

## ABSTRACT

Title of dissertation: VISION AND NATURAL LANGUAGE FOR  
CREATIVE APPLICATIONS  
AND THEIR ANALYSIS

Dissertation directed by: Professor Larry S. Davis,  
Department of Computer Science

Recent advances in machine learning, specifically problems in Computer Vision and Natural Language, have involved training deep neural networks with enormous amounts of data. The first frontier for deep networks was in uni-modal classification and detection problems (which were directed more towards "intelligent robotics" and surveillance applications), while the next wave involves deploying deep networks on more creative tasks and common-sense reasoning. We provide two applications of these, interspersed by an analysis on these deep models.

Automatic colorization is the process of adding color to greyscale images. We condition this process on language, allowing end users to manipulate a colorized image by feeding in different captions. We present two different architectures for language-conditioned colorization, both of which produce more accurate and plausible colorizations than a language-agnostic version. Through this language-based framework, we can dramatically alter colorizations by manipulating descriptive color words in captions.

Researchers have observed that Visual Question Answering(VQA) models tend

to answer questions by learning statistical biases in the data. (for example, the answer to the question “What is the color of the sky?” is usually “Blue”). It is of interest to the community to explicitly discover such biases, both for understanding the behavior of such models, and towards debugging them. In a database, we store the words of the question, answer and visual words corresponding to regions of interest in attention maps. By running simple rule mining algorithms on this database, we discover human-interpretable rules which give us great insight into the behavior of such models. Our results also show examples of unusual behaviors learned by the model in attempting VQA tasks.

Visual narrative is often a combination of explicit information and judicious omissions, relying on the viewer to supply missing details. In comics, most movements in time and space are hidden in the gutters between panels. To follow the story, readers logically connect panels together by inferring unseen actions through a process called closure. While computers can now describe what is explicitly depicted in natural images, in this paper we examine whether they can understand the closure-driven narratives conveyed by stylized artwork and dialogue in comic book panels. We construct a dataset, COMICS, that consists of over 1.2 million panels (120 GB) paired with automatic textbox transcriptions. An in-depth analysis of COMICS demonstrates that neither text nor image alone can tell a comic book story, so a computer must understand both modalities to keep up with the plot. We introduce three cloze-style tasks that ask models to predict narrative and character-centric aspects of a panel given  $n$  preceding panels as context. Various deep neural architectures underperform human baselines on these tasks, suggesting

that COMICS contains fundamental challenges for both vision and language.

For many NLP tasks, *ordered* models, which explicitly encode word order information, do not significantly outperform *unordered* (bag-of-words) models. One potential explanation is that the tasks themselves do not require word order to solve. To test whether this explanation is valid, we perform several time-controlled human experiments with scrambled language inputs. We compare human accuracies to those of both ordered and unordered neural models. Our results contradict the initial hypothesis, suggesting instead that humans may be less robust to word order variation than computers.

VISION AND NATURAL LANGUAGE FOR  
CREATIVE APPLICATIONS, AND THEIR ANALYSIS

by

Varun Manjunatha

Dissertation submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
2018

Advisory Committee:  
Professor Larry Davis, Chair/Advisor  
Professor Jordan Boyd-Graber  
Professor Rama Chellappa  
Professor Thomas Goldstein  
Professor David Jacobs

© Copyright by  
Varun Manjunatha  
2018



## Acknowledgments

First and foremost, I would like to thank my advisor Prof. Larry Davis, who provided me with freedom to pursue ideas which interested me, patience when I was stuck and guidance when I was frustrated. His warmth, generosity, humor and humanity made this thesis possible.

I would like to thank Prof. David Jacobs and Prof. Rama Chellappa from whom I learned the fundamentals of computer vision as a young graduate student. Thanks are due to Prof. Jordan Boyd-Graber for guiding me through, and critiquing several of my publications and Prof. Tom Goldstein for agreeing to peruse through my thesis and appear on my committee.

My interactions with Prof. Srikumar Ramalingam, Dr. Anbumani Subramanian, Prof. Devi Parikh, Dr. Tim Marks and Dr. Kuan-Chuan Peng at various stages of my career have been instrumental to my growth as a researcher. Not only are they inspirational, but from each of them, I have learned critical techniques which I use even now on a daily basis.

I would like to profusely thank Mohit, Yogarshi and Anupam - colleagues from the Computational Linguistics and Information Processing laboratory (CLIP) at Maryland - who were vital to my research. I shall forever cherish our 3 AM coding sessions. I would also like to thank my labmates - Sravanthi, Bharat, Sungmin, Xianzhi, Jason, Venkat and Nirat for fruitful discussions and their friendship. I would like to thank Jennifer Story and Tom Hurst of the Computer Science graduate office at Maryland who have put up with many annoying requests, and Prof. V.S

Subrahmanian who was on my proposal committee. Finally, I would like to thank my former roommates for many years - Parth, Vamsi, Srinivas, Ankit and Sohil for helping me through the ups and downs of graduate life.

A doctoral degree taxes not only the student, but also those who love him the most. Therefore, no quantity of words can be sufficient to thank my parents, my brother and his family, who believed in me more than I believed in myself. This thesis is dedicated to them.



# Table of Contents

|   |     |
|---|-----|
| Acknowledgements  | ii  |
| List of Tables  | vi  |
| List of Figures   | vii |
| List of Abbreviations   | xi  |
| 1 Introduction  | 1   |
| 1.1 Creative Deep Networks                                    | 1   |
| 1.2 Analyzing Deep Networks                                   | 3   |
| 2 Learning to Color from Language                             | 4   |
| 2.1 Overview  | 4   |
| 2.2 Model   | 6   |
| 2.2.1 Images and color spaces                                 | 6   |
| 2.2.2 Fully-convolutional networks for colorization           | 6   |
| 2.2.3 Colorization conditioned on language                    | 7   |
| 2.3 Experiments   | 10  |
| 2.3.1 Experimental setup                                      | 10  |
| 2.3.2 Human experiments                                       | 11  |
| 2.4 Discussion  | 12  |
| 2.5 Future Work   | 13  |
| 3 Explicit Bias Discovery in Visual Question Answering Models | 17  |
| 3.1 Overview  | 17  |
| 3.2 Background and Related Work                               | 20  |
| 3.3 Method  | 21  |
| 3.3.1 Baseline Model  | 23  |
| 3.3.2 Visual Codebook Generation                              | 23  |
| 3.3.3 From attention map to bounding box                      | 24  |
| 3.3.4 Pipeline Summarized                                     | 25  |
| 3.4 Experiments   | 26  |

|         |   |    |
|---------|---|----|
| 3.4.1   | Language only statistical biases in VQA . . . . .                     | 26 |
| 3.4.2   | Vision+Language statistical biases in VQA . . . . .                   | 26 |
| 3.4.2.1 | What time? . . . . .  | 28 |
| 3.4.2.2 | Why? . . . . .  | 28 |
| 3.4.2.3 | What is he/she doing? . . . . .                                       | 29 |
| 3.5     | Limitations and Summary . . . . .                                     | 30 |
| 4       | The Amazing Mysteries of the Gutter:                                  |    |
|         | Drawing Inferences Between Panels in Comic Book Narratives . . . . .  | 38 |
| 4.1     | Overview . . . . .  | 38 |
| 4.2     | Creating a dataset of comic books . . . . .                           | 40 |
| 4.2.1   | Where do our comics come from? . . . . .                              | 42 |
| 4.2.2   | Breaking comics into their basic elements . . . . .                   | 43 |
| 4.2.3   | OCR . . . . .   | 45 |
| 4.3     | Data Analysis . . . . .   | 45 |
| 4.4     | Tasks that test closure . . . . .                                     | 48 |
| 4.4.1   | Task Difficulty . . . . .   | 51 |
| 4.5     | Models & Experiments . . . . .  | 52 |
| 4.5.1   | Model definitions . . . . .   | 53 |
| 4.6     | Error Analysis . . . . .  | 56 |
| 4.7     | Related Work . . . . .  | 58 |
| 4.8     | Summary & Future Work . . . . .                                       | 60 |
| 5       | The Effect of Word Scrambling Across Natural Language Tasks . . . . . | 62 |
| 5.1     | Overview . . . . .  | 62 |
| 5.2     | Experiments . . . . .   | 64 |
| 5.2.1   | Human Timing Experiments . . . . .                                    | 66 |
| 5.2.2   | Neural Network Experiments . . . . .                                  | 66 |
| 5.3     | Discussion . . . . .  | 67 |
| 5.3.1   | Scrambling Degrades Human Accuracy . . . . .                          | 67 |
| 5.3.2   | Time Limits Degrade Scrambled Task Accuracy . . . . .                 | 69 |
| 5.3.3   | Implications for Human Processing . . . . .                           | 69 |
| 5.4     | Summary . . . . .   | 72 |
| 6       | Conclusion . . . . .  | 73 |
|         | Bibliography . . . . .  | 75 |

## List of Tables

|     |  |    |
|-----|--|----|
| 2.1 | While <b>FILM</b> is the most accurate model in <i>ab</i> space, its outputs are about as contextually plausible as <b>CONCAT</b> 's according to our <i>plausibility</i> task, which asks workers to choose which model's output best depicts a given caption (however, both models significantly outperform the language-agnostic <b>FCNN</b> ). This additional plausibility does not degrade the output, as shown by our <i>quality</i> task, which asks workers to distinguish an automatically-colored image from a real one. Finally, our caption <i>manipulation</i> experiment, in which workers are guided by a caption to select one of three outputs generated with varying color words, shows that modifying the caption significantly affects the outputs of <b>CONCAT</b> and <b>FILM</b> . . . . . | 8  |
| 3.1 | We run a language-only VQA baseline and note that although only 40% of the questions are answered correctly in VQA 2.0, a large number of questions (88%) in our experiments are answered with plausibly correct responses. For example, "Sunglasses" would be a perfectly plausible answer to the question "What does that girl have on her face?" - perhaps even more so than the ground-truth answer ("Nothing"). The <b>last example</b> shows an implausible answer provided by the model to the question. . . . .  | 27 |
| 4.1 | Statistics describing dataset size (top) and the number of total instances for each of our three tasks (bottom). . . . .   | 42 |
| 4.2 | Combining image and text in neural architectures improves their ability to predict the next image or dialogue in <b>COMICS</b> narratives. The contextual information present in preceding panels is useful for all tasks: the model that only looks at a single panel ( <b>NC-image-text</b> ) always underperforms its context-aware counterpart. However, even the best performing models lag well behind humans. . . . .   | 56 |
| 5.1 | Accuracy for the five tasks for human participants and neural models.  | 64 |

## List of Figures

|     |  |    |
|-----|--|----|
| 2.1 | Three pairs of images whose colorizations are conditioned on corresponding captions by our <b>FILM</b> architecture. Our model can localize objects mentioned by the captions and properly color them. . . . .   | 5  |
| 2.2 | <b>FILM</b> applies feature-wise affine transformations (conditioned on language) to the output of each convolutional block in our architecture.   | 9  |
| 2.3 | The top row contains successes from our caption manipulation task generated by <b>FILM</b> and <b>CONCAT</b> , respectively. The second row shows examples of how captions guide <b>FILM</b> to produce more accurate colorizations than <b>FCNN</b> (failure cases outlined in red). The final row contains, from left to right, particularly eye-catching colorizations from both <b>CONCAT</b> and <b>FILM</b> , a case where <b>FILM</b> fails to localize properly, and an image whose unnatural caption causes artifacts in <b>CONCAT</b> . . . . .  | 15 |
| 2.4 | Examples of intermediate layer activations while generating colorized images using the <b>FILM</b> network. These activation maps correspond to the mean activation immediately after the <b>FILM</b> layers of the sixth, seventh, and eighth blocks. Interestingly, the activations after the <b>FILM</b> layer of Block 6 always seems to focus on the object that is to be colorized, while those of Block 8 focus almost exclusively on the background. The activation maps do not significantly differ when color words in the caption are manipulated; therefore, we show maps only for the first color word in these examples. . . . . | 16 |

|     |  |    |
|-----|--|----|
| 3.1 | In Figure 1 (left), we show examples of two questions in VQA which the model requires a “skill” to answer (such as telling the time, or reading), and a third which can be answered using statistical biases in the data. On the right, we show examples of statistical biases which lead a model to answer “4” (referred to as <i>consequents</i> ), given a set of questions containing the phrase “How many?” and various visual elements ( <i>antecedents</i> ). Note that each row in this figure represents multiple questions in the VQA validation set. This particular instance of the trained VQA model seems to have learned that giraffes and chairs have four legs, stop signs have four letters, and kitchen stoves have four burners. The * next to the answer reminds us that it is from the set of answer words. Upon inspection, we found 33 questions (out of >200k) in the VQA validation set which contain the words {How,many,burners} and the most common answer predicted by our model for these is 4 (which also resembles the ground-truth distribution). However, some of them were along the lines of “How many burners are turned on?”, which led to answers different from “4” . . . . . | 31 |
| 3.2 | The model from [Kazemi and Elqursh, 2017] tries to answer the question ”Which dessert are you tempted to try?”. In doing so, the visual attention focuses on a region of the image which contains donuts. We use the method by [Chen et al., 2016] to place a bounding box over this region, which maps to a distinct visual word representing <i>donuts</i> in our vocabulary. Our database of items thus contains all of the words of the question, the visual word and the answer words. Rules are then extracted using the Apriori algorithm [Agrawal and Srikant, 1994] . . . . .   | 32 |
| 3.3 | We show visual code-words generated by the method of Section 3.1. In the first (left-most) column, we notice visual code-words corresponding to objects or patches in MSCOCO, but in the latter two columns (on the right) we notice code-words corresponding to more complex visual concepts like “people eating”, “women in bridal-wear” or “black-and-white tennis photographs” . . . . .   | 32 |
| 3.4 | In the first example, critical to answering the question correctly is discovering the presence of a fence (shown in red) in the attention heat-map. The cropping method of Chen et al. [2016] places a conservative box over this region, which corresponds to net-like or fence-like visual code-words like a tennis-net or a baseball batting-cage in the visual codebook. Similarly, in the second example, the attention corresponds to a visual code-word which clearly depicts boats, and in the third example, the attention corresponds to the teddy-bear code-word. . . . .   | 33 |

|     |   |    |
|-----|---|----|
| 3.5 | <b>What time?</b> : Rule 1 shows kite-flying during the daytime, whereas rule 2 shows traffic lights during night. “What time?” asked about an image containing a clock prompts the model to guess a random hour of the day (rule 3). The fall season seems to be associated with a visual word depicting leafless trees (rule 4). . . . .  | 34 |
| 3.6 | <b>Why?</b> : Rules that exceeded the support threshold indicate that arms are outstretched for balance (rule 1), umbrellas protect one from rain and provide shade (rules 2-3), and that fences, orange (vests) and helmets lead to safety (rules 5-7). . . . .  | 34 |
| 3.7 | <b>What is he/she doing?</b> : The rules in this table show standard activities in the VQA (and MSCOCO) datasets like skateboarding, snowboarding, flying a kite, playing frisbee, etc. We observed a difference in diversity of rules for male (he,man,boy) and female pronouns (she,woman,girl,lady) even at very low support. This indicates that the VQA , or more likely, the MSCOCO datasets are unintentionally skewed in terms of gender. . . . .   | 35 |
| 3.8 | <b>What brand?</b> : The VQA model seems to have learned that the Wilson brand is related to tennis, Dell and Apple make laptop computers and that Jetblue is a “brand” of airline. The visual similarity between old models of Nokia phones and TV remotes explains rule 5. Interestingly, rule 7, which pertains to “What brand of soda?” does not have an accompanying visual word. This indicates either that the model has not learned to disambiguate between various soda brands, or that our rule finding method has failed to learn of such a disambiguating rule. . . . . | 36 |
| 3.9 | <b>Where?</b> : The model of Kazemi and Elqursh [2017] has learned that clocks often appear on facades of buildings, elephants are from Africa, aircraft can be found in airports and that buses are found in the downtown of a city . . . . .  | 37 |
| 4.1 | Where did the snake in the last panel come from? Why is it biting the man? Is the man in the second panel the same as the man in the first panel? To answer these questions, readers form a larger meaning out of the narration boxes, speech bubbles, and artwork by applying closure across panels. . . . .   | 39 |
| 4.2 | Different artistic renderings of lions taken from the <b>COMICS</b> dataset. The left-facing lions are more cartoonish (and humorous) than the ones facing right, which come from action and adventure comics that rely on realism to provide thrills. . . . .  | 41 |

|     |  |    |
|-----|--|----|
| 4.3 | Five example panel sequences from <b>COMICS</b> , one for each type of interpanel transition. Individual panel borders are color-coded to match their intrapanel categories (legend in bottom-left). Moment-to-moment transitions unfold like frames in a movie, while scene-to-scene transitions are loosely strung together by narrative boxes. Percentages are the relative prevalence of the transition or panel type in an annotated subset of <b>COMICS</b> . . . . .  | 46 |
| 4.4 | In the character coherence task (top), a model must order the dialogues in the final panel, while visual cloze (bottom) requires choosing the image of the panel that follows the given context. For visualization purposes, we show the original context panels; during model training and evaluation, textboxes are blacked out in every panel. . . . .  | 49 |
| 4.5 | The <b>image-text</b> architecture applied to an instance of the <i>text cloze</i> task. Pretrained image features are combined with learned text features in a hierarchical LSTM architecture to form a context representation, which is then used to score text candidates. . . . .  | 52 |
| 4.6 | Three <i>text cloze</i> examples from the development set, shown with a single panel of context (boxed candidates are predictions by the <b>text-image</b> model). The airplane artwork in the top row helps the <b>image-text</b> model choose the correct answer, while the <b>text-only</b> model fails because the dialogue lacks contextual information. Conversely, the bottom two rows show the <b>image-text</b> model ignoring the context in favor of choosing a candidate that mentions something visually present in the last panel. . . . . | 61 |
| 5.1 | A sample task from CLEVR with a scrambled question about an image. The question is highly ambiguous: the answer depends on whether the cube is red or gray. The original question is "What shape is the small red object to the right of small gray cube?" . . . . .   | 63 |
| 5.2 | In the CLEVR task, scrambled sentences lead to ambiguities in understanding the question. In shorter questions, however, one may expect there to be fewer ambiguities compared to longer ones. This is shown empirically by a gap between human unscrambled and scrambled performance that widens with length of the question. . . . .   | 68 |

