

question "*What color is the bird?*", standard VQA models might answer "*Red*" confidently, based only on language biases in the training set (*i.e.* an overabundance of red birds). In these cases, the predicted answers are senseless at best and misleading at worst, with either case posing serious problems for real-world applications. Like [34], we argue that practical VQA systems must be able to identify and explain irrelevant questions. For instance, a VQA model with this capability might answer "*There is no bird in the image*" when presented with this example question and image.

Premises. In this work, we make the observation that questions about images often contain *premises*, develop a premise extraction pipeline based on SPICE [2], and demonstrate how these premises can be used to address this shortcoming. Concretely, we define premises as facts implied by the language of questions, for example the question *"What brand of racket is the man holding?"* shown in Fig. 7.1 implies the existence of a man, a racket, and that the man is holding the racket. For visually grounded questions, *i.e.* questions asked about a particular image, these premises imply visual qualities of the image, including the presence of objects and their attributes and relationships.

Broadly speaking, we explore the usefulness of premises in two settings – when we know all visual questions are relevant to the images they are asked on (*e.g.* in the VQA dataset) and in real-life situations where such an assumption cannot be made (*e.g.* when asked by visually impaired users). In the former case, we show that knowing that a question is relevant allows us to perform data augmentation by creating additional simple question-answer pairs using the premises of source questions. In the latter case, we show that explicitly reasoning about

premises provides an effective and interpretable way of determining whether a question is relevant to an image.

Irrelevant Question Detection. We consider a question to be relevant to an image if the question's premises apply to the corresponding image *i.e.* the objects, attributes, and interactions implied by the question are depicted in the image. We refer to premises that apply for a given image to be true premises and those that do not apply as false premises. In order to train and evaluate models for this task, we curate a new irrelevant question detection dataset which we call the Question Relevance Prediction and Explanation (QRPE) dataset. QRPE is automatically curated from annotations already present in existing datasets, requiring no additional human supervision.

We collect the QRPE dataset by taking each image-question pair in the VQA dataset [4] and finding the most visually similar other image for which exactly one of the question premises is false. In this way, we collect triplets of two images and a question where the question is relevant for one image and not the other; moreover, the reason the question is irrelevant is known to be the single false premise. For context, the only other existing irrelevant question detection dataset [34] collected irrelevant question-image pairs by human verification of random pairs. In comparison, QRPE is substantially larger, balanced between irrelevant and relevant examples, and presents a considerably more difficult task due to the closeness of the image pairs both visually and with respect to question premises. We train novel models for irrelevant question detection on the QRPE dataset and compare to existing methods. In these experiments, we show that models that explicitly reason about question premises consistently outperform baseline models that do not.

Data Augmentation. Finally, we also introduce an approach to generate simple, templated question-answer pairs about elementary concepts from premises of complex training questions. In initial experiments, we show that adding these simple question-answer pairs to VQA training data can improves performance on tasks requiring compositional reasoning. These simple questions improve training by bringing implicit training concepts "to the surface", *i.e.* introducing direct supervision of important implicit concepts by transforming them to simple training pairs.

7.2 Related Work

Visual Question Answering: VQA has been the subject of a great deal of recent research attention [4, 11, 25]. Simple models like representing questions as bags of words (BoW+I [4]), or encoding the question using a recurrent neural network and train a simple classifier on the encoded question and image (Deeper LSTM [4]) have shown promise. Current top performing approaches include Neural Module Network (NMN [3]) , Hierarchical

Co-Attention (HieCoAtt [24]) and Multi Model Bilinear Pooling (MCB [10]). These models use principles like explicit compositional modules, soft hierarchical co-attention and MCB kernel for pooling multimodal features respectively.

Question Relevance: Most related to our work is [34], which introduced the task of irrelevant question detection for VQA. To evaluate on this task, they created the Visual True and False Question (VTFQ) dataset by pairing VQA questions with random VQA images and having human annotators verify whether or not the question was relevant. As a result, many of the irrelevant image-question pairs exhibit a complete mismatch of image and question content. Our Question Relevance Prediction and Explanation (QRPE) dataset on the other hand is collected such that irrelevant images for each question closely resemble the source image both visually and semantically. We also provide premise-level annotations which can be used to develop models that not only decide whether a question is relevant, but also provide explanations for *why* that is the case.

Semantic Tuple Extraction: Extracting structured facts in the form of semantic tuples from text is a well studied problem [38, 2, 7]; however, recent work has begun extending these techniques to visual domains [43, 14]. Additionally, the Visual Genome [18] dataset contains dense image annotations for objects and their attributes and relationships. However, we are the first to consider these facts to reason about question relevancy and compositionality in VQA.

Textual Entailment: The task of determining whether a hypothesis sentence is true, false, or neither based on some corpus is known as textual entailment and has seen substantial work in recent years [19, 5, 40, 31] including entailment based question answering systems [36, 37]. Generating premises can be viewed as a similar task, where given a corpus (a question in our case), the goal is to extract as many statements implied by the corpus as possible rather than verify a particular statement.

