

**Vision and Language Learning:
From Image Captioning and Visual
Question Answering towards
Embodied Agents**

Peter Anderson

A thesis submitted for the degree of
Doctor of Philosophy of
The Australian National University

April 2018

© Copyright by Peter Anderson 2018
All Rights Reserved

Except where otherwise indicated in the text, this thesis is my own original work.
This thesis does not exceed the stipulated limit of 100,000 words.

Peter Anderson
24 April 2018

For Mack and Sasha. You may regard talking to intelligent machines as completely ordinary, but it wasn't.

Acknowledgments

First and foremost, this thesis would not have been possible without the guidance and support from my supervisor, Steve Gould. Steve is incredibly knowledgeable and technically adept, yet he remains patient, generous with his time and considered in his opinions, even when confronted with the half-baked ideas of research students. He is committed to the highest ethical and research standards and well-connected within the research community. In short, it's hard to imagine a better supervisor. Thank you Steve.

I am also profoundly grateful to Mark Johnson and Macquarie University, for hosting me as a visiting student for an extended period of time. Thank you Mark, for adopting me into your group, for the many insightful discussions we had and for freely sharing your vast knowledge and expertise, which offered a crucial and different perspective on my research.

I would also like to thank Anton van den Hengel, who inspired and supported my two trips to Adelaide University, which resulted in some great collaborations with the very strong vision and language group there. Thank you Anton, for your advice and encouragement. Thanks also to Ian Reid and José Álvarez for taking an interest in my work and serving on my supervisory panel, and Basura Fernando for helping guide my research during my early stages at ANU.

During my PhD I have also benefited from discussions with many other colleagues and collaborators, including Anoop Cherian, Rodrigo Santa Cruz, Edison Guo, Damien Teney, Qi Wu, Phil Roberts, Niko Sünderhauf, Jake Bruce, Mark Dras and others. I benefited enormously from participating in lively reading group discussions at ANU and Macquarie. I also had a great opportunity to intern at Microsoft Research in Seattle, and for that I am grateful to Xiaodong He and Lei Zhang. Thank you all for the ideas, advice, and contributions that you have made to my research.

I would also like to thank everyone involved with the Australian Centre for Robotic Vision (ACRV), including the Centre Executive, researchers, students and administrative staff. It was the formation of the ACRV that convinced me that it was the right time to start a PhD. The ACRV provided me with incredible opportunities to travel and attend summer schools and conferences, fostering a level of interaction across Australian universities that would ordinarily not be possible. Had the ACRV not existed, my PhD outcomes would have been greatly diminished.

Most importantly, I'd like to thank my wife, Christina, for her unwavering support and encouragement. While I was producing papers, you brought our two beautiful kids into the world. While I was juggling travel commitments, you balanced the challenges of your own stellar career with the demands of running a household. You did all this and yet you still found time to care about my paper reviews. No words are adequate. Thank you for everything. To Mack and Sasha, I don't know how you have already learned so much about the world. You guys amaze me every day.

Finally, I would like to acknowledge the financial supporters of this research. This research was supported by an Australian Government Research Training Program (RTP) Scholarship and by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016). Collection of the R2R navigation dataset was generously supported by a Facebook ParLAI Research Award.

Abstract

Each time we ask for an object, describe a scene, follow directions or read a document containing images or figures, we are converting information between visual and linguistic representations. Indeed, for many tasks it is essential to reason jointly over visual and linguistic information. People do this with ease, typically without even noticing. Intelligent systems that perform useful tasks in unstructured situations, and interact with people, will also require this ability.

In this thesis, we focus on the joint modelling of visual and linguistic information using deep neural networks. We begin by considering the challenging problem of automatically describing the content of an image in natural language, i.e., image captioning. Although there is considerable interest in this task, progress is hindered by the difficulty of evaluating the generated captions. Our first contribution is a new automatic image caption evaluation metric that measures the quality of generated captions by analysing their semantic content. Extensive evaluations across a range of models and datasets indicate that our metric, dubbed *SPICE*, shows high correlation with human judgements.