## List of Abbreviations

|        |  |
|--------|--|
| AI     | Artificial Intelligence                                |
| CLEVR  | Compositional Language and Elementary Visual Reasoning |
| CNN    | Convolutional Neural Network                           |
| CV     | Computer Vision  |
| DCM    | Digital Comics Museum                                  |
| DNN    | Deep Neural Network                                    |
| FCNN   | Fully Convolutional Neural Network                     |
| FILM   | Feature-wise Linear Modulation                         |
| GRU    | Gated Recurrent Unit                                   |
| LIME   | Local Interpretable Model-Agnostic Explanations        |
| LSTM   | Long Short Term Memory                                 |
| ML     | Machine Learning                                       |
| MSCOCO | Microsoft Common Objects in Context                    |
| NLP    | Natural Language Processing                            |
| OCR    | Optical Character Recognition                          |
| QI     | Question-Image   |
| QA     | Question-Answer  |
| QI+A   | Question-Image+Answer                                  |
| RCNN   | Region-based Convolutional Neural Networks             |
| SNLI   | Stanford Natural Language Inference                    |
| SQUAD  | Stanford Question Answering Dataset                    |
| SST    | Stanford Sentiment Treebank                            |
| VGG    | Visual Geometry Group                                  |
| VGQA   | Visual Genome Question Answering                       |
| VQA    | Visual Question Answering                              |



## Chapter 1: Introduction

Research in Artificial Intelligence(AI) formally began at the Dartmouth Summer Research Project on Artificial Intelligence in 1956. Since then, the field has gone through ebbs and flows, but we are currently at the forefront of its greatest crest - the era of Deep Neural Networks (DNNs). The fields of Machine Learning (ML), Computer Vision (CV), Natural Language Processing (NLP) and Speech Recognition, amongst others, have been completely revolutionized and dominated by deep architectures. Although neural networks conceptually predate the Dartmouth conference [[Hebb, 1949](#), [McCulloch and Pitts, 1943](#)], it was not until the emergence of Graphical Processing Units (GPUs) and internet scale data that Neural Networks became house-hold technology (this is no exaggeration, for an Amazon Echo device is a home device that relies on Deep Networks for its speech recognition).

### 1.1 Creative Deep Networks

In recent years, Deep Networks has been deployed towards solving non-traditional tasks, which previously required the expertise of a creatively skilled human. Excellent examples are Generative Adversarial Networks[[Goodfellow et al., 2014](#), [Isola et al., 2017](#)], or Image Style Transfer network by [Gatys et al. \[2016\]](#), both of which

can be used to imitate artwork in the style of any Renaissance master. Before this technology existed, one would at best have to employ a skilled art student to do the same. In this thesis, we explore and analyze applications of AI in such creative ventures.

First, we cover the task of image colorization, the art of applying colors to black and white images. Those well versed in the film industry are all too aware that attempts to colorize black and white movies are extremely tedious projects. The work of [Zhang et al. \[2016\]](#) showed that even this task is possible for an end-to-end deep neural network, which takes as input greyscale images and is trained to spit out the respective color images. The intuition behind this work is that patches that correspond to specific objects, like stop signs, or backgrounds, like grass, tend to have fixed colors throughout a large dataset (red and green, respectively). But what about cars or trains or dogs, which occur in nature in a wider variety of colors? The solution we propose in Chapter 2 uses language to uniquely disambiguate objects that can occur in different colors.

A second aspect of Deep Learning is that it serves as a useful tool for social scientists and researchers in the digital humanities. AI has already been used to analyze novels [[Iyyer et al., 2016](#)] and narratives [[Huang et al., 2016a](#)], but a unique confluence of Vision and Language occurs in the analysis of comic books. To understand one panel and predict what might happen in the next, an understanding of both artwork and text is required. In Chapter 4 of this thesis, we create the largest English language dataset of comic books and develop novel tasks and architectures to predict the contents of a future panel, given  $n$  preceding panels.

## 1.2 Analyzing Deep Networks

In the previous section, we heavily alluded to the fact that Deep Networks learn to solve tasks by ingesting large quantities of data. These architectures can be somewhat opaque and complex (i.e., black-boxes), and thus far, are excellent at imitation but not so at reasoning. An important unsolved problem for the research community is Visual Question Answering (VQA), in which a Deep Network answers a question about an image. On popular datasets like [Antol et al. \[2015\]](#) and [Goyal et al. \[2017\]](#), the state of the art model, [\[Teney et al., 2018\]](#) obtains 70% accuracy, while a strong baseline [\[Kazemi and Elqursh, 2017\]](#) obtains around 60% accuracy. This gives one an impression that deep networks are actually intelligent, but this isn't so - they merely have the veneer of intelligence. We show in Chapter 3 empirically that these models answer questions not by logic or reasoning, but by correlating key words in the question and visual elements in the image, with answer words.

Finally, a curious observation in NLP is that models which do not take into account the order of words in a sentence often perform similarly to models which do [\[Iyyer et al., 2015\]](#). This is significant because *unordered* models are simpler and faster than *ordered* ones. To explore this phenomenon further, in Chapter 5 we explore the effect of word order on both humans and machines on a diverse set of Vision and NLP tasks.

## Chapter 2: Learning to Color from Language

### 2.1 Overview

Automatic image colorization [Zhang et al., 2016, Cheng et al., 2015, Larsson et al., 2016, Iizuka et al., 2016, Deshpande et al., 2017]—the process of adding color to a greyscale image—is inherently underspecified. Unlike background scenery such as sky or grass, many common foreground objects could plausibly be of any color, such as a person’s clothing, a bird’s feathers, or the exterior of a car. Interactive colorization seeks human input, usually in the form of clicks or strokes on the image with a selected color, to reduce these ambiguities [Levin et al., 2004, Huang et al., 2005, Endo et al., 2016, Zhang et al., 2017]. We introduce the task of colorization from natural language, a previously unexplored source of color specifications.

Many use cases for automatic colorization involve images paired with language. For example, comic book artwork is normally first sketched in black-and-white by a penciller; afterwards, a colorist selects a palette that thematically reinforces the written script to produce the final colored art. Similarly, older black-and-white films are often colorized for modern audiences based on cues from dialogue and narration [Van Camp, 1995].

Language is a weaker source of supervision for colorization than user clicks. In

particular, language lacks ground-truth information about the colored image (e.g., the exact color of a pixel or region). Given a description like *a blue motorcycle parked next to a fleet of sedans*, an automatic colorization system must first localize the motorcycle within the image before deciding on a context-appropriate shade of blue to color it with. The challenge grows with abstract language: a red color palette likely suits an artistic rendering of *the boy threw down his toy in a rage* better than it does *the boy lovingly hugged his toy*.



Figure 2.1: Three pairs of images whose colorizations are conditioned on corresponding captions by our **FILM** architecture. Our model can localize objects mentioned by the captions and properly color them.

We present two neural architectures for language-based colorization that augment an existing fully-convolutional model [Zhang et al., 2016] with representations learned from image captions. As a sanity check, both architectures outperform a language-agnostic model on an accuracy-based colorization metric. However, we are more interested in whether modifications to the caption properly manifest themselves in output colorizations (e.g., switching one color with another); crowdsourced evaluations confirm that our models properly localize and color objects based on captions (Figure 2.1).

## 2.2 Model

This section provides a quick introduction to color spaces (Sec. 2.2.1) and then describes our baseline colorization network (Sec. 2.2.2) alongside two models (Sec. 2.2.3) that colorize their output on representations learned from language.

### 2.2.1 Images and color spaces

An image is usually represented as a three dimensional tensor with red, green and blue (RGB) channels. Each pixel’s color and intensity (i.e., lightness) are *jointly* represented by the values of these three channels. However, in applications such as colorization, it is more convenient to use representations that separately encode lightness and color. These *color spaces* can be obtained through mathematical transformations of the RGB color space; in this work, following Zhang et al. [2016], we use the CIE *Lab* space [Smith and Guild, 1931]. Here, the first channel ( $L$ ) encodes only lightness (i.e., black-and-white). The two color channels  $a$  and  $b$  represent color values between green to red and blue to yellow, respectively. In this formulation, the task of colorization is equivalent to taking the lightness channel of an image as input and predicting the two missing color channels.

### 2.2.2 Fully-convolutional networks for colorization

Following Zhang et al. [2016], we treat colorization as a classification problem in CIE *Lab* space: given only the lightness channel  $L$  of an image (i.e., a greyscale version), a fully-convolutional network predicts values for the two color channels  $a$

and  $b$ . For efficiency, we deviate from Zhang et al. [2016] by quantizing the color channels into a  $25 \times 25$  grid, which results in 625 labels for classification. To further speed up training, we use a one-hot encoding for the  $ab$  channels instead of soft targets as in Zhang et al. [2016]; preliminary experiments showed no qualitative difference in colorization quality with one-hot targets. The contribution of each label to the loss is downweighted by a factor inversely proportional to its frequency in the training set, which prevents desaturated  $ab$  values. Our baseline network architecture (**FCNN**) consists of eight convolutional blocks, each of which contains multiple convolutional layers followed by batch normalization [Ioffe and Szegedy, 2015].<sup>1</sup> Next, we propose two ways to integrate additional *text* input into **FCNN**.

### 2.2.3 Colorization conditioned on language

Given an image  $I$  paired with a unit of text  $T$ , we first encode  $T$  into a continuous representation  $h$  using the last hidden state of a bi-directional LSTM [Hochreiter and Schmidhuber, 1997]. We integrate  $h$  into every convolutional block of the **FCNN**, allowing language to influence the computation of all intermediate feature maps.

Specifically, say  $\mathbf{Z}_n$  is the feature map of the  $n$ th convolutional block. A conceptually simple way to incorporate language into this feature map is to concatenate  $h$  to the channels at each spatial location  $i, j$  in  $\mathbf{Z}_n$ , forming a new feature map

$$\mathbf{Z}'_{n_{i,j}} = [\mathbf{Z}_{n_{i,j}}; h]. \quad (2.1)$$

---

<sup>1</sup>See Zhang et al. [2016] for complete architectural details. Code and pretrained models are available at <https://github.com/superhans/colorfromlanguage>.

| <b>Model</b>  | <b><i>ab</i> Accuracy</b> |             | <b>Human Experiments</b> |             |             |
|---------------|---------------------------|-------------|--------------------------|-------------|-------------|
|               | acc@1                     | acc@5       | plaus.                   | qual.       | manip.      |
| <b>FCNN</b>   | 15.4                      | 45.8        | 20.4                     | 32.6        | N/A         |
| <b>CONCAT</b> | 17.9                      | 50.3        | 39.0                     | <b>34.1</b> | 77.4        |
| <b>FILM</b>   | <b>23.7</b>               | <b>60.5</b> | <b>40.6</b>              | 32.1        | <b>81.2</b> |

Table 2.1: While **FILM** is the most accurate model in *ab* space, its outputs are about as contextually plausible as **CONCAT**’s according to our *plausibility* task, which asks workers to choose which model’s output best depicts a given caption (however, both models significantly outperform the language-agnostic **FCNN**). This additional plausibility does not degrade the output, as shown by our *quality* task, which asks workers to distinguish an automatically-colored image from a real one. Finally, our caption *manipulation* experiment, in which workers are guided by a caption to select one of three outputs generated with varying color words, shows that modifying the caption significantly affects the outputs of **CONCAT** and **FILM**.

While this method of integrating language with images (**CONCAT**) has been successfully used for other vision and language tasks [Reed et al., 2016, Feichtenhofer et al., 2016], it requires considerably more parameters than the **FCNN** due to the additional language channels.

Inspired by recent work on visual question answering, we also experiment with a less parameter-hungry approach, feature-wise linear modulation [Perez et al., 2018, **FILM**], to fuse the language and visual representations. Since the activations of **FILM** layers have attention-like properties when trained on VQA, we also might



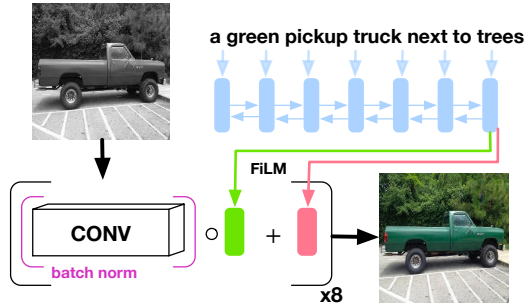


Figure 2.2: **FILM** applies feature-wise affine transformations (conditioned on language) to the output of each convolutional block in our architecture.

expect **FILM** to be better at localizing objects from language than **CONCAT** on colorization (see Figure 2.4 for heatmap visualizations).

**FILM** applies a feature-wise affine transformation to the output of each convolutional block, where the transformation weights are conditioned on language (Figure 2.2). Given  $\mathbf{Z}_n$  and  $h$ , we first compute two vectors  $\gamma_n$  and  $\beta_n$  through linear projection,

$$\gamma_n = \mathbf{W}_{n_\gamma} h \quad \beta_n = \mathbf{W}_{n_\beta} h, \quad (2.2)$$

where  $\mathbf{W}_{n_\gamma}$  and  $\mathbf{W}_{n_\beta}$  are learned weight matrices. The modulated feature map then becomes

$$\mathbf{Z}'_{n_{i,j}} = (1 + \gamma_n) \circ \mathbf{Z}_{n_{i,j}} + \beta_n, \quad (2.3)$$

where  $\circ$  denotes the element-wise product. Compared to **CONCAT**, **FILM** is parameter-efficient, requiring just two additional weight matrices per feature map.

## 2.3 Experiments

We evaluate **FCNN**, **CONCAT**, and **FILM** using accuracy in *ab* space (shown by Zhang et al. [2016] to be a poor substitute for plausibility) and with crowdsourced experiments that ask workers to judge colorization *plausibility*, *quality*, and the colorization flexibly reflects language *manipulations*. Table 2.1 summarizes our results; while there is no clear winner between **FILM** and **CONCAT**, both rely on language to produce higher-quality colorizations than those generated by **FCNN**.

### 2.3.1 Experimental setup

We train all of our models on the 82,783 images in the MSCOCO [Lin et al., 2014] training set, each of which is paired with five crowdsourced captions. Training from scratch on MSCOCO results in poor quality colorizations due to a combination of not enough data and increased image complexity compared to ImageNet [Russakovsky et al., 2015]. Thus, for our final models, we initialize all convolutional layers with a **FCNN** pretrained on ImageNet; we finetune both **FILM** and **CONCAT**'s convolutional weights during training. To automatically evaluate the models, we compute top-1 and top-5 accuracy in our quantized *ab* output space<sup>2</sup> on the MSCOCO validation set. While **FILM** achieves the highest *ab* accuracy, **FILM** and **CONCAT** do not significantly differ on crowdsourced evaluation metrics.

---

<sup>2</sup>We evaluate accuracy at the downsampled 56×56 resolution at which our network predicts colorizations. For human experiments, the prediction is upsampled to 224×224.

### 2.3.2 Human experiments

We run three human evaluations of our models on the Crowdflower platform to evaluate their plausibility, overall quality, and how well they condition their output on language. Each evaluation is run using a random subset of 100 caption/image pairs from the MSCOCO validation set,<sup>3</sup> and we obtain five judgments per pair.

*Plausibility given caption:* We show workers a caption along with three images generated by **FCNN**, **CONCAT**, and **FILM**. They choose the image that best depicts the caption; if multiple images accurately depict the caption, we ask them to choose the most realistic. **FCNN** does not receive the caption as input, so it makes sense that its output is only chosen 20% of the time; there is no significant difference between **CONCAT** and **FILM** in plausibility given the caption.

*Colorization quality:* Workers receive a pair of images, a ground-truth MSCOCO image and a generated output from one of our three architectures, and are asked to choose the image that was *not* colored by a computer. The goal is to fool workers into selecting the generated images; the “fooling rates” for all three architectures are comparable, which indicates that we do not reduce colorization quality by conditioning on language.