Compositionality: Recently, some effort has been seen in making VQA and other joint vision and language models like image captioning more compositional. This is still an open problem which has largely been approached from a model standpoint. For example, [12] integrate data and transfer knowledge between semantically related concepts, to improve upon current deep image captioning models. Similarly, [3] construct and learn neural module networks, which composes collections of jointly-trained neural modules into deep network for VQA. These approaches are different from our work since we prepose to improve compositionality via data augmentation.

Interpretable VQA Systems: Designing interpretable deep learning systems that provide rationales for their decisions is a topic that has received much attention of late. In ([33, 41], VQA models have been presented which can support their answers with explanations. However, there has not been work on generating explanations for false premise detection such as the approach proposed in this work.

Question Premise: To the best of our knowledge no other work has used Question Premises for improving VQA models. [13] however, comes close in this regard. This work proposes use of *basic questions* as a means to improve performance of VQA model.

However, basic questions are less complex questions that the main question can be dissociated into as opposed to premise questions, which are generated from implied facts or premises in the question. They also formulate basic question generation as a learning problem *i.e.* train a model on a collected dataset, while we use a templated rule based pipeline for premise question generation. Our work is also different in the aspect that we use premise questions to improve compositionality, question relevance prediction and explanation. [13] uses basic questions to improve accuracy on the VQA task by adding basic question features along with main question and image features to the classifier.

7.3 Premise Extraction

We now formalize the concept of premises and explain how premises can be extracted from questions. As discussed in 7.1, we define question premises as facts implied about an image from a question asked about it, which we represent as tuples. Returning to our running example question *"What brand of racket is the man holding?"*, we can express these premises as the tuples *'⟨man⟩'*, *'⟨racket⟩'*, and *'⟨man, holding, racket⟩'* respectively.

In this work, we categorize these tuples into three groups based on their complexity. First-order premises represent the presence of objects (*'⟨man⟩'*, *'⟨cat⟩'*, *'⟨sky⟩'*), second-order premises capture the attributes of objects (*'⟨man, tall⟩'*, *'⟨car, moving⟩'*), and third-order premises contain interactions between objects (*e.g. '⟨man, kicking, ball⟩'*, *'⟨cat, above, car⟩'*).

Premise Extraction: To extract premises from questions, we use the semantic tuple extraction pipeline used in the SPICE metric [2]. Originally defined as a metric for image captioning, SPICE transforms a sentence into a scene graph using the Stanford Scene Graph Parser [38] and then extracts semantic tuples from this representation. Fig. 7.2 shows this process for a sample question. The question is represented as a graph of objects, attributes, and relationships from which first, second, and third order premises are extracted respectively. As this pipeline was originally designed for descriptive captions rather than questions, we found a number of minor modifications helpful in extracting quality question premises,

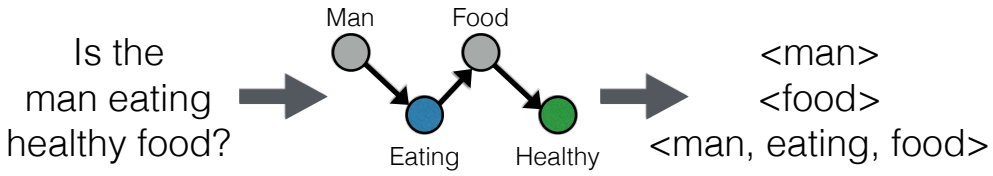

Figure 7.2: **Premise Extraction Pipeline.** Objects (gray), attributes (green), and relations (blue) scene graph nodes are converted into 1st, 2nd, and 3rd order premises respectively.

including disabling pronoun resolution, verb lemmatization and METEOR-based Synset matching. We will release our premise extraction code publicly to encourage reproducibility.

While this extraction process typically produces high quality premise tuples, there are some sources of noise which must be filtered out. The SPICE process occasionally produces duplicate nodes or object nodes not linked to nouns in the question, which we filter out. We also remove premises containing words like photo, image, *etc*. that refer to the image rather than its content.

A more nuanced source of erroneous premises comes from the ambiguity in existential questions, *i.e*. those about the existence of certain image content. For example, while the question *"Is the little girl moving?"* contains the premise '*⟨girl, little⟩*', it is unclear without the answer whether '*⟨girl, moving⟩*' is also a premise. Similarly, for the question *"How many giraffes are in the image?"*, '*⟨giraffe, many⟩*' cannot be considered a premise as there may be 0 giraffes in the image. To avoid introducing false premises, we filter out existential and counting questions which are answered negatively (either "no" or "0") if ground truth is available. Otherwise we threshold SPICE similarity between generated tuples and source questions to avoid repeating ambiguous premises.