Armed with a more effective evaluation metric, we address the challenge of image captioning. Visual attention mechanisms have been widely adopted in image captioning and visual question answering (VQA) architectures to facilitate fine-grained visual processing. We extend existing approaches by proposing a bottom-up and top-down attention mechanism that enables attention to be focused at the level of objects and other salient image regions, which is the natural basis for attention to be considered. Applying this approach to image captioning we achieve state of the art results on the COCO test server. Demonstrating the broad applicability of the method, applying the same approach to VQA we obtain first place in the 2017 VQA Challenge.

Despite these advances, recurrent neural network (RNN) image captioning models typically do not generalise well to out-of-domain images containing novel scenes or objects. This limitation severely hinders the use of these models in real applications. To address this problem, we propose *constrained beam search*, an approximate search algorithm that enforces constraints over RNN output sequences. Using this approach, we show that existing RNN captioning architectures can take advantage of side information such as object detector outputs and ground-truth image anno-

tations at test time, without retraining. Our results significantly outperform previous approaches that incorporate the same information into the learning algorithm, achieving state of the art results for out-of-domain captioning on COCO.

Last, to enable and encourage the application of vision and language methods to problems involving embodied agents, we present the Matterport3D Simulator, a large-scale interactive reinforcement learning environment constructed from densely-sampled panoramic RGB-D images of 90 real buildings. Using this simulator, which can in future support a range of embodied vision and language tasks, we collect the first benchmark dataset for visually-grounded natural language navigation in real buildings. We investigate the difficulty of this task, and particularly the difficulty of operating in unseen environments, using several baselines and a sequence-to-sequence model based on methods successfully applied to other vision and language tasks.

Contents

Acknowledgments	v
Abstract	vii
1 Introduction	1
1.1 Bridging Vision and Language	1
1.2 Image Captioning and Visual Question Answering	1
1.3 Embodied Vision and Language Agents	4
1.4 Contributions	5
1.5 Thesis Outline	7
1.6 List of Publications	8
2 Background	9
2.1 Image Captioning	9
2.2 Related Vision and Language Tasks	11
2.3 Convolutional Neural Networks	12
2.4 Recurrent Neural Networks	13
2.4.1 Long Short-Term Memory (LSTM) Networks	14
2.4.2 LSTM Encoders	15
2.4.3 LSTM Decoders	15
3 Datasets	19
3.1 ImageNet	19
3.2 COCO	21
3.3 VQA	23
3.3.1 Version 1.0	23
3.3.2 Version 2.0	24
3.4 Visual Genome	24
4 SPICE: Semantic Propositional Image Caption Evaluation	27
4.1 Image Caption Evaluation	27
4.2 SPICE Metric	29
4.2.1 Semantic Parsing from Captions to Scene Graphs	31

4.2.2	F-score Calculation	33
4.2.3	Running Time	35
4.2.4	Limitations	37
4.2.5	Alternatives to Scene Graph Representations	38
4.3	Experiments	38
4.3.1	Human Judgements	39
4.3.1.1	COCO	39
4.3.1.2	Flickr 8K.	40
4.3.1.3	Composite Dataset.	40
4.3.1.4	PASCAL-50S	40
4.3.2	System-Level Correlation	41
4.3.3	Colour Perception, Counting and Other Questions	42
4.3.4	Caption-Level Correlation	44
4.3.5	Pairwise Accuracy	44
4.4	Chapter Summary	46
5	Bottom-Up and Top-Down Visual Attention for Image Captioning and VQA	51
5.1	Attention Networks	51
5.1.1	Bottom-Up vs. Top-Down Attention	53
5.2	Approach	55
5.2.1	Bottom-Up Attention Model	55
5.2.2	Captioning Model	58
5.2.2.1	Top-Down Attention LSTM	59
5.2.2.2	Language LSTM	59
5.2.2.3	Objective	60
5.2.3	VQA Model	61
5.2.4	Implementation Details	62
5.2.4.1	Bottom-Up Attention Model	62
5.2.4.2	Captioning and VQA Models	64
5.3	Evaluation	64
5.3.1	Datasets	64
5.3.1.1	COCO	64
5.3.1.2	VQA v2.0	65
5.3.2	ResNet Baseline	65
5.3.3	Image Captioning Results	67
5.3.4	VQA Results	69
5.4	Chapter Summary	70