*Caption manipulation:* Our last evaluation measures how much influence the caption has on the **CONCAT** and **FILM** models. We generate three different col-

---

<sup>3</sup>We only evaluate on captions that contain one of ten “color” words (e.g., **red**, **blue** purple).

orizations of a single image by swapping out different colors in the caption (e.g., *blue car, red car, green car*). Then, we provide workers with a single caption (e.g., *green car*) and ask them to choose which image best depicts the caption. If our models cannot localize and color the appropriate object, workers will be unable to select an appropriate image. Fortunately, **CONCAT** and **FILM** are both robust to caption manipulations (Table 2.1).

## 2.4 Discussion

Both **CONCAT** and **FILM** can manipulate image color from captions (further supported by the top row of Figure 2.3). Here, we qualitatively examine model outputs and identify potential directions for improvement.

Language-conditioned colorization depends on correspondences between language and color statistics (*stop signs* are always red, and *school buses* are always yellow). While this extra information helps us produce more plausible colorizations compared to language-agnostic models (second row of Figure 2.3), it biases models trained on natural images against unnatural colorizations. For example, the yellow sky produced by **CONCAT** in the bottom right of Figure 2.3 contains blue artifacts because skies are usually blue in MSCOCO. Additionally, our models are limited by the lightness channel  $L$  of the greyscale image, which prevents dramatic color shifts like black-to-white. Smaller objects are also problematic; often, colors will “leak” into smaller objects from larger ones, as shown by **FILM**’s colorizations of purple plants (Figure 2.3, bottom-middle) and yellow tires (middle-left).

Figure 2.4 shows activation maps from intermediate layers generated while colorizing images using the **FILM** network. Each intermediate layer is captured immediately after the **FILM** layer and is of dimension  $h \times w \times c$  (e.g.,  $112 \times 112 \times 64$ ,  $28 \times 28 \times 512$ , etc.), where  $h$  is the height of the feature map,  $w$  is its width, and  $c$  is the number of channels.<sup>4</sup> On inspection, the first few activation maps correspond to edges and are not visually interesting. However, we notice that the sixth activation map usually focuses on the principal subject of the image (such as a car or a horse), while the eighth activation map focused everywhere but on that subject (i.e., entirely on the background). This analysis demonstrates that the **FILM** layer emulates visual attention, reinforcing similar observations on visual QA datasets by Perez et al. [2018].

## 2.5 Future Work

While these experiments are promising, that there are many avenues to improve language-conditioned colorization. From a vision perspective, we would like to more accurately colorize parts of objects (e.g., a person’s shoes); moving to more complex architectures such as variational autoencoders [Deshpande et al., 2017] or PixelCNNs [Guadarrama et al., 2017] might help here, as could increasing training image resolution. We also plan on using refinement networks [Shrivastava et al., 2017] to correct for artifacts in the colorized output image. On the language side, moving from explicitly specified colors to abstract or emotional language is a par-

---

<sup>4</sup>We compute the mean across the  $c$  dimension and scale the resulting  $h \times w$  feature map between the limits  $[0, 255]$ .

ticularly interesting. We plan to train our models on dialogue/image pairs from datasets such as COMICS [Iyyer et al., 2017] and visual storytelling [Huang et al., 2016a]; these models could also help learn powerful joint representations of vision and language to improve performance on downstream prediction tasks.

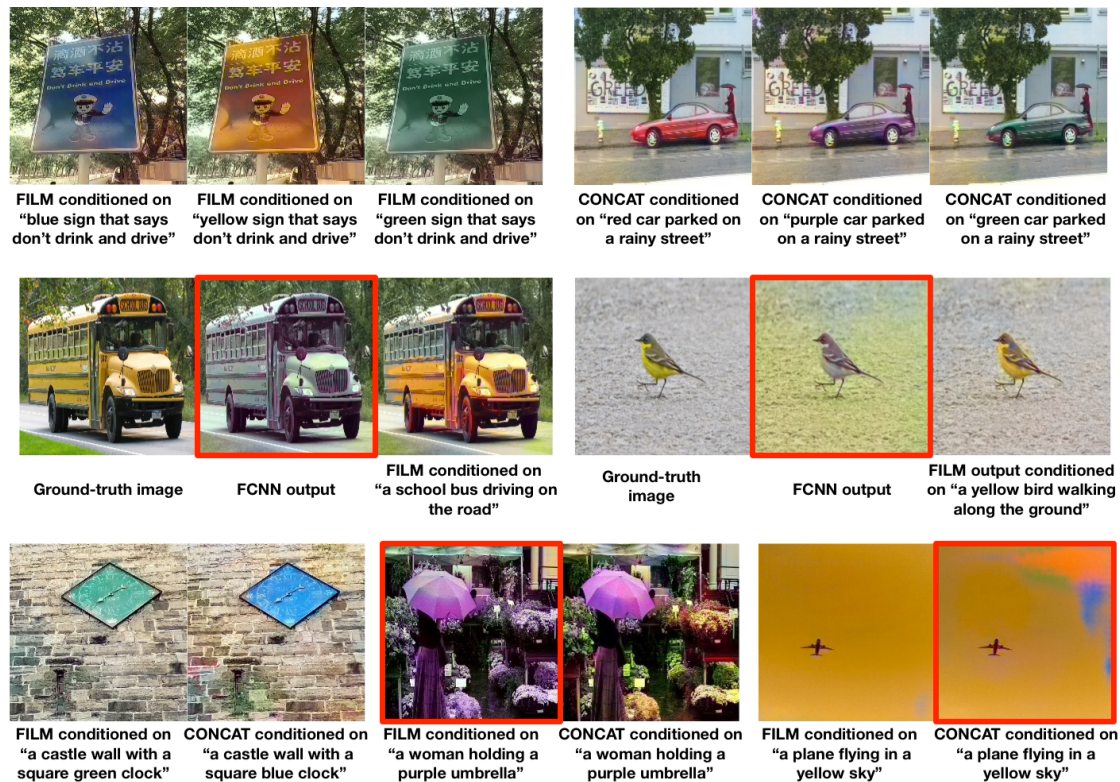


Figure 2.3: The top row contains successes from our caption manipulation task generated by **FILM** and **CONCAT**, respectively. The second row shows examples of how captions guide **FILM** to produce more accurate colorizations than **FCNN** (failure cases outlined in red). The final row contains, from left to right, particularly eye-catching colorizations from both **CONCAT** and **FILM**, a case where **FILM** fails to localize properly, and an image whose unnatural caption causes artifacts in **CONCAT**.

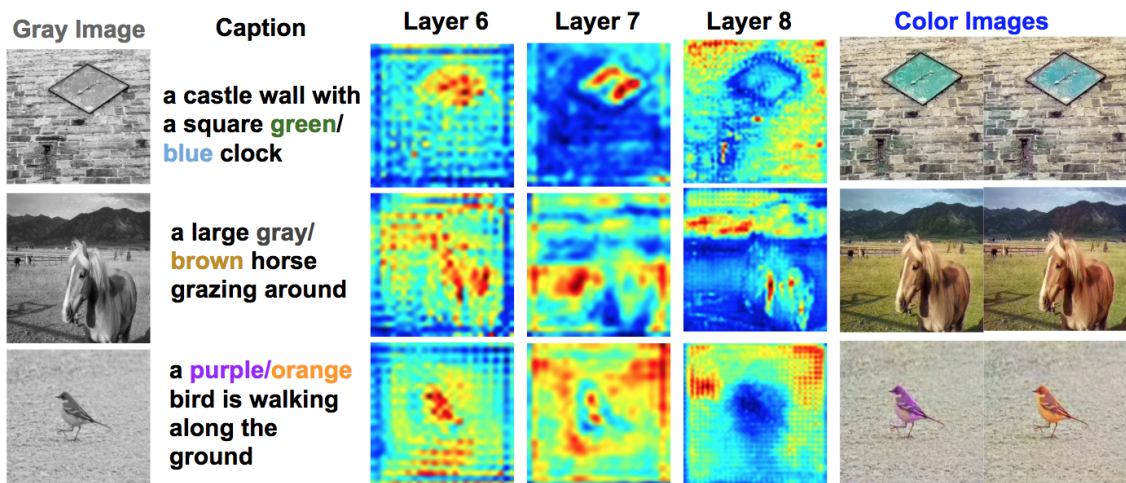


Figure 2.4: Examples of intermediate layer activations while generating colorized images using the **FILM** network. These activation maps correspond to the mean activation immediately after the **FILM** layers of the sixth, seventh, and eighth blocks. Interestingly, the activations after the **FILM** layer of Block 6 always seems to focus on the object that is to be colorized, while those of Block 8 focus almost exclusively on the background. The activation maps do not significantly differ when color words in the caption are manipulated; therefore, we show maps only for the first color word in these examples.



## Chapter 3: Explicit Bias Discovery in Visual Question Answering Models

### 3.1 Overview

In recent years, the problem of Visual Question Answering (VQA) - the task of answering a question about an image has become a hotbed of research activity in the computer vision community. While there are several publicly available VQA datasets [Antol et al., 2015, Johnson et al., 2017a, Krishna et al., 2016, Malinowski and Fritz, 2014], our focus in this chapter will be on the dataset provided in [Antol et al., 2015] and [Goyal et al., 2017], which is the largest natural image-question-answer dataset and the most widely cited. Even so, the narrowed-down version of the VQA problem on this dataset is not monolithic - ideally, several different skills are required by a model to answer the various questions. In Figure 3.1(left), a question like “What time is it?” requires the acquired skill of being able to read the time on a clock-face, “What is the title of the top book?” requires an OCR-like ability to read sentences, whereas the question “What color is the grass?” can be answered largely using statistical biases in the data itself (because frequently in this dataset, grass is green in color). Many models have attempted to solve the problem

of VQA with varying degrees of success, but among them, the vast majority still attempt to solve the VQA task by exploiting biases in the dataset [Kazemi and Elqursh, 2017, Teney et al., 2018, Agrawal et al., 2018, Fukui et al., 2016, Benyounes et al., 2017], while a smaller minority address the individual problem types [Andreas et al., 2016, Trott et al., 2018]. Keeping the former in mind, in this work, we provide a method to discover and enumerate explicitly, the various biases that are learned by a VQA model. For example in Figure 3.1(right), we provide examples of questions containing the phrase “How many?”, which a strong baseline model [Kazemi and Elqursh, 2017] answers with “4”. Our method discovers that this trained VQA model seems to have learned that giraffes and chairs have four legs, stop signs have four letters, and kitchen stoves have four burners. The core of our method to discovering such biases is the classical Apriori algorithm [Agrawal and Srikant, 1994] which is used to discover rules in large databases - here the *database* refers to the VQA validation set, which can be mined to produce these rules.

In theory, it can be argued that most deep learning algorithms reduce training error by learning biases in the data, no matter what the task [Wang et al., 2017]. This is evident from the observation that validation/test samples from the long tail of a data distribution are hard to solve, simply because similar examples do not occur frequently enough in the training set. However, explicitly enumerating these biases in a human-interpretable form is possible only in a handful of problems, such as VQA . VQA is particularly illustrative because the questions and answers are in human language, while the images (and attention maps) can also be interpreted by human beings. VQA is also interesting because it is a multi-modal problem - both

language and vision are required to solve this problem. The language alone (i.e., an image agnostic model) can generate plausible (but often incorrect) answers to *most* questions (as we show in Section 3.4.1), but incorporating the image generates more accurate answers. That the language alone is able to produce plausible answers strongly indicates that VQA models implicitly use simple rules to produce answers - we endeavour in this chapter to find an approach that can discover these rules.

Finally, we note that in this work, we do not seek to improve upon the state of the art. We do most of our experiments on the model of [Kazemi and Elqursh \[2017\]](#), which is a strong baseline for this problem. We choose this model because it is simple to train and analyze. To concretely summarize, our main contribution is to provide a system that can capture macroscopic rules that a VQA model ostensibly utilizes to answer questions. To the best of our knowledge, this is the first detailed work that analyzes the VQA dataset of [Goyal et al. \[2017\]](#) in this manner.

The rest of this chapter is arranged as follows : In Section 3.2 , we discuss related work, specifically those which look into “debugging” models and identifying pathological behaviors of VQA models. In Section 3.3, we discuss details of our method. In Section 5.2, we provide experimental results and list (in a literal sense) some rules we believe the model is employing to answer questions. We discuss limitations of this method and conclude in Section 5.4.

## 3.2 Background and Related Work

The VQA problem is most often solved as a multi-class classification problem. In this formulation, an image(I) usually fed through a CNN, and a question(Q) fed through a language module like an LSTM [Hochreiter and Schmidhuber, 1997] or GRU [Cho et al., 2014], are jointly mapped to an answer category (like “yes”, “no”, “1”, “2”, etc). Although the cardinality of the set of all answers given a QI dataset is potentially infinite, researchers have observed that a set of a few thousand (typically 3000 or so) most frequently occurring answers can account for over 90% of all answers in the VQA dataset. Further, the evaluation of VQA in Antol et al. [2015] and Goyal et al. [2017] is performed such that an answer receives partial credit if at least one human annotator agreed with the answer, even if it might not be the answer provided by the majority of the annotators. This further encourages the use of a classification based VQA system that limits the number of answers to the most frequent ones, rather than an answer generation based VQA system (say, using a decoder LSTM like Vinyals et al. [2015]).

**On debugging deep networks:** The seminal work by Lipton [2018] suggests that the Machine Learning community does not have a good understanding of what it means to interpret a model. In particular, this work expounds *post-hoc interpretability* - the act of interpreting a model’s behavior based on some criteria, such as visualizations of gradients [Selvaraju et al., 2017], or attention maps [Xu et al., 2015] after the model has been trained. Locally Interpretable Model Agnostic Explanations (LIME), [Ribeiro et al., 2016] explain a classifier’s behavior at a particular

point by perturbing the sample and building a linear model using the perturbations and their predictions. A follow up work [Ribeiro et al., 2018] constructs *Anchors*, which are features such that, in an instance where these features hold, a model’s prediction does not change. This work is the most similar prior work to ours, and the authors provide a few results on VQA as well. However, they only assume the existence of a model, and perturb instances of the data, whereas ours assumes the existence of responses to a dataset, but not the model itself. We use standard rule finding algorithms and provide much more detailed results on the VQA problem.

**On debugging VQA :** Agrawal et al. [2016a] study the behavior of models on the VQA 1.0 dataset. Through a series of experiments, they show that VQA models fail on novel instances, tend to answer after only partially reading the question and fail to change their answers across different images. In Agrawal et al. [2018], recognizing that deep models tend to use a combination of identifying visual concepts and prediction of answers using biases learned from the data, the authors develop a mechanism to disentangle the two. However, they do not explicitly find a way to discover such biases in the first place. In Goyal et al. [2017], the authors introduce a second, more balanced version of the VQA dataset that mitigates biases (especially language based ones) in the original dataset.

### 3.3 Method

We cast our bias discovery task as an instance of the rule mining problem, which we shall describe below. The connection between discovering biases in VQA

and rule mining is as follows : each (Image, Question, Answer) triplet can be cast a transaction in database, where each word in the question, answer and image patch (or visual word) is akin to an item. There are now three components to our rule mining operation :

- First, a frequent itemset miner picks out a set of all itemsets which occur at least  $s$  times in the dataset where  $s$  is the support. Because our dataset has over 200,000 questions (the entire VQA validation set), and the number of items exceeds 40,000 (all question words+all answer words+all visual words), we choose GMiner [Chon et al., 2018] due to its speed and efficient GPU implementation.
- Next, a rule miner Apriori [Agrawal and Srikant, 1994] forms all valid association rules  $A \rightarrow C$ , such that the rule has a support  $> s$  and a confidence  $> c$ . Here, the itemset  $A$  is called *antecedent* and the itemset  $C$  is called *consequent*. We choose  $s = 0.0005$  in this instance and do not place initial bounds on  $c$ .
- Finally, a post-processing step removes obviously spurious rules by considering the causal nature of the VQA problem (i.e., only considering rules that obey : Image/Question  $\rightarrow$  Answer). For the purpose of the current work, we query these rules with search terms like {What,sport}.

More concretely, let the  $i^{th}$  (Image, Question) pair result in the network predicting the answer  $a^i$ . Let the question itself contain the words  $\{w_1^i, w_2^i, \dots, w_k^i\}$ . Further, while answering the question, let the part of the image that the network shows attention towards correspond to the visual code-word  $v^i$ . Then, this QI+A

corresponds to the transaction  $\{w_1^i, w_2^i, \dots, w_k^k, v^i, a^i\}$ . By pre-computing and combining question, answer and visual vocabularies, each item in a transaction can be indexed uniquely. This is shown in Figure 3.2 and explained in greater detail in the following sections.

### 3.3.1 Baseline Model

The baseline model we use in this work is from [Kazemi and Elqursh, 2017], which was briefly a state-of-the-art method, yielding higher performance than other, more complicated models. We choose this model for two reasons : first, its simplicity (in other words, an absence of “bells and whistles”) makes it a good test-bed for our method and has been used by other works that explore the behavior of VQA algorithms [Mudrakarta et al., 2018, Feng et al., 2018]. The second reason is that the performance of this baseline is within 4% of the single-model state of the art [Teney et al., 2018] without using external data. We use the implementation of <https://github.com/Cyanogenoid/pytorch-vqa>.