7.4 Question Relevance Prediction and Explanation (QRPE) Dataset

Figure 7.3: **Some Examples from QRPE Dataset.** For a given question Q and a relevant image I^+, we find an irrelevant image I^- for which exactly one premise of the question is false. If there are multiple such candidates, we select the candidate most visually most similar to I^+. As can be seen from these examples, the QRPE dataset is very challenging, with only minor visual and semantic differences separating the relevant and irrelevant images.

As discussed in Section 7.1, modern VQA models fail to differentiate between relevant and irrelevant questions, answering either with confidence. This behavior is detrimental to the real world application of VQA systems. In this chapter, we curate a new dataset for question relevance in VQA which we call the Question Relevance Prediction and Explanation (QRPE) dataset. We plan to release QRPE publicly to help future efforts.

In order to train and evaluate models for irrelevant question detection, we would like to create a dataset of triplets (I^+, Q, I^-) comprised of a natural language question Q, an image I^+ for which Q is relevant, and an image I^- for which Q is irrelevant. While it is not required to collect both a relevant and irrelevant image for each question, we argue that doing so is a simple way to balance the dataset and it ensures that biases against rarer questions (which would be irrelevant for most images) cannot be exploited to inflate performance.

We base our dataset on the existing VQA corpus [4], taking the human-generated (and therefore relevant) image-question pairs from VQA as I^+ and Q. As previously discussed, we can define the relevancy of a question in terms of the grounding of its premises within an image, so we extract premises from each question Q and must find a suitable irrelevant image I^-. However, there are certainly many images for which one or more of Q's premises are false. One design decision is how to select I^- from this set.

To ensure our dataset is as realistic and challenging as possible, we consider irrelevant images which have only a single false question premise under Q. For example the question *"Is the big red dog old?"* could be matched with an image containing a big, white dog or a small red dog, but not a small white dog. In this way, we ensure that image content is semantically appropriate for the question topic but not quite relevant. Additionally, this provides each irrelevant image with an explanation for why the question does not apply.

Furthermore, we sort this subset of irrelevant image by their visual distance to the source image I^+ based on image encodings from a VGGNet [39] pre-trained on ImageNet [35]. This ensures that the relevant and irrelevant images are visually similar and act as difficult examples.

A major difficulty with our proposed data collection process is how to verify whether a premise if true or false for any given image in order to identify irrelevant images. We detail dataset construction and our approach for this problem in the following section.

7.4.1 Dataset Construction

We curate our QRPE dataset automatically from existing annotations in COCO and Visual Genome. For first order premises (*i.e.* existential premises), we consider only the 80 classes present in COCO [21]. As VQA and COCO share the same images, we can easily determine if a first order premise is true or false for a candidate irrelevant image simply by checking for the absence of the appropriate class annotation.

For second order premises (*i.e.* attributed objects), we rely on Visual Genome [18] annotations for object and attribute labels. Unlike in COCO, the lack of a particular object label in an image for Visual Genome does not necessarily indicate that the object is not present, due both to annotation noise and the use of multiple synonyms for objects by human labelers. As a consequence, we restrict the set of candidate irrelevant images to those which contain a matching object to the question premise but a different attribute. Without further restriction, the selected irrelevant attributes do not tend to be mutually exclusive with the source attribute (*i.e.* matching *'⟨dog, old⟩'* and *'⟨dog, red⟩'*). To correct this and ensure a false premise, we further restrict the set to attributes which are antonyms (*e.g.* *'⟨young⟩'* for source attribute *'⟨old⟩'*) or taxonomic sister terms (*e.g.* *'⟨green⟩'* for source attribute *'⟨red⟩'*) of the original premise attribute. We also experimented with third order premises; however, the lack of a corresponding sense of mutual exclusion for verbs and the sparsity of ⟨object, relationship, object⟩ premises made finding non-trivial irrelevant images difficult.

To recap, our data collection approach is then to take each image-question pair in the VQA dataset and extract its first and second order question premises. For each premise, we find all images which lack only this premise and rank them by their visual distance. The closest of these is kept as the irrelevant image for each image-question pair.