6	Guided Image Captioning using Constrained Beam Search	77
6.1	Test-Time Novel Object Captioning	77
6.2	Related Work	79
6.3	Approach	79
6.3.1	RNN Decoding with Beam Search	79
6.3.2	Constrained Beam Search	80
6.3.3	Application to Image Captioning	84
6.3.4	Vocabulary Expansion	86
6.4	Experiments	86
6.4.1	Dataset Pre-processing	87
6.4.2	Out-of-Domain Image Captioning	87
6.4.3	Captioning ImageNet	91
6.5	Chapter Summary	94
7	Vision and Language Navigation in Real Environments	95
7.1	Vision and Language Navigation	95
7.2	Related Work	97
7.3	Matterport3D Simulator	99
7.3.1	Matterport3D Dataset	99
7.3.2	Simulator	101
7.3.2.1	Observations	101
7.3.2.2	Action Space	101
7.3.2.3	Implementation Details	103
7.3.2.4	Biases	103
7.4	Room-to-Room (R2R) Navigation	104
7.4.1	Task	104
7.4.2	Data Collection	104
7.4.3	R2R Dataset Analysis	106
7.4.4	Evaluation Protocol	106
7.5	Vision and Language Navigation Agents	109
7.5.1	Sequence-to-Sequence Model	109
7.5.2	Training	110
7.5.3	Additional Baselines	111
7.6	Results	112
7.7	Chapter Summary	114
8	Conclusion and Future Directions	117
8.1	Summary	117

8.2	Future Directions	118
8.2.1	Complete Agents	118
8.2.2	Large-scale 3D Datasets as Simulation Environments	119
8.2.3	The Long Tail	120

List of Figures

1.1	The image captioning and visual question answering (VQA) tasks. . . .	2
1.2	The Room-to-Room (R2R) navigation task.	5
2.1	An encoder-decoder neural network architecture for image captioning.	10
2.2	The joint-embedding approach to VQA.	11
2.3	An example of a convolutional neural network (CNN).	13
2.4	Diagram of a Long Short-Term Memory (LSTM) cell.	14
3.1	Examples from the ImageNet image classification dataset.	20
3.2	Examples of annotated images in the COCO dataset.	22
3.3	Examples of questions and answers from the VQA v1.0 dataset.	24
3.4	Examples of questions and answers from the VQA v2.0 dataset.	25
3.5	An example image annotation from the Visual Genome dataset.	26
4.1	SPICE uses semantic propositional content to assess image captions. . .	30
4.2	Full example of the SPICE calculation.	36
4.3	Automatic evaluations vs. human judgements for entries in the 2015 COCO Captioning Challenge.	42
4.4	Pairwise classification accuracy of automated metrics on PASCAL-50S.	45
4.5	Additional example of the SPICE calculation.	48
4.6	Additional example of the SPICE calculation with parsing errors.	49
5.1	Calculating attention at the level of objects.	54
5.2	Example outputs from our Faster R-CNN bottom-up attention model. .	57
5.3	Overview of the proposed captioning model.	58
5.4	Overview of the VQA model used in experiments.	61
5.5	Example of a generated caption showing attended image regions. . . .	68
5.6	Qualitative differences between attention methodologies.	69
5.7	VQA example illustrating attention output.	71
5.8	Further examples of generated captions showing attended image regions.	73
5.9	Further examples of generated captions showing attended image regions.	74
5.10	Further examples of successful VQA results, showing attended image regions.	75

5.11	Examples of visual question answering (VQA) failure cases.	76
6.1	Captioning images containing previously unseen objects by incorporating image tags during RNN decoding.	78
6.2	Example of constrained beam search decoding.	83
6.3	Examples of constrained captions generated on the COCO dataset.	88
6.4	Examples of ImageNet captions constrained to include the ground-truth synset.	92
6.5	Human evaluations of ImageNet captions by super-category.	93
7.1	The Room-to-Room (R2R) navigation task.	97
7.2	Differences between Vision and Language Navigation (VLN) and VQA.	98
7.3	A snapshot of the visual diversity in the Matterport3D dataset.	100
7.4	An example navigation graph from the Matterport3D Simulator.	102
7.5	Randomly selected examples of R2R navigation instructions.	105
7.6	Distribution of instruction and trajectory length in R2R.	106
7.7	AMT data collection interface for the R2R dataset.	107
7.8	Distribution of navigation instructions based on their first four words.	108
7.9	Distribution of navigation error in validation environments.	113
7.10	Validation loss, navigation error and success rate during training.	114
8.1	A light-hearted look at inter-disciplinary research in AI.	120
8.2	Difficulties caused by the long tail of visual and linguistic concepts.	121