### 3.3.2 Visual Codebook Generation

We generate the visual codebook using the classical “feature extraction followed by clustering” technique of [Sivic and Zisserman, 2003]. First, we use the bounding-box annotations in MSCOCO [Lin et al., 2014] and COCO-Stuff [Caesar et al., 2018] to extract 300,000 patches from the MSCOCO training set. After resizing each of the patches to  $224 \times 224$  pixels, we extract ResNet-152 [He et al.,

[2016] features for each of these patches, and cluster them into 1250 clusters using k-means clustering [Ding et al., 2015]. We note in Figure 3.3 that the clusters have both expected and unexpected characteristics beyond “objectness” and “stuffness”. Expected clusters include dominant objects in the MSCOCO dataset like zebras, giraffes, elephants, cars, buses, trains, people, etc. However, other clusters have textural content, unusual combinations of objects as well as actions. For example, we notice visual words like “people eating”, “cats standing on toilets”, “people in front of chain link fences”, etc, as shown in Figure 3.3. The presence of more *eclectic* code-words casts more insight into the model’s learning dynamics - we would prefer frequent itemsets containing the visual code-word corresponding to “people eating” than just “people” for a QA pair of (*what is she doing?*, *eating*).

### 3.3.3 From attention map to bounding box

In this work, we make an assumption that the network focuses on exactly one part of the image, although our method can be easily extended to multiple parts. Following the elucidation of our method in Section 3.3 and, given an attention map, we would like to compute the nearest visual code-word. Doing so requires making the choice of a bounding box that covers enough of the salient parts of the image. While there are trainable (deep network based) methods of doing so [Wang and Shen, 2017], we instead follow the simpler formulation suggested by Chen et al. [2016], which states that : given a percentage ratio  $\tau$ , find the smallest bounding



box B which satisfies :

$$\sum_{p \in B} G(p) \geq \sum_p G(p), \tau \in [0, 1]$$

Since we follow [Kazemi and Elqursh \[2017\]](#) who use a ResNet-152 architecture for visual feature extraction, the attention maps are of size  $14 \times 14$ . It can be shown easily that given a  $m \times n$  grid, the number of unique bounding boxes that can be drawn on this grid, i.e.,  $num\_boxes = \frac{m \times n \times (m+1) \times (n+1)}{4}$ , and when  $m = n = 14$ ,  $num\_boxes$  turns out to be 11,025. Because  $m(=n)$  is small and fixed in this case, we pre-compute and enumerate all 11,025 bounding boxes and pick the smallest one which encompasses the desired attention, with  $\tau = 0.3$ . This part of the pipeline is depicted in [Figure 3.4](#).

### 3.3.4 Pipeline Summarized

Now, the pipeline for the experiments ([Figure 3.2](#)) on the VQA dataset including images is as follows. We provide as input to the network in - an image and a question. We observe the second attention map and use the method of [Section 3.3.3](#) to place a tight-fitting bounding-box around those parts of the image that the model attends to. We then extract features on this bounding-box using a ResNet-152 network and perform a  $k$ -nearest neighbor search (with  $k = 1$ ) to obtain its nearest visual word from the vocabulary. The words in the question, visual code-word and predicted answer for the entire validation set are provided as the database of transactions to the frequent itemset miner [[Chon et al., 2018](#)], and rules are then obtained using the Apriori algorithm [[Agrawal and Srikant, 1994](#)].

## 3.4 Experiments

### 3.4.1 Language only statistical biases in VQA

We show that a large number of statistical biases in VQA are due to language alone. We illustrate this with an obvious example : a language-only model, i.e., one that does not see the image, but still attempts the question, answers about 40% of the questions correctly on VQA 2.0 validation set and 48% of the questions correctly on VQA 1.0 validation set. However, on a random set of 200 questions from VQA 2.0, we observed empirically that the language-only model answers 88.0% of questions with a *plausibly correct* answer even with a harsh metric of what *plausible* means. Some of these responses are fairly sophisticated as can be seen in Table 3.1. We note, for example, that questions containing “kind of bird” are met with a species of bird as response, “What kind of cheese” is answered with a type of cheese, etc. To the naked eye, it seems that the model maps out key words or phrases in the question and *ostensibly* tries to map them through a series of rules to answer words. This strongly indicates that these are biases learned from the data, and the ostensible rules can be mined through a rule-mining algorithm.

### 3.4.2 Vision+Language statistical biases in VQA

In this section, we will examine some rules that have been learned by our method on some popular question types in VQA . Question types are taken from [Antol et al., 2015] and for the purpose of brevity, only a very few instructive rules

| Question                                      | Predicted  | Ground-truth |
|---|------------|--------------|
| What kind of bird is perched on this branch ? | Owl        | Sparrow      |
| What does that girl have on her face ?        | Sunglasses | Nothing      |
| What kind of cheese is on pizza ?             | Mozzarella | Mozzarella   |
| What is bench made of ?                       | Wood       | Wood         |
| <b>What brand of stove is in kitchen ?</b>    | Electric   | LG           |

Table 3.1: We run a language-only VQA baseline and note that although only 40% of the questions are answered correctly in VQA 2.0, a large number of questions (88%) in our experiments are answered with plausibly correct responses. For example, “Sunglasses” would be a perfectly plausible answer to the question “What does that girl have on her face?” - perhaps even more so than the ground-truth answer (“Nothing”). The **last example** shows an implausible answer provided by the model to the question.

for each question type are displayed. These question types are : “What is he/she doing?”, “Where?” (Figure 3.9), “What time?”, “What brand?” (Figure 3.8), and “Why?”. The tables we present are to be interpreted thus : A question containing the antecedent words paired with an image containing the antecedent visual words can sometimes (but not always) lead to the consequent answer. Two instances of patches mapping to this visual word (Section 3.3.2) are provided. The presence of an \* after the consequent is to remind the reader that the consequent word came from the set of answers.

#### 3.4.2.1 What time?

A selection of rules involving “What time?” questions are provided in Figure 3.5 which depend on whether the query is for the general time of the day, the current time obtained by reading a clock-face or the time (i.e., season) of the year. The model used in our work, [Kazemi and Elqursh \[2017\]](#), does not have the ability to read the time - it merely guesses a random time in the HH:MM format, as long as this is one of the answer categories. A single antecedent word phrase can be associated with multiple antecedent visual words. Indeed, there are several visual words associated with afternoon and night, but we have provided only two for brevity.

#### 3.4.2.2 Why?

Traditionally, “Why?” questions in VQA are considered challenging because they require a reason based answer. We describe some of the rules purportedly

learned by our model for answering “Why?” questions, in Figure 3.6. Some interesting but intuitive beliefs that the model has learned are that movements cause blurry photographs (why,blurry→movement), outstretching one’s arms help in balancing (why,arm→balance) and that people wear helmets or orange vests for the purpose of safety (why,helmet/orange→safety). In many of these cases, no visual element has been picked up by the rule mining algorithm - this strongly indicates that the models are memorizing the answers to the “Why?” questions, and not performing any reasoning. In other words, we could ask the question “Why is the photograph blurry?” to an irrelevant image and obtain “Movement” as the predicted answer.





### 3.4.2.3 What is he/she doing?

More interesting are our results on the “What is he/she doing?” category of questions (Figure 3.7). While common activities like “snowboarding” or “typing” are prevalent among the answers, we noticed a difference in rules learned for male and female pronouns. For the female pronoun (she/woman/girl), we observed only stereotypical outputs like “texting” even for a very low support, as compared to a more diverse set of responses with the male pronoun. This is likely, a reflection on the inherent bias of the MSCOCO dataset which the VQA dataset of [Antol et al., 2015, Goyal et al., 2017] is based on. Curiously, another work by Hendricks et al. [2018] had similar observations for image captioning models also based on MSCOCO.

### 3.5 Limitations and Summary

In this work, we present a simple technique to explicitly discover biases and correlations learned by VQA models. To do so, we store in a database - the words in the question, the response of the model to the question and the portion of the image attended to by the model. Our method then leverages the Apriori algorithm to discover rules from this database. We glean from our experiments that VQA models intuitively seem to correlate *elements* in the question and image to answers. While simplicity is the primary advantage of our method, some drawbacks are the following : the exact nature of these elements is limited by the process used to generate the visual vocabulary. As a result, rules involving colors are difficult to identify because ResNets are trained to be somewhat invariant to colors. The visual attention could also focus on the wrong part of the image. Further, the mapping between an image region and the visual vocabulary is an inexact process.

Our work is consistent with other works in deep learning on fairness and accountability, which often shows a skew towards one set of implied factors (like gender), compared to others. It is also possible to use the ideas in this work to demonstrate effectiveness of VQA systems - showing dataset biases presented by a frequent itemset and rule miner is a viable alternative to cherry-picking examples of questions answered correctly by the system. Finally, our method is not limited only to VQA , but any problem with a discrete vocabulary (textual or visual). A possible future extension of this work is to track the development of these rules as a function of training time.

| No. | antecedant words | antecedant visual words  | consequents | support<br>$\times 10^{-5}$ | confidence |
|-----|------------------|--|-------------|-----------------------------|------------|
| 1   | chair,many,how   |  | 4*          | 2.33259                     | 0.56       |
| 2   | many,burner,how  |  | 4*          | 3.73214                     | 0.38       |
| 3   | many,leg,how     |  | 4*          | 2.33259                     | 0.33       |
| 4   | how,letter,many  |  | 4*          | 2.33259                     | 0.71       |




|   |   |   |
|---|---|---|
|  |  |  |
| Q : What time is it ?<br>A : 1:40 PM  | Q : What is the title of the top book?<br>A : Essential System Administration     | Q : What color is the grass ?<br>A : Green  |

Figure 3.1: In Figure 1 (left), we show examples of two questions in VQA which the model requires a “skill” to answer (such as telling the time, or reading), and a third which can be answered using statistical biases in the data. On the right, we show examples of statistical biases which lead a model to answer “4” (referred to as *consequents*), given a set of questions containing the phrase “How many?” and various visual elements (*antecedents*). Note that each row in this figure represents multiple questions in the VQA validation set. This particular instance of the trained VQA model seems to have learned that giraffes and chairs have four legs, stop signs have four letters, and kitchen stoves have four burners. The \* next to the answer reminds us that it is from the set of answer words. Upon inspection, we found 33 questions (out of >200k) in the VQA validation set which contain the words {How,many,burners} and the most common answer predicted by our model for these is 4 (which also resembles the ground-truth distribution). However, some of them were along the lines of “How many burners are turned on?”, which led to answers different from “4”.

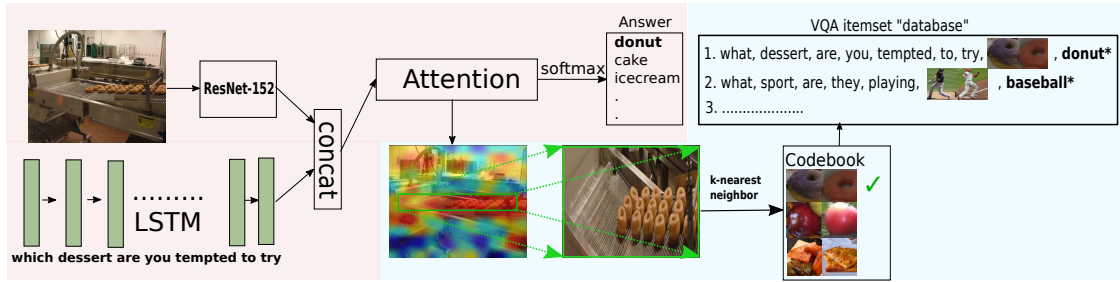


Figure 3.2: The model from [Kazemi and Elqursh, 2017] tries to answer the question "Which dessert are you tempted to try?". In doing so, the visual attention focuses on a region of the image which contains donuts. We use the method by [Chen et al., 2016] to place a bounding box over this region, which maps to a distinct visual word representing *donuts* in our vocabulary. Our database of items thus contains all of the words of the question, the visual word and the answer words. Rules are then extracted using the Apriori algorithm [Agrawal and Srikant, 1994]



Figure 3.3: We show visual code-words generated by the method of Section 3.1. In the first (left-most) column, we notice visual code-words corresponding to objects or patches in MSCOCO, but in the latter two columns (on the right) we notice code-words corresponding to more complex visual concepts like "people eating", "women in bridal-wear" or "black-and-white tennis photographs".






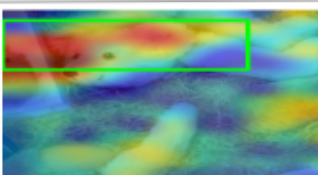
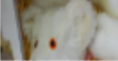

| Image + Question<br>+ Pred. Answer   | Attention map  | Crop  | Nearest Codeword<br>+ Description   |
|--|--|---|---|
|  <p data-bbox="302 621 623 653">Is there a fence? A:Yes</p>                 |   |  |  <p data-bbox="1130 590 1403 621">Net or fence like textures</p>   |
|  <p data-bbox="302 825 623 898">What are these ladies doing? A:Boating</p>  |   |  |  <p data-bbox="1138 764 1398 825">Boats, sometimes with people</p> |
|  <p data-bbox="302 1071 623 1144">What type of animal is this? A: Bear</p> |  |   |  <p data-bbox="1138 1010 1403 1041">Teddy bears</p>                |

Figure 3.4: In the first example, critical to answering the question correctly is discovering the presence of a fence (shown in red) in the attention heat-map. The cropping method of [Chen et al. \[2016\]](#) places a conservative box over this region, which corresponds to net-like or fence-like visual code-words like a tennis-net or a baseball batting-cage in the visual codebook. Similarly, in the second example, the attention corresponds to a visual code-word which clearly depicts boats, and in the third example, the attention corresponds to the teddy-bear code-word.





| No. | antecedant words     | antecedant visual words   | consequents | support<br>x 10 <sup>-5</sup> | confidence |
|-----|----------------------|---|-------------|-------------------------------|------------|
| 1   | what,time,day        |  | afternoon*  | 5.1317                        | 0.55       |
| 2   | what,time,day        |  | night*      | 3.26563                       | 1.0        |
| 3   | what,time,clock,show |  | 11:30*      | 3.26563                       | 0.21       |
| 4   | what,time,year       |  | fall*       | 2.33259                       | 0.38       |

Figure 3.5: **What time?** : Rule 1 shows kite-flying during the daytime, whereas rule 2 shows traffic lights during night. “What time?” asked about an image containing a clock prompts the model to guess a random hour of the day (rule 3). The fall season seems to be associated with a visual word depicting leafless trees (rule 4).



| No. | antecedant words | antecedant visual words   | consequents | support<br>x 10 <sup>-5</sup> | confidence |
|-----|------------------|---|-------------|-------------------------------|------------|
| 1   | arm,why          | -   | balance*    | 3.26563                       | 0.47       |
| 2   | why,umbrella     |  | raining*    | 2.79911                       | 0.6        |
| 3   | umbrella,why     |  | shade*      | 6.06473                       | 0.62       |
| 4   | why,blurry       | -   | movement*   | 6.06473                       | 0.46       |
| 5   | behind,why,fence | -   | safety*     | 2.33259                       | 0.63       |
| 6   | orange,why       | -   | safety*     | 2.33259                       | 0.5        |
| 7   | helmet,why       | -   | safety*     | 4.66518                       | 0.77       |

Figure 3.6: **Why?** : Rules that exceeded the support threshold indicate that arms are outstretched for balance (rule 1), umbrellas protect one from rain and provide shade (rules 2-3), and that fences, orange (vests) and helmets lead to safety (rules 5-7).




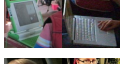

| No. | antecedant words | antecedant visual words   | consequents      | support<br>x 10 <sup>-5</sup> | confidence |
|-----|------------------|---|------------------|-------------------------------|------------|
| 1   | doing,what,man   |    | skateboarding*   | 13.529                        | 0.81       |
| 2   | doing,what,man   |    | snowboarding*    | 2.79911                       | 0.55       |
| 3   | doing,what,man   |    | flying kite*     | 2.79911                       | 0.75       |
| 4   | doing,what,man   |    | surfing*         | 4.19866                       | 1.0        |
| 5   | doing,what,man   |    | playing frisbee* | 2.33259                       | 0.31       |
| 6   | doing,what,man   |   | typing*          | 1.86607                       | 0.67       |
| 7   | doing,what,woman |  | texting*         | 1.86607                       | 0.4        |

Figure 3.7: **What is he/she doing?** : The rules in this table show standard activities in the VQA (and MSCOCO) datasets like skateboarding, snowboarding, flying a kite, playing frisbee, etc. We observed a difference in diversity of rules for male (he,man,boy) and female pronouns (she,woman,girl,lady) even at very low support. This indicates that the VQA , or more likely, the MSCOCO datasets are unintentionally skewed in terms of gender.