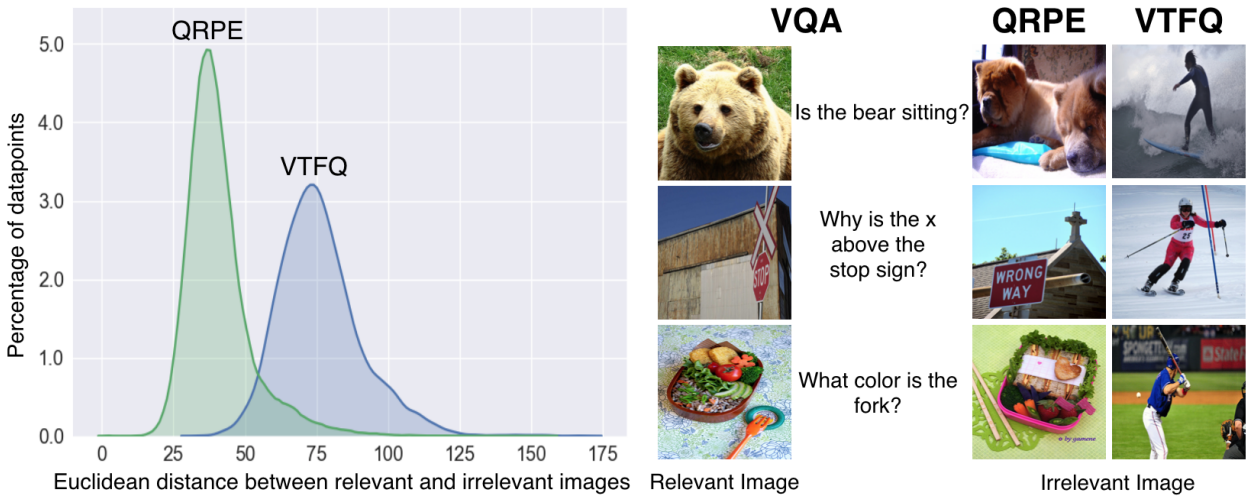

Figure 7.4: **A comparison of the QRPE and VTFQ Datasets.** On the left, we plot the Euclidean distance between VGGNet-fc7 features extracted from each relevant-irrelevant image pair for each dataset. Note that VTFQ has significantly higher visual distances. On the right, we show some qualitative examples of irrelevant images for questions that occur in both datasets. VTFQ images are significantly less related to the source image and question than in our dataset.

7.4.2 Exploring the Dataset

Fig. 7.3 shows sample (I^+, Q, I^-) triplets from our dataset along with the falsified premise that makes I^- irrelevant for Q. These examples illustrate the difficulty of our dataset. The images in the second column differ only because a red firetruck is not a red bus and the final column are differentiated only by the number of giraffe. Both of these are fine details of the image content.

The QRPE dataset contains 102,432 (I^+, Q, I^-) triplets generated from as many premises. In total, it contains 2961 unique premises and 96,812 unique questions. Among the 102,432 premises, 11,065 are second-order, attributed object premises while the remaining 91,367 are first-order object/scene premises. We divide our dataset into two parts – a training set with 68,037 premises that is generated from the VQA training set, and a validation set with 34,395 premises, generated from the VQA validation set.

7.4.3 Comparison to VTFQ

We contrast our approach to the VTFQ dataset of [34]. As discussed prior, VTFQ was collected by selecting a random question and image from the VQA set and asking human annotators to report if the question was relevant. This approach results in irrelevant image-question pairs that are unambiguously unrelated, with the visual content of the image having nothing at all to do with the question or its source image from VQA.

In Fig. 7.4, we present a quantitative and qualitative comparison of the two datasets. For the sake of comparison, we generate (I^+, Q, I^-) triplets from VTFQ by find the nearest neighbor question Q^{nn} in the VQA dataset to Q for each (Q, I^-) pair in VTFQ. We then select the image on which Q^{nn} was asked as I^+. We plot the Euclidean distance between the fc7 features of each (I^+, I^-) pair in both datasets. As shown on the left side of Fig. 7.3, we find that the mean distance in the VTFQ dataset is nearly twice that of our QRPE dataset, indicating that irrelevant images in VTFQ are less visually related to source images.

On the right side of Fig. 7.4, we also provide qualitative examples of questions that occur in both datasets. The example on the last row is perhaps most striking. The source question is asking the color of a fork and the relevant image shows an overhead view of a meal with an orange fork set nearby. The irrelevant image in QRPE is a similar image of food, but with chopsticks! Conversely, the image from VTFQ is a man playing baseball.

7.5 Question Relevance Detection

In this chapter, we introduce a simple baseline for irrelevant question detection on the QRPE dataset and demonstrate that explicitly reasoning about premises improves performance for both our new model and existing methods. More formally, we consider the binary classification task of predicting if a question Q_i from an image-question pair (I_i, Q_i) is relevant to image I_i.

A Simple Premise-Aware Model. Like the standard VQA task, question relevance detection also requires making a prediction based on an encoded image and question. With this in mind, we begin with a straight-forward approach based on the Deeper LSTM VQA model architecture of [4]. This model encodes the image I via a VGGNet and the question Q with an LSTM over one-hot word encodings. The concatenation of these embeddings are input to a multi-layer perceptron. We fine-tune this model for the binary question relevance detection task starting from a model pre-trained on the VQA task. We denote this model as `VQA-Bin`.

We extend the `VQA-Bin` model to explicitly reason about premises. We extract first and second order premises from the question Q and encode them as two concatenated one-

hot vectors. We add an additional LSTM to encode the premises and concatenate this added feature to the image and question feature. We refer to this premise-aware model as `VQA-Bin-Premise`.