List of Tables

4.1	A summary of textual similarity metrics.	28
4.2	Runtime of SPICE vs. other metrics.	35
4.3	System-level Pearson’s ρ correlation between automatic evaluations and human judgements.	41
4.4	Human evaluations vs. SPICE scores by semantic proposition subcategory.	43
4.5	Caption-level Kendall’s τ correlation between automatic evaluations and human quality scores.	44
4.6	Caption-level classification accuracy of evaluation metrics at matching human judgement on PASCAL-50S.	45
5.1	Faster R-CNN detection scores on COCO vs. Visual Genome.	63
5.2	Single-model image captioning performance on the COCO Karpathy test split.	66
5.3	Breakdown of SPICE F-scores over various tuple subcategories.	66
5.4	Highest ranking published image captioning results on the online COCO test server.	68
5.5	Single-model performance on the VQA v2.0 validation set.	70
5.6	VQA v2.0 test-standard server accuracy.	70
6.1	Evaluation of captions generated using constrained beam search.	89
6.2	F1 scores for mentions of objects not seen during caption training.	90
6.3	Human evaluations comparing ImageNet captions.	91
7.1	R2R navigation results on seen and unseen environments.	112

Introduction

1.1 Bridging Vision and Language

Each time we ask for an object, describe a scene, follow directions or read a document containing images or figures, we are converting information between visual and linguistic representations. Indeed, for many tasks it is essential to reason jointly over visual and linguistic information. People do this with ease, typically without even noticing. Intelligent systems that perform useful tasks in unstructured situations, and interact with people, will also require this ability. Consider domestic service robots, voice-controlled drones, visually-aware virtual personal assistants (i.e., the future descendants of Siri, Cortana, Alexa and Google Assistant), smart buildings and appliances that respond to natural language commands, intelligent surveillance and search systems for querying large image and video collections, language-based image and video editing software, and personal navigation systems that generate and comprehend visually-grounded instructions. Many of these applications will have far-reaching importance, and in each case, research that combines the traditional fields of computer vision (CV) and natural language processing (NLP) is the only plausible approach.

1.2 Image Captioning and Visual Question Answering

Despite significant progress in CV and NLP, until recently the interaction between these fields had been much less explored. However, given the significant opportunities that await, in recent years there has been an increase in research directed towards ambitious tasks that combine visual and linguistic learning. Two tasks have emerged as key focus areas within the vision and language research community. The first is the task of automatically describing the content of an image in natural language, i.e., *image captioning* [Vinyals et al., 2015; Xu et al., 2015]. As illustrated in Figure 1.1 left, image captioning is an excellent test of visual and linguistic understanding,



	Image Captioning	Visual Question Answering
Input		
Output	A man in a pink bow tie and a pink shirt is being hugged by a man in a blue shirt.	What sort of bus is this? double decker

Figure 1.1: The image captioning (left) and visual question answering (right) tasks. Examples are taken from the COCO [Chen et al., 2015] and VQAv2 [Goyal et al., 2017] datasets respectively. Further background regarding these datasets is provided in Chapter 3.

requiring a model to identify and describe the most salient elements of an image, such as the objects present and their attributes, as well as the spatial and semantic relationships between them [Fang et al., 2015]. The second task is the task of *visual question answering*, or VQA. As illustrated in Figure 1.1 right, a VQA model takes as input an image and an open-ended natural language question about the image, and must output one of possibly several correct answers [Antol et al., 2015]. Interest in these tasks has been driven in part by the development of new and larger benchmark datasets [Chen et al., 2015; Goyal et al., 2017]. However, these tasks also have direct applications, most obviously in terms of intelligent assistants for the low vision community [Gurari et al., 2018].