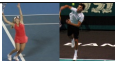




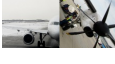
| No. | antecedant words    | antecedant visual words   | consequents | support<br>x 10 <sup>-5</sup> | confidence |
|-----|---------------------|---|-------------|-------------------------------|------------|
| 1   | brand,what          |  | wilson*     | 5.59822                       | 0.57       |
| 2   | brand,computer,what |  | dell*       | 5.1317                        | 0.5        |
| 3   | brand,what          |  | apple*      | 4.19866                       | 0.56       |
| 4   | brand,what          |  | yamaha*     | 6.99777                       | 0.43       |
| 5   | brand,what          |  | nokia*      | 2.33259                       | 0.63       |
| 6   | brand,what          |  | jetblue*    | 2.33259                       | 0.38       |
| 7   | brand,what,soda     | -   | coca cola*  | 6.06473                       | 0.33       |

Figure 3.8: **What brand?** : The VQA model seems to have learned that the Wilson brand is related to tennis, Dell and Apple make laptop computers and that Jetblue is a “brand” of airline. The visual similarity between old models of Nokia phones and TV remotes explains rule 5. Interestingly, rule 7, which pertains to “What brand of soda?” does not have an accompanying visual word. This indicates either that the model has not learned to disambiguate between various soda brands, or that our rule finding method has failed to learn of such a disambiguating rule.







| No. | antecedant words | antecedant visual words   | consequents  | support<br>x 10 <sup>-5</sup> | confidence |
|-----|------------------|---|--------------|-------------------------------|------------|
| 1   | where            |    | on building* | 6.06473                       | 0.21       |
| 2   | where,elephant   |    | africa*      | 4.19866                       | 0.39       |
| 3   | where            |    | skate park*  | 5.1317                        | 0.24       |
| 4   | where            |   | bathroom*    | 5.59822                       | 0.23       |
| 5   | where            |  | airport*     | 21.9263                       | 0.61       |
| 6   | where            |  | downtown*    | 4.19866                       | 0.41       |

Figure 3.9: **Where?** : The model of [Kazemi and Elqursh \[2017\]](#) has learned that clocks often appear on facades of buildings, elephants are from Africa, aircraft can be found in airports and that buses are found in the downtown of a city

## Chapter 4: The Amazing Mysteries of the Gutter:

### Drawing Inferences Between Panels in Comic Book Narratives

#### 4.1 Overview

Comics are fragmented scenes forged into full-fledged stories by the imagination of their readers. A comics creator can condense anything from a centuries-long intergalactic war to an ordinary family dinner into a single panel. But it is what the creator *hides* from their pages that makes comics truly interesting: the unspoken conversations and unseen actions that lurk in the spaces (or gutters) between adjacent panels. For example, the dialogue in Figure 4.1 suggests that between the second and third panels, Gilda commands her snakes to chase after a frightened Michael in some sort of strange cult initiation. Through a process called *closure* [McCloud, 1994], which involves (1) understanding individual panels and (2) making connective inferences across panels, readers form coherent storylines from seemingly disparate panels such as these. In this chapter, we study whether computers can do the same by collecting a dataset of comic books (**COMICS**) and designing several tasks that require closure to solve.



Figure 4.1: Where did the snake in the last panel come from? Why is it biting the man? Is the man in the second panel the same as the man in the first panel? To answer these questions, readers form a larger meaning out of the narration boxes, speech bubbles, and artwork by applying closure across panels.

Section 4.2 describes how we create **COMICS**,<sup>1</sup> which contains  $\sim 1.2$  million panels drawn from almost 4,000 publicly-available comic books published during the “Golden Age” of American comics (1938–1954). **COMICS** is challenging in both style and content compared to natural images (e.g., photographs), which are the focus of most existing datasets and methods [Xu et al., 2015, Krizhevsky et al., 2012, Xiong et al., 2016]. Much like painters, comic artists can render a single object or concept in multiple artistic styles to evoke different emotional responses from the reader. For example, the lions in Figure 4.2 are drawn with varying de-

<sup>1</sup>Data, code, and annotations to be made available after blind review.

degrees of realism: the more cartoonish lions, from humorous comics, take on human expressions (e.g., surprise, nastiness), while those from adventure comics are more photorealistic.

Comics are not just visual: creators push their stories forward through text—speech balloons, thought clouds, and narrative boxes—which we identify and transcribe using optical character recognition (OCR). Together, text and image are often intricately woven together to tell a story that neither could tell on its own (Section 4.3). To understand a story, readers must connect dialogue and narration to characters and environments; furthermore, the text must be read in the proper order, as panels often depict long scenes rather than individual moments [Cohn, 2010]. Text plays a much larger role in **COMICS** than it does for existing datasets of visual stories [Huang et al., 2016b].

To test machines’ ability to perform closure, we present three novel cloze-style tasks in Section 4.4 that require a deep understanding of narrative and character to solve. In Section 4.5, we design four neural architectures to examine the impact of multimodality and contextual understanding via closure. All of these models perform significantly worse than humans on our tasks; we conclude with an error analysis (Section 5.3) that suggests future avenues for improvement.

## 4.2 Creating a dataset of comic books

Comics, defined by cartoonist Will Eisner as *sequential art* [Eisner, 1990], tell their stories in sequences of *panels*, or single frames that can contain both images





Figure 4.2: Different artistic renderings of lions taken from the **COMICS** dataset. The left-facing lions are more cartoonish (and humorous) than the ones facing right, which come from action and adventure comics that rely on realism to provide thrills and text. Existing comics datasets [Guérin et al., 2013, Matsui et al., 2015] are too small to train data-hungry machine learning models for narrative understanding; additionally, they lack diversity in visual style and genres. Thus, we build our own dataset, **COMICS**, by (1) downloading comics in the public domain, (2) segmenting each page into panels, (3) extracting textbox locations from panels, and (4) running OCR on textboxes and post-processing the output. Table 4.1 summarizes the contents of **COMICS**. The rest of this section describes each step of our data

|                           |           |
|---------------------------|-----------|
| # Books                   | 3,948     |
| # Pages                   | 198,657   |
| # Panels                  | 1,229,664 |
| # Textboxes               | 2,498,657 |
| Text cloze instances      | 89,412    |
| Visual cloze instances    | 587,797   |
| Char. coherence instances | 72,313    |

Table 4.1: Statistics describing dataset size (top) and the number of total instances for each of our three tasks (bottom).

creation pipeline.

#### 4.2.1 Where do our comics come from?

The “Golden Age of Comics” began during America’s Great Depression and lasted through World War II, ending in the mid-1950s with the passage of strict censorship regulations. In contrast to the long, world-building story arcs popular in later eras, Golden Age comics tend to be small and self-contained; a single book usually contains multiple different stories sharing a common theme (e.g., crime or mystery). While the best-selling Golden Age comics tell of American superheroes triumphing over German and Japanese villains, a variety of other genres (such as romance, humor, and horror) also enjoyed popularity [Goulart, 2004]. The Digital

Comics Museum (DCM)<sup>2</sup> hosts user-uploaded scans of many comics by lesser-known Golden Age publishers that are now in the public domain due to copyright expiration. To avoid off-square images and missing pages, as the scans vary in resolution and quality, we download the 4,000 highest-rated comic books from DCM.<sup>3</sup>

## 4.2.2 Breaking comics into their basic elements

The DCM comics are distributed as compressed archives of JPEG page scans. To analyze closure, which occurs from panel-to-panel, we first extract panels from the page images. Next, we extract textboxes from the panels, as both location and content of textboxes are important for character and narrative understanding.

Panel segmentation: Previous work on panel segmentation uses heuristics [Li et al., 2014] or algorithms such as density gradients and recursive cuts [Tanaka et al., 2007, Pang et al., 2014a, Rigaud et al., 2015] that rely on pages with uniformly white backgrounds and clean gutters. Unfortunately, scanned images of eighty-year old comics do not particularly adhere to these standards; furthermore, many DCM comics have non-standard panel layouts and/or textboxes that extend across gutters to multiple panels.

After our attempts to use existing panel segmentation software failed, we turned to deep learning. We annotate 500 randomly-selected pages from our dataset with rectangular bounding boxes for panels. Each bounding box encloses both the

---

<sup>2</sup><http://digitalcomicmuseum.com/>

<sup>3</sup>Some of the panels in **COMICS** contain offensive caricatures and opinions reflective of that period in American history.

panel artwork and the textboxes within the panel; in cases where a textbox spans multiple panels, we necessarily also include portions of the neighboring panel. After annotation, we train a region-based convolutional neural network to automatically detect panels. In particular, we use Faster R-CNN [Ren et al., 2015] initialized with a pretrained VGG\_CNN\_M\_1024 model [Chatfield et al., 2014] and alternatingly optimize the region proposal network and the detection network. In Western comics, panels are usually read left-to-right, top-to-bottom, so we also have to properly order all of the panels within a page after extraction. We compute the midpoint of each panel and sort them using Morton order [Morton, 1966], which gives incorrect orderings only for rare and complicated panel layouts.

Textbox segmentation: Since we are particularly interested in modeling the interplay between text and artwork, we need to also convert the text in each panel to a machine-readable format.<sup>4</sup> As with panel segmentation, existing comic textbox detection algorithms [Ho et al., 2012, Rigaud et al., 2013] could not accurately localize textboxes for our data. Thus, we resort again to Faster R-CNN: we annotate 1,500 panels for textboxes,<sup>5</sup> train a Faster-R-CNN, and sort the extracted textboxes within each panel using Morton order.

---

<sup>4</sup>Alternatively, modules for text spotting and recognition [Jaderberg et al., 2016] could be built into architectures for our downstream tasks, but since comic dialogues can be quite lengthy, these modules would likely perform poorly.

<sup>5</sup>We make a distinction between *narration* and *dialogue*; the former usually occurs in strictly rectangular boxes at the top of each panel and contains text describing or introducing a new scene, while the latter is usually found in speech balloons or thought clouds.

### 4.2.3 OCR

The final step of our data creation pipeline is applying OCR to the extracted textbox images. We unsuccessfully experimented with two trainable open-source OCR systems, Tesseract [Smith, 2007] and Ocular [Berg-Kirkpatrick et al., 2013], as well as Abbyy’s consumer-grade FineReader.<sup>6</sup> The ineffectiveness of these systems is likely due to the considerable variation in comic fonts as well as domain mismatches with pretrained language models (comics text is always capitalized, and dialogue phenomena such as dialects may not be adequately represented in training data). Google’s Cloud Vision OCR<sup>7</sup> performs much better on comics than any other system we tried. While it sometimes struggles to detect short words or punctuation marks, the quality of the transcriptions is good considering the image domain and quality. We use the Cloud Vision API to run OCR on all 2.5 million textboxes for a cost of \$3,000. We post-process the transcriptions by removing systematic spelling errors (e.g., failing to recognize the first letter of a word). Finally, each book in our dataset contains three or four full-page product advertisements; since they are irrelevant for our purposes, we train a classifier on the transcriptions to remove them.<sup>8</sup>

## 4.3 Data Analysis

In this section, we explore what makes understanding narratives in **COMICS** difficult, focusing specifically on *intrapanel* behavior (how images and text interact

---

<sup>6</sup><http://www.abbyy.com>

<sup>7</sup><http://cloud.google.com/vision>

<sup>8</sup>See supplementary material for specifics about our post-processing.

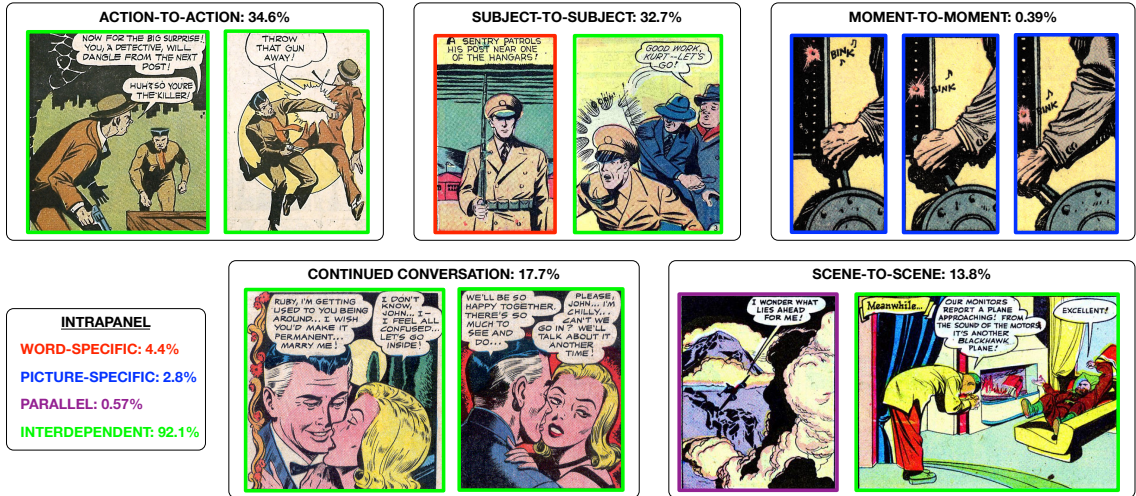


Figure 4.3: Five example panel sequences from **COMICS**, one for each type of interpanel transition. Individual panel borders are color-coded to match their intrapanel categories (legend in bottom-left). Moment-to-moment transitions unfold like frames in a movie, while scene-to-scene transitions are loosely strung together by narrative boxes. Percentages are the relative prevalence of the transition or panel type in an annotated subset of **COMICS**.

within a panel) and *interpanel* transitions (how the narrative advances from one panel to the next). We characterize panels and transitions using a modified version of the annotation scheme in Scott McCloud’s “Understanding Comics” [McCloud, 1994]. Over 90% of panels rely on both text and image to convey information, as opposed to just using a single modality. Closure is also important: to understand most transitions between panels, readers must make complex inferences that often require common sense (e.g., connecting jumps in space and/or time, recognizing when new characters have been introduced to an existing scene). We conclude that any model trained to understand narrative flow in **COMICS** will have to effectively

tie together multimodal inputs through closure.

To perform our analysis, we manually annotate 250 randomly-selected pairs of consecutive panels from **COMICS**. Each panel of a pair is annotated for intrapanel behavior, while an interpanel annotation is assigned to the transition between the panels. Two annotators independently categorize each pair, and a third annotator makes the final decision when they disagree. We use four intrapanel categories (definitions from McCloud, percentages from our annotations):

**Word-specific, 4.4%:** The pictures illustrate, but do not significantly add to a largely complete text.

**Picture-specific, 2.8%:** The words do little more than add a soundtrack to a visually-told sequence.

**Parallel, 0.6%:** Words and pictures seem to follow very different courses without intersecting.

**Interdependent, 92.1%:** Words and pictures go hand-in-hand to convey an idea that neither could convey alone.

We group interpanel transitions into five categories:

**Moment-to-moment, 0.4%:** Almost no time passes between panels, much like adjacent frames in a video.

**Action-to-action, 34.6%:** The same subjects progress through an action within the same scene.

**Subject-to-subject, 32.7%:** New subjects are introduced while staying within the same scene or idea.

**Scene-to-scene, 13.8%:** Significant changes in time or space between the two panels.

**Continued conversation, 17.7%:** Subjects continue a conversation across panels without any other changes.

The two annotators agree on 96% of the intrapanel annotations (Cohen’s  $\kappa = 0.657$ ), which is unsurprising because almost every panel is interdependent. The interpanel task is significantly harder: agreement is only 68% (Cohen’s  $\kappa = 0.605$ ). Panel transitions are more diverse, as all types except moment-to-moment are relatively common (Figure 4.3); interestingly, moment-to-moment transitions require the least amount of closure as there is almost no change in time or space between the panels. Multiple transition types may occur in the same panel, such as simultaneous changes in subjects and actions, which also contributes to the lower interpanel agreement.

#### 4.4 Tasks that test closure

To explore closure in **COMICS**, we design three novel tasks (*text cloze*, *visual cloze*, and *character coherence*) that test a model’s ability to understand narratives and characters given a few panels of context. As shown in the previous section’s analysis, a high percentage of panel transitions require non-trivial inferences from the reader; to successfully solve our proposed tasks, a model must be able to make the same kinds of connections.