Existing Methods. We compare our approaches with the best performing model of [34]. This model (which we call `QC-Sim`) uses NeuralTalk2 [16] trained on the MS COCO dataset [21] to generate a caption for each image and trains a multilayer perceptron to learn a similarity between LSTM embeddings (with shared weights) the question and generated caption (encoded as word2vec [32] representations). We consider two additional versions of this approach that consider premise-caption similarity (`PC-Sim`) and question-premise-caption similarities (`QPC-Sim`). We note that the caption similarity based methods makes use of significant outside data through NeuralTalk2 captioning and word2vec embeddings.

Models	Accuracy
VQA-Bin	65.9%
VQA-Bin-Prem	66.3%
QC-Sim	73.6%
PC-Sim	74.1%
QPC-Sim	**74.5**%

Table 7.1: Accuracy of Question Relevance models on the QRPE test set. We find that premise-aware models consistently outperform alternative models.

Results. We train each model on the QRPE train split and report results on the test set in Table 7.1. We find that the addition of extracted premise representations consistently improves performance of base models. This is especially interesting given that the models *already* have access to the question from which the premises were extracted. This result seems to imply there is value in explicitly isolating premises from sentence grammar.

We find that the caption similarity models significantly outperform our proposed VQA style models, with the overal best performing approach being premise augmented `QPC-Sim` model.

7.5.1 Question Relevance Explanation

In addition to identifying whether a question is irrelevant to an image, being able to indicate *why* carries significant real-world utility. From an interpretability perspective, reporting which premise is false is more informative than simply answering the question in the negative, as it can help to correct the questioner's misconception regarding the scene. We propose to generate such explanations by identifying the particular question premise(s) that do not apply to an image.

Figure 7.5: **Question relevance explanation:** We provide selected examples of predictions from the False Premise Detection model (FPD) on the QRPE test set. Reasoning about premises presents the opportunity to produce natural language statements indicating *why* a question is irrelevant to an image, by pointing to the premise that is invalid.

By construction, irrelevant images in the QRPE dataset are picked on the basis of negating a single premise – we now use our dataset to train models to detect false premises, and use the premises classified as irrelevant to generate templated natural language explanations.

Fig. 7.5 illustrates the task setup for false premise detection. Given a question-image pair, say *"Is the dog chasing the sheep?"*, the objective is to identify which (if any) question premises are not grounded in the image, in this case both *⟨dog⟩* and *⟨sheep⟩*. Alternatively, for the question *"What color are the four cones next to the truck?"*, both premises *⟨truck⟩* and *⟨cones, four⟩* are true premises grounded in the image.

For this task we train a binary classifier similar to which uses a one-hot encoding of premises (generates from the vocabulary of all premise words) and features from the last hidden layer of VGGNet [39] to represent the image. We concatenate these features and feed them into a multilayer perceptron which predicts whether the premise is grounded in the image or not. We trained our false premise detection model (FPD) model on all premises in the QRPE dataset.

Our FPD model achieves an accuracy of 60.8% on the QRPE dataset. In Fig. 7.5, we present qualitative results of our premise classification and explanation pipeline. For the question *"Is the dog chasing the sheep?"*, the model correctly recognizes 'dog' and 'sheep' as false premises, and we generate statements in natural language indicating the same. Thus, determining question relevance by reasoning about each premise presents the opportunity to generate simple explanations that can provide valuable feedback to the questioner, and help improve model trust.

7.6 Premise-Based Data Augmentation for VQA

In this chapter, we develop a premise-based data augmentation scheme for VQA that generates simple, templated questions based on premises present in complex visually-grounded questions from the VQA training set. As these question-image pairs were collected from sighted humans instructed to ask questions about a given an image, we assume that the questions (and the premises they imply) are grounded in the objects and relationships depicted in the corresponding image. We discuss question generation pipeline and various data augmentation experiments performed with these premise questions in Sections 7.6.1, 7.6.2 respectively.

7.6.1 Question Generation

Using the pipeline presented in 7.3, we extract premises from questions in the VQA dataset and apply a simple templated question generation strategy to transform premises into question and answer pairs. This process transforms implicit premise concepts which previously had to be learned as part of understanding more complex questions, into simple, explicit training examples that can be directly supervised. The generated premise questions cover a large number of objects, attributes, and relations, and as a result, including them in the VQA training set greatly expands and alters the answer space distribution. We designed specific templates for each type of premise.

Figure 7.6: **Question generation** For every source question, premise tuples are extracted and then used to generate premise questions using a rule-based NLP pipeline.