One way of evaluating the current state of the art in image captioning is by undertaking a user study. An analysis of the output of one recently proposed captioning model found that people considered 52% of the generated captions to be ‘bad’¹, but this figure was only 13% for captions written by people [Liu et al., 2017a]. Clearly, there remains considerable scope for improvement. However, since user studies are expensive and difficult to replicate, generated image captions are more typically evaluated using automated metrics such as Bleu [Papineni et al., 2002] and CIDEr [Vedantam et al., 2015]. Unfortunately however, these metrics have proven to be inadequate

¹Defined as a caption that misses the foreground, main objects, events or theme of the image, or contains obviously hallucinated objects, activities or relationships.

substitutes for human judgement, and they are difficult to interpret [Kulkarni et al., 2013; Hodosh et al., 2013; Elliott and Keller, 2014]. Motivated by these limitations, one of the contributions of this thesis is an improved automated evaluation metric. Our metric, dubbed *SPICE* (an acronym for Semantic Propositional Image Caption Evaluation), evaluates generated captions in terms of the truth value of the propositions they contain. As we show in Chapter 4, *SPICE* substantially addresses the limitations of existing metrics, helping to track the state of the art and supporting the ongoing development of more effective captioning models.

In comparison to image captioning, the VQA task is much easier to evaluate as the answers are typically only one or two words. This makes it relatively straightforward to determine accuracy by comparing candidate answers to reference answers. However, as with image captioning, there remains considerable scope for further research. For example, prior to the 2017 VQA Challenge², the question answering accuracy of a VQA baseline model using the largest available dataset was only 62.3% [Goyal et al., 2017]. To improve the performance of both image captioning and VQA models, in Chapter 5 we propose a novel neural network architecture based on visual attention [Xu et al., 2015; Zhu et al., 2016]. In general, attention mechanisms focus on relevant inputs while ignoring irrelevant or distracting stimuli. Our approach leverages insights from neuroscience and psychology to make objects the basis of attention in our model [Egley et al., 1994; Scholl, 2001]. Using this approach we achieve state of the art performance in both image captioning and VQA, while simultaneously helping to improve the interpretability of the resulting systems. Our final (ensembled) entry in the 2017 VQA Challenge obtained first place with an overall accuracy of 69.0%, 0.8% ahead of the second-placed entry.

To put this result in context, a similar level of accuracy on this task could probably be attained by a toddler. For example, our model is able to correctly answer simple questions about common objects and colours, but struggles with counting and more difficult questions requiring a greater amount of prior knowledge, or ‘commonsense’. However, this characterisation is subject to two strong qualifications. First, the VQA model can only perform this single narrowly defined task, while even very young children can obviously perform many tasks and learn new tasks very quickly. Second, the VQA model’s performance deteriorates catastrophically when tested on visual concepts that are not found in the training set. This limitation also holds for image captioning models [Tran et al., 2016]. To address this brittleness, in Chapter 6 we extend our image captioning model to make use of additional information sources, such as images tagged with keywords, when generating captions. The resulting method—*constrained beam search*—is a general and principled approach to

²www.visualqa.org/challenge_2017.html

adding constraints to the output of a recurrent neural network (RNN). Using this approach, we achieve state of the art performance on ‘out-of-domain’ images containing novel scenes or objects not seen in training.

1.3 Embodied Vision and Language Agents

The image captioning and VQA tasks are ideal for encouraging and quantifying progress in vision and language understanding, but they suffer from a serious limitation. Both tasks are *passive*. The input images are static, and the model (or agent) is not allowed to move or control the camera, seek clarification, or take any other action in the environment. This neglects a crucial aspect of the motivating applications listed in Section 1.1, as each of these examples demands an agent that is embodied (or in control of some task-specific programming interface at least). In Chapter 7, the final technical chapter of this thesis, we address this limitation by connecting vision and language to *actions*. We focus on the problem of a robot executing a natural-language navigation instruction in a real 3D environment. We refer to this challenge as Vision-and-Language Navigation (VLN).

The idea that we might be able to give general, verbal instructions to a robot and have at least a reasonable probability that it will carry out the required task is one of the long-held goals of robotics, and artificial intelligence (AI). Although interpreting natural-language navigation instructions has already received significant attention [Chaplot et al., 2018; Chen and Mooney, 2011; Guadarrama et al., 2013; Mei et al., 2016; Misra et al., 2017; Tellex et al., 2011], previous approaches to the natural language command of robots have often restricted the visual complexity of the problem. In contrast, we are motivated by recent work in image captioning and VQA in which natural images are used. We note that both VQA and VLN can be interpreted as visually grounded sequence-to-sequence translation problems, and many of the same methods are applicable. Therefore, to enable and encourage the application of vision and language methods to the problem of interpreting visually-grounded navigation instructions, in Chapter 7 we present the Matterport3D Simulator. The simulator is a large-scale interactive reinforcement learning (RL) environment constructed from the Matterport3D dataset [Chang et al., 2017] which contains 10,800 densely-sampled panoramic RGB-D images of 90 real-world building-scale indoor environments. Compared to synthetic RL environments [Beattie et al., 2016; Kempka et al., 2016; Zhu et al., 2017], the use of real-world image data preserves visual and linguistic richness, maximising the potential for trained agents to be transferred to real-world applications.