While their objectives are different, all three tasks follow the same format:



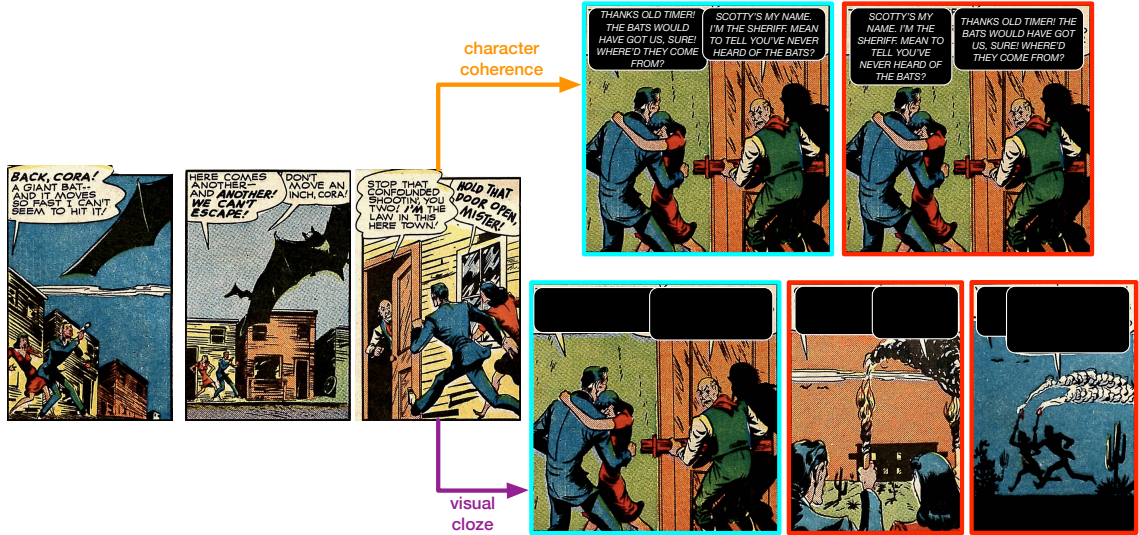


Figure 4.4: In the character coherence task (top), a model must order the dialogues in the final panel, while visual cloze (bottom) requires choosing the image of the panel that follows the given context. For visualization purposes, we show the original context panels; during model training and evaluation, textboxes are blacked out in every panel.

given preceding panels  $p_{i-1}, p_{i-2}, \dots, p_{i-n}$  as context, a model is asked to predict some aspect of panel  $p_i$ . While previous work on visual storytelling focuses on *generating* text given some context [huang2016visual](#), the dialogue-heavy text in **COMICS** makes evaluation difficult (e.g., dialects, grammatical variations, many rare words). We want our evaluations to focus specifically on closure, not generated text quality, so we instead use a cloze-style framework [[Taylor, 1953](#)]: given  $c$  candidates—with a single correct option—models must use the context panels to rank the correct candidate higher than the others. The rest of this section describes each of the three tasks in detail; [Table 4.1](#) provides the total instances of each task

with the number of context panels  $n = 3$ .

**Text Cloze:** In the *text cloze* task, we ask the model to predict what text out of a set of candidates belongs in a particular textbox, given both context panels (text and image) as well as the current panel image. While initially we did not put any constraints on the task design, we quickly noticed two major issues. First, since the panel images include textboxes, any model trained on this task could in principle learn to crudely imitate OCR by matching text candidates to the actual image of the text. To solve this problem, we “black out” the rectangle given by the bounding boxes for each textbox in a panel (see Figure 4.4).<sup>9</sup> Second, panels often have multiple textboxes (e.g., conversations between characters); to focus on interpanel transitions rather than intrapanel complexity, we restrict  $p_i$  to panels that contain only a single textbox. Thus, nothing from the current panel matters other than the artwork; the majority of the predictive information comes from previous panels.

**Visual Cloze:** We know from Section 4.3 that in most cases, text and image work interdependently to tell a story. In the *visual cloze* task, we follow the same set-up as in *text cloze*, but our candidates are images instead of text. A key difference is that models are not given text from the final panel; in *text cloze*, models are allowed to look at the final panel’s artwork. This design is motivated by eyetracking studies in single-panel cartoons, which show that readers look at artwork before reading the text [Carroll et al., 1992], although atypical font style and text length can invert

---

<sup>9</sup>To reduce the chance of models trivially correlating candidate length to textbox size, we remove very short and very long candidates.

this order [Foulsham et al., 2016].

**Character Coherence:** While the previous two tasks focus mainly on narrative structure, our third task attempts to isolate character understanding through a re-ordering task. Given a jumbled set of text from the textboxes in panel  $p_i$ , a model must learn to match each candidate to its corresponding textbox. We restrict this task to panels that contain exactly two dialogue boxes (narration boxes are excluded to focus the task on characters). While it is often easy to order the text based on the language alone (e.g., “how’s it going” always comes before “fine, how about you?”), many cases require inferring which character is likely to utter a particular bit of dialogue based on both their previous utterances and their appearance (e.g., Figure 4.4, top).

#### 4.4.1 Task Difficulty

For *text cloze* and *visual cloze*, we have two difficulty settings that vary in how cloze candidates are chosen. In the *easy* setting, we sample textboxes (or panel images) from the entire **COMICS** dataset at random. Most incorrect candidates in the easy setting have no relation to the provided context, as they come from completely different books and genres. This setting is thus easier for models to “cheat” on by relying on stylistic indicators instead of contextual information. With that said, the task is still non-trivial; for example, many bits of short dialogue can be applicable in a variety of scenarios. In the *hard* case, the candidates come from nearby pages, so models must rely on the context to perform well. For *text cloze*, all

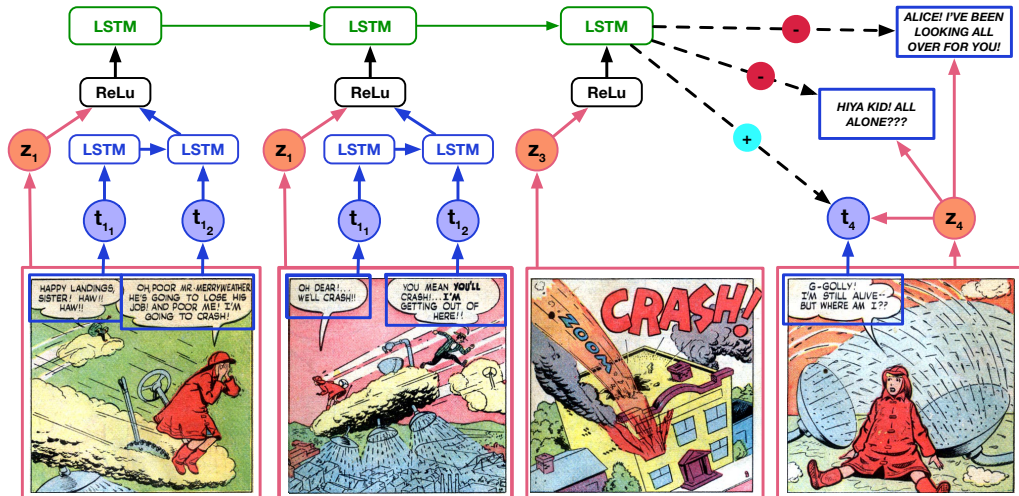


Figure 4.5: The **image-text** architecture applied to an instance of the *text cloze* task. Pretrained image features are combined with learned text features in a hierarchical LSTM architecture to form a context representation, which is then used to score text candidates.

candidates are likely to mention the same character names and entities, while color schemes and textures become much less distinguishing for *visual cloze*.

## 4.5 Models & Experiments

To measure the difficulty of these tasks for deep learning models, we adapt strong baselines for multimodal language and vision understanding tasks to the comics domain. We evaluate four different neural models, variants of which were also used to benchmark the Visual Question Answering dataset [Antol et al., 2015] and encode context for visual storytelling [Huang et al., 2016b]: *text-only*, *image-only*, and two *image-text* models. Our best-performing model encodes panels with a hierarchical LSTM architecture (see Figure 4.5).

On *text cloze*, accuracy increases when models are given images (in the form of pretrained VGG-16 features) in addition to text; on the other tasks, incorporating both modalities is less important. Additionally, for the *text cloze* and *visual cloze* tasks, models perform far worse on the *hard* setting than the *easy* setting, confirming our intuition that these tasks are non-trivial when we control for stylistic dissimilarities between candidates. Finally, none of the architectures outperform human baselines, which demonstrates the difficulty of understanding **COMICS**: image features obtained from models trained on natural images cannot capture the vast variation in artistic styles, and textual models struggle with the richness and ambiguity of colloquial dialogue highly dependent on visual contexts. In the rest of this section, we first introduce a shared notation and then use it to specify all of our models.

#### 4.5.1 Model definitions

In all of our tasks, we are asked to make a prediction about a particular panel given the preceding  $n$  panels as context.<sup>10</sup> Each panel consists of three distinct elements: image, text (OCR output), and textbox bounding box coordinates. For any panel  $p_i$ , the corresponding image is  $z_i$ . Since there can be multiple textboxes per panel, we refer to individual textbox contents and bounding boxes as  $t_{i_x}$  and  $b_{i_x}$ , respectively. Each of our tasks has a different set of answer candidates  $A$ : *text cloze* has three text candidates  $t_{a_1\dots 3}$ , *visual cloze* has three image candidates

---

<sup>10</sup>Test and validation instances for all tasks come from comic books that are unseen during training.

$z_{a_1\dots 3}$ , and *character coherence* has two combinations of text / bounding box pairs,  $\{t_{a_1}/b_{a_1}, t_{a_2}/b_{a_2}\}$  and  $\{t_{a_1}/b_{a_2}, t_{a_2}/b_{a_1}\}$ . Our architectures differ mainly in the encoding function  $g$  that converts a sequence of context panels  $p_{i-1}, p_{i-2}, \dots, p_{i-n}$  into a fixed-length vector  $c$ . We score the answer candidates by taking their inner product with  $c$  and normalizing with the softmax function,

$$s = \text{softmax}(A^T c), \quad (4.1)$$

and we minimize the cross-entropy loss against the ground-truth labels.<sup>11</sup>

Text-only: The text-only baseline only has access to the text  $t_{i_x}$  within each panel. Our  $g$  function encodes this text on multiple levels: we first compute a representation for each  $t_{i_x}$  with a word embedding sum<sup>12</sup> and then combine multiple textboxes within the same panel using an *intrapanel* LSTM [Hochreiter and Schmidhuber, 1997]. Finally, we feed the panel-level representations to an *interpanel* LSTM and take its final hidden state as the context representation (Figure 4.5). For *text cloze*, the answer candidates are also encoded with a word embedding sum; for *visual cloze*, we project the 4096-d fc7 layer of VGG-16 down to the word embedding dimensionality with a fully-connected layer.<sup>13</sup>

---

<sup>11</sup>Performance falters slightly on a development set with contrastive max-margin loss functions [Socher et al., 2014] in place of our softmax alternative.

<sup>12</sup>As in previous work for visual question answering [Zhou et al., 2015], we observe no noticeable improvement with more sophisticated encoding architectures.

<sup>13</sup>For training and testing, we use three panels of context and three candidates. We use a vocabulary size of 30,000 words, restrict the maximum number of textboxes per panel to three, and set the dimensionality of word embeddings and LSTM hidden states to 256. Models are optimized

Image-only: The image-only baseline is even simpler: we feed the `fc7` features of each context panel to an LSTM and use the same objective function as before to score candidates. For *visual cloze*, we project both the context and answer representations to 512-d with additional fully-connected layers before scoring. While the **COMICS** dataset is certainly large, we do not attempt learning visual features from scratch as our task-specific signals are far more complicated than simple image classification. We also try fine-tuning the lower-level layers of VGG-16 [Aytar et al., 2016]; however, this substantially lowers task accuracy even with very small learning rates for the fine-tuned layers.

Image-text: We combine the previous two models by concatenating the output of the intrapanel LSTM with the `fc7` representation of the image and passing the result through a fully-connected layer before feeding it to the interpanel LSTM (Figure 4.5). For *text cloze* and *character coherence*, we also experiment with a variant of the image-text baseline that has no access to the context panels, which we dub **NC-image-text**. In this model, the scoring function computes inner products between the image features of  $p_i$  and the text candidates.<sup>14</sup>

| Model         | Text Cloze  |             | Visual Cloze |             | Char. Coheren. |
|---------------|-------------|-------------|--------------|-------------|----------------|
|               | <i>easy</i> | <i>hard</i> | <i>easy</i>  | <i>hard</i> |                |
| Random        | 33.3        | 33.3        | 33.3         | 33.3        | 50.0           |
| Text-only     | 63.4        | 52.9        | 55.9         | 48.4        | 68.2           |
| Image-only    | 51.7        | 49.4        | <b>85.7</b>  | <b>63.2</b> | <b>70.9</b>    |
| NC-image-text | 63.1        | 59.6        | -            | -           | 65.2           |
| Image-text    | <b>68.6</b> | <b>61.0</b> | 81.3         | 59.1        | 69.3           |
| Human         | -           | 84          | -            | 88          | 87             |

Table 4.2: Combining image and text in neural architectures improves their ability to predict the next image or dialogue in **COMICS** narratives. The contextual information present in preceding panels is useful for all tasks: the model that only looks at a single panel (**NC-image-text**) always underperforms its context-aware counterpart. However, even the best performing models lag well behind humans.

## 4.6 Error Analysis

Table 4.2 contains our full experimental results, which we briefly summarize here. On *text cloze*, the image-text model dominates those trained on a single modality. However, text is much less helpful for *visual cloze* than it is for *text cloze*, using Adam [Kingma and Ba, 2014] for ten epochs, after which we select the best-performing model on the dev set.

<sup>14</sup>We cannot apply this model to *visual cloze* because we are not allowed access to the artwork in panel  $p_i$ .



suggesting that visual similarity dominates the former task. Having the context of the preceding panels helps across the board, although the improvements are lower in the *hard* setting. There is more variation across the models in the *easy* setting; we hypothesize that the *hard* case requires moving away from pretrained image features, and transfer learning methods may prove effective here. Differences between models on *character coherence* are minor; we suspect that more complicated attentional architectures that leverage the bounding box locations  $b_{i_x}$  are necessary to “follow” speech bubble tails to the characters who speak them.

We also compare all models to a human baseline, for which the authors manually solve one hundred instances of each task (in the *hard* setting) given the same preprocessed input that is fed to the neural architectures. Most human errors are the result of poor OCR quality (e.g., misspelled words) or low image resolution. Humans comfortably outperform all models, making it worthwhile to look at where computers fail but humans succeed.

The top row in Figure 4.6 demonstrates an instance (from *easy text cloze* where the image helps the model make the correct prediction. The text-only model has no idea that an airplane (referred to here as a “ship”) is present in the panel sequence, as the dialogue in the context panels make no mention of it. In contrast, the image-text model is able to use the artwork to rule out the two incorrect candidates.

The bottom two rows in Figure 4.6 show *hard text cloze* instances in which the image-text model is deceived by the artwork in the final panel. While the final panel of the middle row does contain what looks to be a creek, “catfish creek jail” is more suited for a narrative box than a speech bubble, while the meaning of the

correct candidate is obscured by the dialect and out-of-vocabulary token. Similarly, a camera films a fight scene in the last row; the model selects a candidate that describes a fight instead of focusing on the context in which the scene occurs. These examples suggest that the contextual information is overridden by strong associations between text and image, motivating architectures that go beyond similarity by leveraging external world knowledge to determine whether an utterance is truly appropriate in a given situation.

## 4.7 Related Work

Our work is related to three main areas: (1) multimodal tasks that require language and vision understanding, (2) computational methods that focus on non-natural images, and (3) models that characterize language-based narratives.

Deep learning has renewed interest in jointly reasoning about vision and language. Datasets such as MS COCO [Lin et al., 2014] and Visual Genome [Krishna et al., 2016] have enabled image captioning [Vinyals et al., 2015, Xu et al., 2015, Karpathy and Li, 2015] and visual question answering [Malinowski et al., 2015, Lu et al., 2016]. Similar to our *character coherence* task, researchers have built models that match TV show characters with their visual attributes [Everingham et al., 2006] and speech patterns [Haurilet et al., 2016].

Closest to our own comic book setting is the visual storytelling task, in which systems must generate [Huang et al., 2016a] or reorder [Agrawal et al., 2016b] stories given a dataset (SIND) of photos from Flickr galleries of “storyable” events

such as weddings and birthday parties. SIND’s images are fundamentally different from **COMICS** in that they lack coherent characters and accompanying dialogue. Comics are created by skilled professionals, not crowdsourced workers, and they offer a far greater variety of character-centric stories that depend on dialogue to further the narrative; with that said, the text in **COMICS** is less suited for generation because of OCR errors.

We build here on previous work that attempts to understand non-natural images. Zitnick et al. [Zitnick et al., 2016] discover semantic scene properties from a clip art dataset featuring characters and objects in a limited variety of settings. Applications of deep learning to paintings include tasks such as detecting objects in oil paintings [Crowley and Zisserman, 2014, Crowley et al., 2015] and answering questions about artwork [Guha et al., 2016]. Previous computational work on comics focuses primarily on extracting elements such as panels and textboxes [Rigaud, 2014]; in addition to the references in Section 4.2, there is a large body of segmentation research on manga [Aramaki et al., 2014, Pang et al., 2014b, Matsui, 2015, Kovanen and Aizawa, 2015].

To the best of our knowledge, we are the first to computationally model *content* in comic books as opposed to just extracting their elements. We follow previous work in language-based narrative understanding; very similar to our *text cloze* task is the “Story Cloze Test” [Mostafazadeh et al., 2016], in which models must predict the ending to a short (four sentences long) story. Just like our tasks, the Story Cloze Test proves difficult for computers and motivates future research into commonsense knowledge acquisition. Others have studied characters [Iyyer et al., 2016, Elson

et al., 2010, Bamman et al., 2014] and narrative structure [Schank and Abelson, 1977, Lehnert, 1981, Chambers and Jurafsky, 2009] in novels.