First order facts like ⟨man⟩, ⟨bus⟩ lead to existential questions like “Is there a man?”, “Is there a bus?” and so on. Second order facts can generate two kinds of questions depending on whether the second element is an action or an attribute. For example, ⟨man, walking⟩ would become “What is the man doing?” while ⟨car, red⟩ would become ”What is the color of the car?”. In general, questions generated from third order facts look like “Is the man

What will happen when the finger pushes the button?
What is the finger pushing? Button
Is there a finger in the image? Yes
Is there a button in the image? Yes

Has someone already eaten off the plate?
What is the someone eating off? Plate
Is there a someone in the image? Yes
Is there a plate in the image? Yes

What is the item called that the cat is looking at?
Is there a cat in the image? Yes
Is there an item in the image? Yes

What is the child sitting on?
What is the child doing? Sitting
Is there a child in the image? Yes

What player number is about to swing at the ball?
Is there a player number? Yes
Is there a number in the image? Yes
Is there a ball in the image? Yes

What is the man carrying in his left hand?
Who is carrying in the hand? Man
What is the man carrying in? Hand

Figure 7.7: Sample generated premise questions from source questions. Source questions are in bold. Ground-truth answers are extracted using the premise tuples.

holding the racket?", and "What is the cat on top of?" for ⟨man, holding, racket⟩ and ⟨cat, on top of, box⟩, respectively. However, third order facts are slightly more complicated and many kinds of questions can be generated from them depending on the types of elements present.

Question generation also involves minor pre-processing and post-processing. In the former we remove erroneous premises from the set while in the latter we remove generated questions which are linguistically ambiguous. We also run SPICE on the generated questions using the source questions as references to eliminate generated questions that are duplicates of the source questions. A random selection of premise questions generated from the VQA dataset can be seen in Fig. 7.7. The answer type distribution of generated premise questions can be seen in Table 7.2. We find that generated premise questions are twice in number as compared to source questions. It can also be seen that very few 'Number' questions are extracted. This is because people generally do not ask questions about multiple number of objects at the same time. By design, we generate only 'Yes' questions and zero 'No' questions. The reason for that is twofold – first, we only generate premise questions from true premises, and second, first order premises are the most frequent premises in source questions (first order premises generate 'Yes' questions).

7.6.2 Data Augmentation

Using the premise questions generated by employing the approach described in Section 7.6.1, we train VQA models on the augmented training set with various augmentation strategies

Training Data	Other	Number	Yes	No	Total
Source	123,817	29,698	57217	35842	246,574
Premise	137,483	1,850	387,941	0	527,274

Table 7.2: Answer type distribution of source and premise questions on the Compositional VQA train set.

based on different subsets of the premise questions. These ablations are described as follows:

1. **None:** No premise questions added to the training set.
2. **All:** Adding all the generated premise questions along with source questions to the training set.
3. **Only-Binary:** Only Binary (Questions with answers "Yes/No") premise questions added along with the source questions.
4. **No-Other:** All questions except the type Other premise questions (answers outside of Binary and Number answers) added to the training set.
5. **No-Binary:** All questions except Binary premise question types added to the training set.
6. **Comm-Other:** All Binary premise questions added. From Other and Number premise question types, selected premise questions are added whose answers lie in the source question answer space.
7. **Top1k-A:** All Binary premise questions added. From 'Others', selected premise questions are added whose answers are amongst the most top1000 VQA source responses.

Note that evaluation is performed on standard validation set containing no premise questions. We evaluate performance of VQA models trained with these various ablations on both - Standard VQA split as well as the Compostitional VQA split [1].

7.6.3 Results and Analysis

	Augmentation	Overall	Other	Number	Yes/No
Standard	None	54.23	40.34	33.27	79.82
	All	53.74	39.28	**33.38**	79.89
	Top-1k-A	**54.47**	**40.56**	33.24	**80.19**
Comp.	None	46.69	31.92	29.73	70.49
	All	47.63	31.97	**30.77**	**72.52**
	Top-1k-A	**47.85**	**32.58**	30.59	72.38

Table 7.3: Accuracy on the standard and compositional VQA validation sets for different augmentation strategies.

Standard VQA Split. We evaluate the augmentation settings described above with the Deeper LSTM/CNN VQA model by [22] on the Standard VQA split and show the results in the top half of Table 7.4. We find minor improvements of 0.34% using when restricting premise questions to match VQA answers (Top-1k-A).

Data Ablation	Overall	Other	Number	Yes/No
None	46.69	31.92	29.73	70.49
All	47.63	31.97	**30.77**	**72.52**
Only-Binary	47.25	32.45	29.65	71.30
No-Other	47.33	32.47	29.85	71.42
No-Binary	46.76	31.69	29.39	71.09
Comm-Other	47.53	32.41	28.88	72.33
Top1k-A	**47.85**	**32.58**	30.59	72.38

Table 7.4: Performance of DeeperLSTM [4] on Compositional VQA test split with different augmentations.

Compositional VQA Split. We also evaluate on a custom compositional VQA dataset split [1] that is specifically designed to test a model's ability to generalize to unseen/rarely seen combinations of concepts at test time. The bottom half of Table 7.4 shows results on this split. With our best ablation we observe a 1.16% gain in VQA accuracy over no augmentation. In this setting, explicitly reasoning about objects and attributes seen in the questions helps the model to disentangle objects from their most common characteristics. Some qualitative examples where an augmented model performs better than a non-augmented model are shown in Fig. 7.8.

Figure 7.8: Some interesting examples of how augmentation helps the DeeperLSTM model [4] on the compositional VQA split.