Based on the Matterport3D environments, we collect the Room-to-Room (R2R)



Instruction: Head upstairs and walk past the piano through an archway directly in front. Turn right when the hallway ends at the pictures and table. Wait by the moose antlers hanging on the wall.

Figure 1.2: The Room-to-Room (R2R) navigation task. We focus on executing natural language navigation instructions in previously unseen real-world buildings. The agent’s camera can be rotated freely. Blue discs indicate nearby (discretized) navigation options.

navigation dataset containing 21,567 open-vocabulary, crowd-sourced navigation instructions with an average length of 29 words. Each instruction describes a trajectory traversing typically multiple rooms. As illustrated in Figure 1.2, the associated task requires an agent to follow natural-language instructions to navigate to a goal location in a previously unseen building. We investigate the difficulty of this task, and particularly the difficulty of operating in unseen environments, using several baselines and a sequence-to-sequence model based on methods successfully applied to other vision and language tasks [Antol et al., 2015; Chen et al., 2015; Goyal et al., 2017].

1.4 Contributions

In summary, this thesis makes four main contributions, beginning with image captioning and visual question answering (VQA), before moving to consider embodied agents:

1. **An effective evaluation metric for image descriptions.** To rapidly develop

agents that can communicate visual content, we need fast, accurate and inexpensive metrics to evaluate the quality of the generated language. Our SPICE metric reflects human judgements significantly more accurately than other automatic metrics. Even after widespread adoption, SPICE is still the only automatic evaluation metric that (correctly) judges human image descriptions to be superior to those from any proposed image captioning model³.

- 2. A combined bottom-up and top-down visual attention model.** Visual attention mechanisms have become an important component of many vision and language models [Xu et al., 2015; Zhu et al., 2016]. We leverage insights from neuroscience and psychology to make objects the basis of attention in our model. This is a general approach that more closely unifies tasks involving visual and linguistic understanding with recent progress in object detection [Ren et al., 2015a]. Using this approach we achieve state of the art performance in image captioning and VQA, while simultaneously helping to improve the interpretability of the resulting systems.
- 3. Constrained beam search for controlling the output of an RNN.** Neural network captioning models typically generalise poorly to out-of-domain images containing novel scenes or objects not seen in training [Tran et al., 2016]. Motivated by this problem, we propose constrained beam search, a principled approach to completing partial sequences by adding constraints to the output of a recurrent neural network (RNN). Using this approach to incorporate additional text fragments, such as image tags, during caption generation, we achieve state of the art captioning performance on out-of-domain images without degrading in-domain performance.
- 4. The Matterport3D Simulator and the R2R navigation dataset.** Datasets are a critical driver of progress in computer vision, not just as a source of training data, but also as a means of measuring and comparing the performance of competing algorithms [Torralba and Efros, 2011]. The proposed Room-to-Room (R2R) navigation task requires agents to follow natural language navigation instructions in previously unseen environments (using the Matterport3D Simulator). This is the first visually realistic, interactive and reproducible benchmark for evaluating embodied vision and language agents.

³Based on COCO C40 test leaderboard as at March 2018 (refer <http://cocodataset.org/#captions-leaderboard>).

1.5 Thesis Outline

The remaining chapters of this thesis are summarised below:

Chapter 2—Background. This chapter provides a general overview of the existing literature relating to image captioning, visual question answering (VQA), and related vision and language tasks, in order to put the contributions of this thesis in context. We also introduce some basic technical background that is assumed in the remainder of the thesis.

Chapter 3—Datasets. In this chapter, we outline the various pre-existing datasets that we use for training and evaluating our models, including datasets designed for image classification, image captioning, VQA and object detection.