## 4.8 Summary & Future Work

We present the **COMICS** dataset, which contains over 1.2 million panels from “Golden Age” comic books. We design three cloze-style tasks on **COMICS** to explore *closure*, or how readers connect disparate panels into coherent stories. Experiments with different neural architectures, along with a manual data analysis, confirm the importance of multimodal models that combine text and image for comics understanding. We additionally show that context is crucial for predicting narrative or character-centric aspects of panels.

However, for computers to reach human performance, they will need to become better at leveraging context. Readers rely on commonsense knowledge to make sense of dramatic scene and camera changes; how can we inject such knowledge into our models? Another potentially intriguing direction, especially given recent advances in generative adversarial networks [Goodfellow et al., 2014], is generating artwork given dialogue (or vice versa). Finally, **COMICS** presents a golden opportunity for transfer learning; can we train models that generalize across natural and non-natural images much like humans do?

correct candidate

guess i ' ll ... great guns ! another ship !



incorrect candidates

you won ' t be using this transmitter

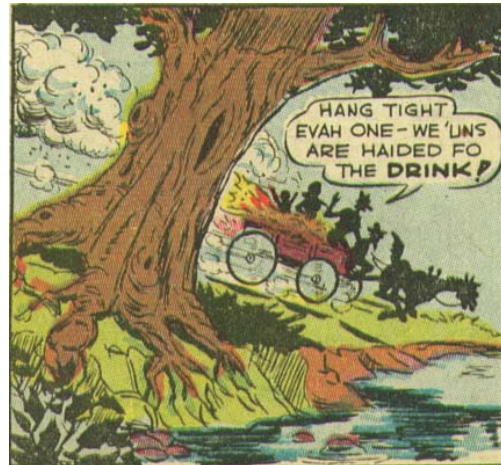
here is sorcery black magic UNK him , boys !



hang tight evah one - we ' uns are UNK for the drink !

thanks , lem ah sho nuff will

catfish creek jail



the shooting begins

black hood overcoming scorio

about this why i might be murdered next!



## Chapter 5: The Effect of Word Scrambling Across Natural Language Tasks

### 5.1 Overview

How much does word order matter for natural language processing tasks? Bag-of-words representations, which ignore word order, have historically served as reliable features for machine learning models [Wang and Manning, 2012]. This trend has continued as deep learning has gained prominence, and neural network architectures that ignore word order are often competitive with those that explicitly encode it [Iyyer et al., 2015, Wieting et al., 2016, Hill et al., 2016]. This suggests that word order is not essential for many NLP tasks, even in languages such as English that have relatively strict word order.

We ask crowdsourced participants to read both normal and randomly scrambled sentences to solve five different sentence-level tasks: sentiment analysis, textual entailment, reading comprehension, and visual question-answering in two settings.<sup>1</sup>

---

<sup>1</sup>All of our experiments are on English; we expect potentially very different results on languages with freer word order. Previous work [Yamashita, 1997] suggests that human processing time is not affected by deviations from canonical word order in Japanese, which has overt case and allows scrambling.

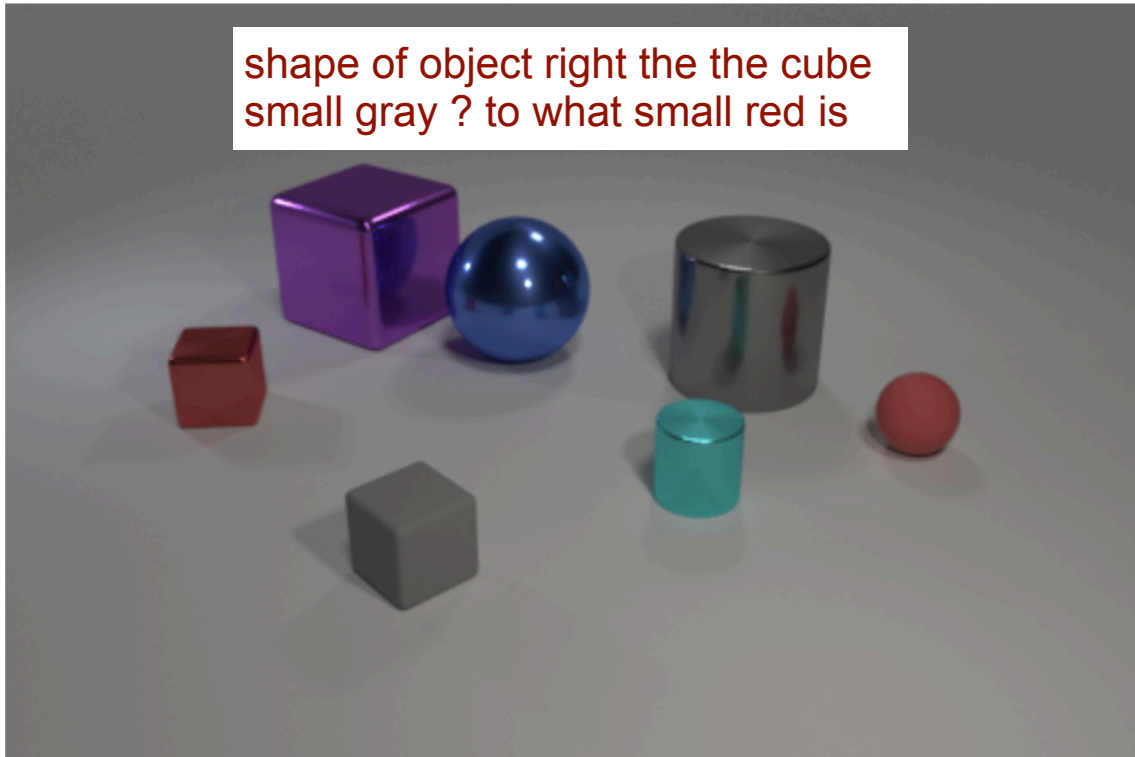


Figure 5.1: A sample task from CLEVR with a scrambled question about an image. The question is highly ambiguous: the answer depends on whether the cube is red or gray. The original question is "What shape is the small red object to the right of small gray cube?"

To discourage workers from exhaustively decoding scrambled sentences into grammatical English, we perform experiments in limited-time scenarios where we might expect them to "satisfice" [Simon, 1957], relying on "good enough" heuristics [Ferreira et al., 2002] with bags of words rather than syntactic processing. To our knowledge, this is the first human study of scrambling across multiple linguistic decision tasks.

We compare the human results with those of neural unordered and ordered

| Task  | Unscrambled |      |      | Scrambled |      |      | Neural Models |      |            |
|-------|-------------|------|------|-----------|------|------|---------------|------|------------|
|       | 5s          | 30s  | INF  | 5s        | 30s  | INF  | NBOW          | LSTM | LSTM-scram |
| SST   | 80.7        | 86.7 | 87.1 | 70.3      | 76.0 | 79.9 | 83.6          | 84.3 | 78.4       |
| SNLI  | 91.4        | 90.2 | 94.8 | 85.6      | 88.5 | 88.6 | 92.3          | 92.1 | 91.3       |
| SQUAD | 71.0        | 71.6 | 78.0 | 49.4      | 53.1 | 64.0 | 43.8          | 48.7 | N/A        |
| VGQA  | 84.0        | 85.2 | 89.3 | 83.9      | 82.3 | 88.3 | 67.6          | 67.0 | 61.5       |
| CLEVR | 79.7        | 83.3 | 86.1 | 59.7      | 64.5 | 67.5 | 54.3          | 55.8 | 47.4       |

Table 5.1: Accuracy for the five tasks for human participants and neural models.

models: our experiments demonstrate that for complex tasks, humans decline significantly more than computers in scrambled settings. This result suggests that “good enough” heuristics are not good enough to make sense of bags-of-words, implying that humans have in some sense overfit to grammatically-correct sentences.

## 5.2 Experiments

We investigate NLP tasks that differ both in difficulty and the types of reasoning required to solve them. Sentiment analysis, for example, is limited to single-sentence understanding, while solving visual question-answering problems requires connecting question text to image representations. We limit our analysis to tasks that can be cast as classification problems, which excludes complex tasks such as



machine translation and summarization.<sup>2</sup> Concretely, we measure human accuracy on scrambled and unscrambled versions of the following five tasks:

**SST**, sentiment analysis [Socher et al., 2013]: Sentence-level binary classification (positive or negative). Many sentences in this dataset contain negations and complex syntactic modifiers that require word order to properly understand.

**SNLI**, textual entailment [Bowman et al., 2015]: Given two sentences, the task is to determine the relationship between them (entailment or contradiction).<sup>3</sup> The first sentence in the pair is always unscrambled, while the second sentence varies.

**SQUAD**, reading comprehension QA [Rajpurkar et al., 2016]: Given a question about a paragraph from Wikipedia, find a one or two-word span of text within the paragraph that answers the question. The paragraph is always unscrambled, while the question varies.

**Visual Genome QA (VGQA)**, simple visual question answering [Krishna et al., 2016]: questions about photographs with five answer choices each. We limit the task to only questions whose answers are numbers ("How many lamps?") or colors ("What color is the hat?") to simplify selection of distractor candidates.

**CLEVR**: complex visual question answering [Johnson et al., 2017b]: Questions about relationships between abstract 3D objects in an artificial image (Figure 5.1) that require positional reasoning to solve. We restrict the maximum ques-

---

<sup>2</sup>This decision simplifies our crowdsourced UI; quality control is far more difficult (and expensive) when workers' answers are unconstrained.

<sup>3</sup>SNLI also contains "neutral" pairs, but in all of our experiments Turkers strongly disagreed on the distinction between "neutral" and "contradiction" even with training. Thus, our reported results are just for the binary case.

tion length to 15 words, as longer questions are difficult to solve under time pressure even in the unscrambled setting.

### 5.2.1 Human Timing Experiments

We impose three time limits for reading a given sentence: five seconds, thirty seconds, and infinite (INF) time. Our motivation is to see whether accuracy degrades more in scrambled settings when time pressure is increased vs. unscrambled settings.

We conduct our experiments on the Crowdfunder platform. For each task, we randomly select 200 examples from the test set and scramble 100 of them. We then give these examples to crowdsourced workers, where each example is answered by seven different workers for redundancy. In timed experiments, we ask workers to click a button to reveal the input text; the text disappears after a fixed number of seconds. For tasks that ask questions about some provided context (e.g., SQUAD or VQA), we allow workers to always see the context.

### 5.2.2 Neural Network Experiments

Deep learning methods are state-of-the-art on all five of our chosen tasks. Thus, we restrict our computational comparison to neural bag-of-words (word embedding sum) vs. order-aware recurrent LSTM architectures [Hochreiter and Schmidhuber, 1997]. All of the tasks aside from sentiment analysis require reasoning about additional contextual input (e.g., a Wikipedia passage in SQUAD or an image in CLEVR), which adds further architectural complexity to encode the context. We

keep the context representation method fixed for both NBOW and LSTM models.<sup>4</sup> To mimic the human setting for entailment, we always use an LSTM to encode the first sentence and switch between NBOW and LSTM encoders for the second sentence. In the visual QA tasks, we represent the image using the penultimate layer of a VGG-19 convolutional network [Simonyan and Zisserman, 2014] pretrained on ImageNet [Deng et al., 2009].

### 5.3 Discussion

In this section, we analyze the effects of scrambling and time pressure on both human and computer accuracy. We then compare humans to different neural architectures and training settings. Finally, we analyze how sentence length and complexity affect human accuracy and draw connections to psycholinguistic theories that may explain the results.

#### 5.3.1 Scrambling Degrades Human Accuracy

While human performance is above that of neural models in all of the untimed unscrambled settings, Table 1 shows that when scrambling is introduced, human performance on CLEVR, SQUAD, and SST degrades significantly. CLEVR, for which syntax is critical (as shown in Figure 5.1), drops almost 20% in absolute accuracy when word order is removed, while the simpler VGQA decreases by less than 1%.

---

<sup>4</sup>We cite previously-published NBOW and LSTM results for SQUAD [Weissenborn et al., 2017], as the implementation is complex; consequently, we are unable to report the “human-like” LSTM setting.

The difference between unordered (NBOW) and ordered (LSTM) neural models is much lower than the difference between human scrambled and unscrambled accuracy on these harder tasks; we give some possible explanations in Section 5.3.3. We also observe that human performance significantly degrades with the length of the question, as shown for CLEVR in Figure 5.2.

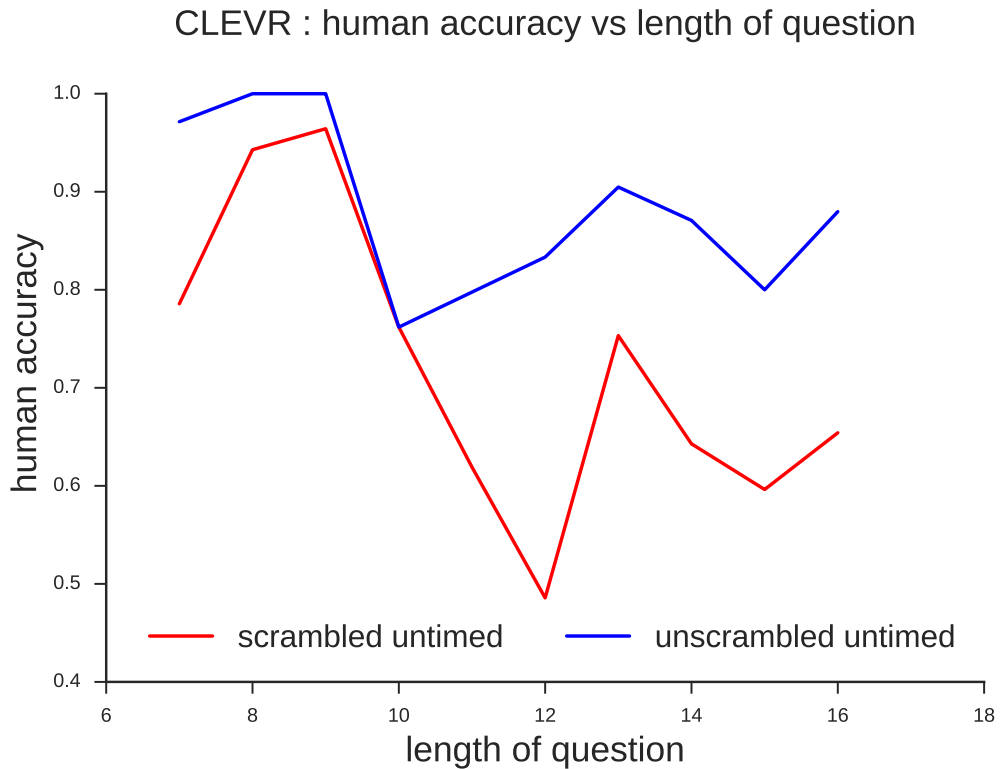


Figure 5.2: In the CLEVR task, scrambled sentences lead to ambiguities in understanding the question. In shorter questions, however, one may expect there to be fewer ambiguities compared to longer ones. This is shown empirically by a gap between human unscrambled and scrambled performance that widens with length of the question.

### 5.3.2 Time Limits Degrade Scrambled Task Accuracy

For humans, not all tasks are equally difficult under time pressure – the accuracy difference between scrambled and unscrambled settings is generally larger for more difficult tasks :

**SQUAD:**  $(64.0 - 49.4) - (78.0 - 71.0) = 7.6$

**SST:**  $(87.1 - 80.7) - (79.9 - 70.3) = 3.2$

**CLEVR:** 1.4

**SNLI:**  $-0.4$

**VGQA:**  $-0.9$

This suggests that for harder scrambled tasks, humans are actively trying to unscramble the sentence; by limiting their time, their performance degrades more in the scrambled setting. In contrast, tasks such as VGQA are easy enough that humans do not have to resort to unscrambling, which is shown by the approximately equal decline in scrambled and unscrambled settings as the time limit decreases.

### 5.3.3 Implications for Human Processing

Psycholinguistics research studies how humans and machines resolve ambiguities such as prepositional attachment [Brill and Resnik, 1994, Hindle and Rooth, 1993] and garden paths [Ferreira and Henderson, 1991, Patson et al., 2009]. In our scrambled tasks, since we randomly permute the words, workers cannot rely on syn-

tactic processing (especially in limited-time scenarios). The experiments thus force them to rely on plausible semantics, or “good enough” heuristics [Ferreira et al., 2002]. Our results show that for more difficult tasks, these heuristics are not in fact good enough to correctly solve them, and that syntactic processing is necessary. With this in mind, we consider the results for each data set.

In **SST**, while important phenomena such as negation and polarity require deeper processing [Wilson et al., 2005, Wiegand et al., 2010] and often confuse bag-of-words sentiment analyzers, we know from our neural experiments and prior research [Pang et al., 2002, Turney, 2002] that words are good enough for most cases. If words with positive polarity appear, the sentiment is likely positive, for example. Our experiments indicate that the same is true for humans: in the timed experiments, there is a consistent 10% gap between the scrambled and unscrambled conditions, but given unlimited time, humans can achieve nearly 80% accuracy.

**SNLI** True textual entailment requires not only syntactic processing, but also logical inference. High human performance on this task is likely due to the simplified definition of entailment in the dataset. Specifically, sentences often marked as “contradictions” are unrelated propositions, and in some examples, it may be possible to use only lexical similarity/dissimilarity to classify the entailment relation, obviating the need for syntax. Despite this, it is still surprising that on scrambled sentences, humans even outperform the LSTM models on *normal* sentences. It is possible that humans can correctly intuit that too many unrelated words in a sentence indicate a “contradiction”.