VQA Model	Baseline	With Premises
DeeperLSTM[4]	46.69	47.85
HieCoAtt[24]	50.17	49.98
NMN[3]	49.05	48.43
MCB[10]	50.13	50.57

Table 7.5: Accuracy of different VQA models on the Compositional VQA test split using Top1k-A augmentation.

We also train Deeper LSTM model on various other ablations of premise question and evaluate their performance. We observe that most of the ablations (Table 7.4) using premises show some improvement in performance over the model trained with no premise questions at all. Finally, we try replacing Deeper LSTM model with other VQA models to see if the performance boost is replicated. We observe that though data augmentation using premise questions help some models, there are other models for which that's not the case (Table 7.5).

Chapter 8

Conclusion

In Chapter 7, we made the simple observation that questions about images often contain premises implied by the question and showed that reasoning about premises can help VQA models respond more intelligently to irrelevant or previously unseen questions.

We develop a system for automatically extracting these question premises. Using extracted premises, we automatically created a novel dataset for Question Relevance Prediction and Explanation (QRPE) which consists of 102,432 question, relevant image, and irrelevant image triplets. We also train novel question relevance prediction models and show that models that take advantage of premise information outperform models that do not. Furthermore, we demonstrated that questions generated from premises may be an effective data augmentation techniques for VQA tasks that require compositional reasoning.

In Chapters 1 - 6, we proposed a novel framework to go from free-form natural language commands to performing fine-grained image edits. Figures 6.1 and 6.2 present a proof of concept showing executions from original images, their natural language query, the corresponding SIMPL commands and lastly the probable edited images. Such a high-level language does present limitations on the range of possible editing applications it can cover. The main limitations usually occur when a user does not define any specifics thereby leaving the execution open-ended. In such cases, we present the user with a list of most commonly used options to choose from. There are also cases when parsers fail to accurately annotate and parse the input language although that is not the primary focus of this work. We believe that an intermediate language approach is a more efficient approach at attempting to solve this problem acting as a bridge between natural language and the back-end algorithms used to execute image edits.

8.1 Future Work

Integrating Question Relevance Prediction and Explanation models with existing VQA systems would form a natural extension to VQA models. In this setting, the Relevance Prediction model would determine the applicability of a question to an image, and select an appropriate path of action. If the question is classified as relevant, the VQA model would generate a prediction; otherwise, a Question Relevance explanation model would provide a natural language sentence indicating which premise(s) are not valid for the image.

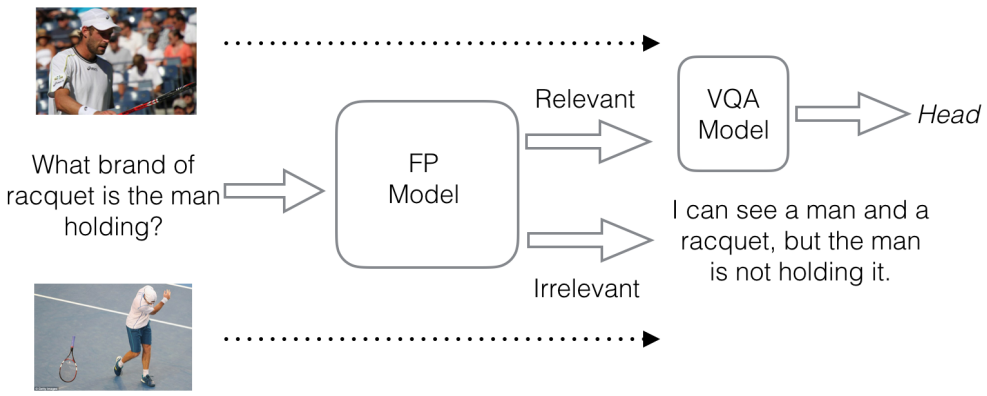

Figure 8.1: A complete VQA system that can additionally determine and explain the applicability of a question to an image.

Premises can also aid users in deciding when to place trust in the predictions from a VQA model. If a VQA model answers a question correctly, but is unable to correctly answer the simple premise questions (as described in 7.6), it would indicate that the model does not understand the underlying concepts in the question. However, being able to answer both source and premise questions correctly would suggest that the model understands underlying concepts, and thus the prediction can be trusted.

Such applications of question premises can help VQA systems to take a step in the direction of moving beyond academic settings to real-world environments.

The task of question relevance detection described in 7.5 could be naturally extended and applied to SIMPL where an input query could be validated for relevancy with the corresponding image input before processing the execution. A possible extension to the SIMPL framework would be developing and integrating an interface that can allow users to interact within an image editing environment. Such an interface can enhance the accessibility giving users the ability to directly manipulate the various parameters associated with images such as region selections, attribute modifications etc. Such a framework together with the significant progress being made in the field of computer vision and deep learning can eventually lead to solving such a challenging task.

Bibliography

[1] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi. C-vqa: A compositional split of the visual question answering (vqa) v1.0 dataset. *CoRR*, 2017.

[2] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

[3] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48, 2016.

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

[5] J. Bos and K. Markert. Recognising textual entailment with logical inference. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 628–635, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[6] J. Chen, Y. Shen, J. Gao, J. Liu, and X. Liu. Language-based image editing with recurrent attentive models. *arXiv preprint arXiv:1711.06288*, 2017.

[7] M. Elhoseiny, S. Cohen, W. Chang, B. Price, and A. Elgammal. Automatic annotation of structured facts in images. *arXiv preprint arXiv:1604.00466*, 2016.

[8] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[9] M. Fisher, M. Savva, and P. Hanrahan. Characterizing structural relationships in scenes using graph kernels. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 34:1–34:12, New York, NY, USA, 2011. ACM.

[10] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016.

[11] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu. Are you talking to a machine? dataset and methods for multilingual image question answering. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 2296–2304, Cambridge, MA, USA, 2015. MIT Press.

[12] L. A. Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell. Deep compositional captioning: Describing novel object categories without paired

training data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] J.-H. Huang, M. Alfadly, and B. Ghanem. VQABQ: Visual Question Answering by Basic Questions. *ArXiv e-prints*, March 2017.

[14] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *Conference on Computer Vision and Pattern Recognition*, pages 3668–3678, 2015.

[15] J. Johnson, R. Krishna, M. J. Stark, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, 2015.

[16] A. Karpathy and F.-F. Li. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[17] J.-H. Kim, S.-W. Lee, D. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang. Multimodal residual learning for visual qa. In *Advances in Neural Information Processing Systems*, pages 361–369, 2016.

[18] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

[19] A. Lai and J. Hockenmaier. Learning to predict denotational probabilities for modeling entailment. In *EACL*, 2017.

[20] G. P. Laput, M. Dontcheva, G. Wilensky, W. Chang, A. Agarwala, J. Linder, and E. Adar. Pixeltone: A multimodal interface for image editing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2185–2194, New York, NY, USA, 2013. ACM.

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

[22] J. Lu, X. Lin, D. Batra, and D. Parikh. Deeper lstm and normalized cnn visual question answering model. `https://github.com/VT-vision-lab/VQA_LSTM_CNN`, 2015.

[23] J. Lu, C. Xiong, D. Parikh, and R. Socher. Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning. *CVPR*, 2017.

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

[25] L. Ma, Z. Lu, and H. Li. Learning to answer questions from image using convolutional neural network. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 3567–3573. AAAI Press, 2016.

[26] A. Mahendru. *Role of Premises in Visual Question Answering*. PhD thesis, 2017.

[27] A. Mahendru, V. Prabhu, A. Mohapatra, D. Batra, and S. Lee. The promise of premise: Harnessing question premises in visual question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 926–935, Copenhagen, Denmark, September 2017. Association for Computational Lin-

guistics.

[28] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in Neural Information Processing Systems*, pages 1682–1690, 2014.

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

[30] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[31] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), 2014.

[32] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[33] D. H. Park, L. A. Hendricks, Z. Akata, B. Schiele, T. Darrell, and M. Rohrbach. Attentive explanations: Justifying decisions and pointing to the evidence. *arXiv preprint arXiv:1612.04757*, 2016.

[34] A. Ray, G. Christie, M. Bansal, D. Batra, and D. Parikh. Question relevance in vqa: Identifying non-visual and false-premise questions. *arXiv preprint arXiv:1606.06622*, 2016.

[35] O. Russakovsky, J. Deng, J. Krause, A. Berg, and L. Fei-Fei. The ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012). http://www.image-net.org/challenges/LSVRC/2012/, 2012.

[36] B. Sacaleanu, C. Orasan, C. Spurk, S. Ou, O. Ferrandez, M. Kouylekov, and M. Negri. Entailment-based question answering for structured data. In *22Nd International Conference on on Computational Linguistics: Demonstration Papers*, COLING '08, pages 173–176, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[37] M. Sammons, V. G. V. Vydiswaran, and D. Roth. "ask not what textual entailment can do for you...". In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1199–1208, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[38] S. Schuster, R. Krishna, A. Chang, L. Fei-Fei, and C. D. Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Workshop on Vision and Language (VL15)*, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[40] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. In *ICLR*, 2016.

[41] P. Wang, Q. Wu, C. Shen, A. v. d. Hengel, and A. Dick. Fvqa: Fact-based visual question answering. *arXiv preprint arXiv:1606.05433*, 2016.

[42] Q. Wu, C. Shen, L. Liu, A. Dick, and A. van den Hengel. What value do explicit high level concepts have in vision to language problems? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 203–212, 2016.

[43] D. Xu, Y. Zhu, C. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[44] C. L. Zitnick, A. Agrawal, S. Antol, M. Mitchell, D. Batra, and D. Parikh. Measuring machine intelligence through visual question answering. *arXiv preprint arXiv:1608.08716*, 2016.