**SQUAD** This task is cognitively extremely demanding, and it is therefore

unsurprising that it yields the most precipitous drop in performance in the scrambled case but that accuracy increases significantly as more time is allotted. The timed experiments also test human memory, as in the five second setting it is likely that longer questions are not fully processed before the question disappears. However, low accuracy even in the untimed scrambled case suggests the questions cannot be comprehended by “good enough” heuristics alone.

**VGQA** The questions and answers in this task are extremely simple, thus lending themselves to easier plausibility-based processing. For instance, *color what ball the is* has only one plausible interpretation. In other words, a fast, plausibility-based heuristic representation is “good enough”; thus, we see very little difference in accuracy across any conditions.

**CLEVR** Unlike VGQA, this task requires that highly-specific semantic roles are preserved (Figure 5.1), but there is little inherent plausibility to the expected semantic relations between two different colored cubes in the same way that there is with the roles of the words such as *man-dog-bite*, for which humans have known semantic preferences. [Frankland, 2015, Van Herten et al., 2006] However, since the participants are *given the image*, the “plausibility” is provided by the image itself. [Gigerenzer and Goldstein, 1996] Within a short time frame, humans must process the image and possible answers, and then make a semantically-plausible choice based on a bag-of-words. As we expect, there is a massive drop (around 20% for each given timeframe) in performance when scrambling the words.

The vision-based experiments (VGQA and CLEVR) require integration of both visual and linguistic information, but yield notable differences in accuracy. With 30

seconds or more and no scrambling, the difference in human accuracy between them is slight (around 2-3%), but in the scrambled condition, CLEVR shows massive drops while VGQA shows almost none. The latter can rely general plausibility heuristics both because these are images with typical actors and objects and because the sentences are simple enough to be interpretable when scrambled, neither of which is the case in CLEVR.

## 5.4 Summary

The competitive performance of unordered neural models when compared to LSTM based models raises questions about the importance of word order in NLP tasks. To shine further light on this problem, we present a set of timed human experiments that involve solving NLP tasks with scrambled and unscrambled textual inputs. We show empirically that the degradation of human performance on scrambled inputs lies on a spectrum. This indicates that many common NLP tasks, such as those in our experiments might not intrinsically require word order, but humans struggle in orderless settings possibly because they are accustomed to grammatically ordered sentences.



## Chapter 6: Conclusion

In this thesis, we introduce two novel problems to the research community meant to be solved by DNNs and provide two novel means of analyzing problems that are typically solved using DNNs.

In Chapter 2, we introduce the task of colorization of greyscale images from natural language. We show promising results which show that objects with statistically ambiguous colors can be uniquely colored using language. Possible extensions include the colorization of images using more abstract language, which might contain words like “angry” or “winter”, and correction of artifacts in the colorized image using refinement networks [Shrivastava et al., 2017]. Inspired by our work, Gunel et al. [2018] add generative adversarial networks [Goodfellow et al., 2014] to manipulate fashion images using language.

In Chapter 4, we create the world’s largest dataset of English language comic books, design three *cloze* style tasks around this dataset, and develop models to solve these tasks. Experiments with different neural architectures, along with a manual data analysis, confirm the importance of multimodal models that combine text and image for comics understanding. However, there is a gap in terms of performance, between humans and computers in accurately predicting the contents of a future

panel of a comic book.

In Chapter 3, we develop a simple technique that helps us understand sources of bias in solving VQA problems. To the best of our knowledge, such a detailed analysis of the behavior of VQA models has not been done before. To do this, we subject the responses of a VQA model to frequent itemset and rule mining algorithms. Through our experiments, we can confidently state that a VQA model tries to correlate words and elements in the question and image, with words in the answer. Future applications involve studying how these rules develop as a function of training time.

Finally, in Chapter 5, we study for the first time, the effect that word order has on human and machine performance on a variety of NLP and Vision tasks. We show empirically that unlike unordered neural models, human performance remains relatively high on some tasks with scrambled word order, but degrades significantly in others. This might be because many tasks do not intrinsically require word order, but humans are accustomed to sentences with fixed word orders.

## Bibliography

- Vahid Kazemi and Ali Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. 2017. URL <http://arxiv.org/abs/1704.03162>.
- Jiansheng Chen, Gaocheng Bai, Shaoheng Liang, and Zhengqin Li. Automatic image cropping: A computational complexity study. In *CVPR*, 2016.
- Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, 1994.
- Donald O. Hebb. *The organization of behavior: A neuropsychological theory*. New York, 1949.
- Warren S. McCulloch and Walter Pitts. In *The bulletin of mathematical biophysics*, 1943.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. 2014.
- Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.
- Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.
- Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. 2016.
- Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. 2016.
- Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. Visual storytelling. 2016a.
- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. 2015.

- 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.
- Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. Tips and tricks for visual question answering. 2018.
- Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. 2015.
- Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. 2015.
- Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. 2016.
- Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. In *ACM Transactions on Graphics*, 2016.
- Aditya Deshpande, Jiajun Lu, Mao-Chuang Yeh, Min Jin Chong, and David Forsyth. Learning diverse image colorization. 2017.
- Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM Transactions on Graphics*, 2004.
- Yi-Chin Huang, Yi-Shin Tung, Jun-Cheng Chen, Sung-Wen Wang, and Ja-Ling Wu. An adaptive edge detection based colorization algorithm and its applications. In *Proceedings Annual ACM International Conference on Multimedia*, 2005.
- Yuki Endo, Satoshi Iizuka, Yoshihiro Kanamori, and Jun Mitani. Deepprop: Extracting deep features from a single image for edit propagation. In *Eurographics*, 2016.
- Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics*, 9(4), 2017.
- Julie Van Camp. The colorization controversy. *The Journal of Value Inquiry*, 29(4), 1995.
- Thomas Smith and John Guild. The C.I.E. colorimetric standards and their use. *Transactions of the Optical Society*, 33(3):73, 1931.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.
- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

- Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. In *Advances in Neural Information Processing Systems*, 2016.
- Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. 2016.
- Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. FiLM: Visual reasoning with a general conditioning layer. 2018.
- Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. 2014. URL <http://arxiv.org/abs/1405.0312>.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. 2015.
- Sergio Guadarrama, Ryan Dahl, David Bieber, Mohammad Norouzi, Jonathon Shlens, and Kevin Murphy. Pixcolor: Pixel recursive colorization. *arXiv preprint arXiv:1705.07208*, 2017.
- Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. 2017.
- Mohit Iyyer, Varun Manjunatha, Anupam Guha, Yogarshi Vyas, Jordan Boyd-Graber, Hal Daumé III, and Larry Davis. The amazing mysteries of the gutter: Drawing inferences between panels in comic book narratives. 2017.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017a.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. URL <https://arxiv.org/abs/1602.07332>.
- Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 2014.
- 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.

- 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. In *EMNLP*, 2016.
- Hedi Ben-younes, Rémi Cadène, Matthieu Cord, and Nicolas Thome. MUTAN: multimodal tucker fusion for visual question answering. In *ICCV*, 2017.
- Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *CVPR*, 2016.
- Alexander Trott, Caiming Xiong, and Richard Socher. Interpretable counting for visual question answering. In *ICLR*, 2018.
- Yuxiong Wang, Deva Kannan Ramanan, and Martial Hebert. Learning to model the tail. In *31st Conference on Neural Information Processing Systems (NIPS)*, December 2017.
- Kyunghyun Cho, Bart van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. 2015.
- Zachary C. Lipton. The mythos of model interpretability. In *Queue*, 2018.
- Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. 2015.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ”why should i trust you?”: Explaining the predictions of any classifier. In *Knowledge Discovery and Data Mining (KDD)*, 2016.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.
- Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. In *EMNLP*, 2016a.
- Kang-Wook Chon, Sang-Hyun Hwang, and Min-Soo Kim. Gminer: A fast gpu-based frequent itemset mining method for large-scale data. In *Inf. Sci.*, volume 439-440, pages 19–38, 2018.

- Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. Did the model understand the question? In *ACL*, 2018.
- Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. Pathologies of neural models make interpretation difficult. In *Empirical Methods in Natural Language Processing*, 2018.
- Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.
- Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- Yufei Ding, Yue Zhao, Xipeng Shen, Madanlal Musuvathi, and Todd Mytkowicz. Yinyang k-means: A drop-in replacement of the classic k-means with consistent speedup. In *ICML*, 2015.
- Wenguan Wang and Jianbing Shen. Deep cropping via attention box prediction and aesthetics assessment. In *ICCV*, 2017.
- Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *ECCV*, 2018.
- Scott McCloud. *Understanding Comics*. HarperCollins, 1994.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. 2012.
- Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. 2016.
- Neil Cohn. The limits of time and transitions: challenges to theories of sequential image comprehension. *Studies in Comics*, 1(1), 2010.
- Ting-Hao (Kenneth) Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross B. Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. Visual storytelling. 2016b.
- Will Eisner. *Comics & Sequential Art*. Poorhouse Press, 1990.
- Clément Guérin, Christophe Rigaud, Antoine Mercier, Farid Ammar-Boudjelal, Karel Bertet, Alain Bouju, Jean-Christophe Burie, Georges Louis, Jean-Marc Ogier, and Arnaud Revel. eBDtheque: a representative database of comics. 2013.

- Yusuke Matsui, Kota Ito, Yuji Aramaki, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *arXiv preprint arXiv:1510.04389*, 2015.
- Ron Goulart. *Comic Book Encyclopedia: The Ultimate Guide to Characters, Graphic Novels, Writers, and Artists in the Comic Book Universe*. HarperCollins, 2004.
- Luyuan Li, Yongtao Wang, Zhi Tang, and Liangcai Gao. Automatic comic page segmentation based on polygon detection. *Multimedia Tools and Applications*, 69(1), 2014.
- Takamasa Tanaka, Kenji Shoji, Fubito Toyama, and Juichi Miyamichi. Layout analysis of tree-structured scene frames in comic images. 2007.
- Xufang Pang, Ying Cao, Rynson WH Lau, and Antoni B Chan. A robust panel extraction method for manga. In *Proceedings of the ACM International Conference on Multimedia*, 2014a.
- Christophe Rigaud, Clément Guérin, Dimosthenis Karatzas, Jean-Christophe Burie, and Jean-Marc Ogier. Knowledge-driven understanding of images in comic books. *International Journal on Document Analysis and Recognition*, 18(3), 2015.
- Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. 2015.
- Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. 2014.
- Guy M Morton. *A computer oriented geodetic data base and a new technique in file sequencing*. International Business Machines Co, 1966.
- Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1), 2016.
- Anh Khoi Ngo Ho, Jean-Christophe Burie, and Jean-Marc Ogier. Panel and speech balloon extraction from comic books. In *IAPR International Workshop on Document Analysis Systems*, 2012.
- Christophe Rigaud, Jean-Christophe Burie, Jean-Marc Ogier, Dimosthenis Karatzas, and Joost Van de Weijer. An active contour model for speech balloon detection in comics. 2013.
- Ray Smith. An overview of the tesseract ocr engine. 2007.
- Taylor Berg-Kirkpatrick, Greg Durrett, and Dan Klein. Unsupervised transcription of historical documents. 2013.



- Wilson L Taylor. Cloze procedure: a new tool for measuring readability. *Journalism and Mass Communication Quarterly*, 30(4), 1953.
- Patrick J Carroll, Jason R Young, and Michael S Guertin. Visual analysis of cartoons: A view from the far side. In *Eye movements and visual cognition*. Springer, 1992.
- Tom Foulsham, Dean Wybrow, and Neil Cohn. Reading without words: Eye movements in the comprehension of comic strips. *Applied Cognitive Psychology*, 30, 2016.
- Richard Socher, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded compositional semantics for finding and describing images with sentences. 2014.
- Bolei Zhou, Yuandong Tian, Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167*, 2015.
- Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2014.
- Yusuf Aytar, Lluís Castrejon, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Cross-modal scene networks. *arXiv*, 2016. URL <http://arxiv.org/abs/1610.09003>.
- Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015.
- Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. 2015.
- Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering, 2016.
- Mark Everingham, Josef Sivic, and Andrew Zisserman. Hello! my name is... Buffy” – automatic naming of characters in TV video. In *Proceedings of the British Machine Vision Conference*, 2006.
- Monica-Laura Haurilet, Makarand Tapaswi, Ziad Al-Halah, and Rainer Stiefelhagen. Naming TV characters by watching and analyzing dialogs. In *IEEE Winter Conference on Applications of Computer Vision*, 2016.
- Harsh Agrawal, Arjun Chandrasekaran, Dhruv Batra, Devi Parikh, and Mohit Bansal. Sort story: Sorting jumbled images and captions into stories. 2016b.
- C. Lawrence Zitnick, Ramakrishna Vedantam, and Devi Parikh. Adopting abstract images for semantic scene understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(4):627–638, 2016.

- Elliot Crowley and Andrew Zisserman. The state of the art: Object retrieval in paintings using discriminative regions. 2014.
- E. J. Crowley, O. M. Parkhi, and A. Zisserman. Face painting: querying art with photos. 2015.
- Anupam Guha, Mohit Iyyer, and Jordan Boyd-Graber. A distorted skull lies in the bottom center: Identifying paintings from text descriptions. In *NAACL Human-Computer Question Answering Workshop*, 2016.
- Christophe Rigaud. *Segmentation and indexation of complex objects in comic book images*. PhD thesis, University of La Rochelle, France, 2014.
- Yuji Aramaki, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. Interactive segmentation for manga. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference*, 2014.
- Xufang Pang, Ying Cao, Rynson W. H. Lau, and Antoni B. Chan. A robust panel extraction method for manga. In *Proceedings of the ACM International Conference on Multimedia*, 2014b.
- Yusuke Matsui. Challenge for manga processing: Sketch-based manga retrieval. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, 2015.
- Samu Kovanen and Kiyoharu Aizawa. A layered method for determining manga text bubble reading order. In *International Conference on Image Processing*, 2015.
- Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. 2016.
- David K Elson, Nicholas Dames, and Kathleen R McKeown. Extracting social networks from literary fiction. 2010.
- David Bamman, Ted Underwood, and Noah A. Smith. A Bayesian mixed effects model of literary character. 2014.
- Roger Schank and Robert Abelson. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*. L. Erlbaum, 1977.
- Wendy G Lehnert. Plot units and narrative summarization. *Cognitive Science*, 5(4), 1981.
- Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative schemas and their participants. 2009.
- Sida I. Wang and Christopher D. Manning. Baselines and bigrams: Simple, good sentiment and topic classification. 2012.

- John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. 2016.
- Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. 2016.
- Hiroko Yamashita. The effects of word-order and case marking information on the processing of japanese. *Journal of psycholinguistic research*, 26(2):163–188, 1997.
- Herbert A Simon. Models of man; social and rational. 1957.
- Fernanda Ferreira, Karl GD Bailey, and Vittoria Ferraro. Good-enough representations in language comprehension. *Current Directions in Psychological Science*, 11(1), 2002.
- Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. 2013.
- Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. 2015.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. 2016.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *CVPR*, 2017b.
- Dirk Weissenborn, Georg Wiese, and Laura Seiffe. Making neural qa as simple as possible but not simpler. *arXiv*, 2017.
- K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *cvpr*, 2009.
- Eric Brill and Philip Resnik. A rule-based approach to prepositional phrase attachment disambiguation. In *Proceedings of the 15th conference on Computational linguistics-Volume 2*, pages 1198–1204. Association for Computational Linguistics, 1994.
- Donald Hindle and Mats Rooth. Structural ambiguity and lexical relations. *Computational linguistics*, 19(1):103–120, 1993.
- Fernanda Ferreira and John M Henderson. Recovery from misanalyses of garden-path sentences. *Journal of Memory and Language*, 30(6):725–745, 1991.

- Nikole D Patson, Emily S Darowski, Nicole Moon, and Fernanda Ferreira. Lingering misinterpretations in garden-path sentences: evidence from a paraphrasing task. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(1): 280, 2009.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. 2005.
- Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, and Andrés Montoyo. A survey on the role of negation in sentiment analysis. In *Proceedings of the workshop on negation and speculation in natural language processing*, pages 60–68. Association for Computational Linguistics, 2010.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.
- Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.
- Steven Michael Frankland. *Man Bites Dog: The Representation of Structured Meaning in Left-Mid Superior Temporal Cortex*. PhD thesis, 2015.
- Marieke Van Herten, Dorothee J Chwilla, and Herman HJ Kolk. When heuristics clash with parsing routines: Erp evidence for conflict monitoring in sentence perception. *Journal of cognitive neuroscience*, 18(7):1181–1197, 2006.
- Gerd Gigerenzer and Daniel G Goldstein. Reasoning the fast and frugal way: models of bounded rationality. *Psychological review*, 103(4):650, 1996.
- Mehmet Gunel, Erkut Erdem, and Aykut Erdem. Language guided fashion image manipulation with feature-wise transformations. In *First Workshop on Computer Vision in Art, Fashion and Design) – in conjunction with ECCV 2018*, 2018.