Chapter 4—SPICE: Semantic Propositional Image Caption Evaluation. In this chapter, we introduce the task of automatic image caption evaluation. We first review existing automated evaluation metrics. Then, motivated by the role of semantic propositions in human evaluations, we present the SPICE metric. Finally, we compare SPICE with existing metrics in terms of correlation with human judgements over a range of models and datasets.

Chapter 5—Bottom-Up and Top-Down Visual Attention for Image Captioning and VQA. Using the SPICE evaluation metric from Chapter 4 and others, in this chapter we develop our image captioning model that places objects at the centre of the visual attention mechanism. We additionally illustrate the application of the same principle to VQA, and comprehensively evaluate the impact our bottom-up and top-down visual attention model on both tasks.

Chapter 6—Guided Image Captioning using Constrained Beam Search. Addressing the generalisation of captioning models to images containing novel scenes or objects, we propose constrained beam search to guarantee the inclusion of selected words or phrases in the output of a recurrent neural network, while leaving the model free to determine the syntax and additional details. We demonstrate that, when combined with image tag predictions, this approach outperforms prior work.

Chapter 7—Vision and Language Navigation in Real Environments. In this chapter, we begin to connect vision and language agents with actions. We propose a new reinforcement learning (RL) environment constructed from dense RGB-D imagery of

90 real buildings Chang et al. [2017], and collect the Room-to-Room (R2R) dataset containing 21,567 open-vocabulary, crowd-sourced navigation instructions. Finally, we examine the performance of several baseline models, and human performance, on this newly established benchmark.

Chapter 8—Conclusion and Future Directions. We conclude the thesis with a summary of our main contributions and a discussion of future research directions for improving the work.

1.6 List of Publications

Much of the work described in this thesis has been previously published or accepted in conference proceedings, as follows:

- Our SPICE image caption evaluation metric (described in Chapter 4) was published at ECCV [Anderson et al., 2016].
- The bottom-up and top-down visual attention model (described in Chapter 5) has been accepted for publication and full oral presentation at CVPR [Anderson et al., 2018a].
- The constrained beam search approach to out-of-domain image captioning (described in Chapter 6) was published at EMNLP [Anderson et al., 2017].
- The work on vision-and-language navigation (described in Chapter 7) has been accepted for publication and spotlight presentation at CVPR [Anderson et al., 2018b].

The author also contributed to the following projects and publications. This work does not form part of this thesis:

- A video sequence encoding method for activity recognition based on hierarchical rank pooling [Fernando et al., 2016],
- An exploration of architectures and hyperparameters identifying the tips and tricks that lead to the success of our competition-winning VQA model [Teney et al., 2018], and
- A method for predicting accuracy on large datasets from small pilot training datasets [Johnson et al., 2018].

Background

In this chapter we review existing literature relating to image captioning, visual question answering (VQA), and related vision and language tasks, in order to put the contributions of this thesis in context. We also provide a brief technical introduction to the convolutional neural network (CNN) and recurrent neural network (RNN) models that we use as image encoders and language encoders/decoders in later chapters.

2.1 Image Captioning

Research interest in the task of image captioning arguably dates back to the origins of computer vision, when Marvin Minsky asked an undergraduate in 1966 to ‘spend the summer linking a camera to a computer and getting the computer to describe what it saw’ [Boden, 2008]. Although the distinction is becoming increasingly blurred, existing work can be broadly grouped into three categories: (1) template-based image captioning, (2) retrieval-based image captioning, and (3) models that generate novel captions.

In general, template-based natural language generation systems map their non-linguistic input directly (i.e., without intermediate representations) to linguistic surface structures (i.e., templates), which contain gaps or slots that must be filled to generate the final output [Reiter and Dale, 1997]. In template-based image captioning models such as Baby Talk [Kulkarni et al., 2013], slots typically correspond to object and attributes, and they are filled using the output of object detectors and other visual classifiers. In principle, template-based approaches assume that there are a limited number of salient syntactic patterns in descriptive language that can be encoded as templates, although in practice the final template may be the result of a number of consecutive transformations [Deemter et al., 2005], or even the output of a neural network [Lu et al., 2018].

In retrieval-based approaches, captions are produced by first finding similar images, and then copying their captions [Oliva and Torralba, 2006; Farhadi et al., 2010;

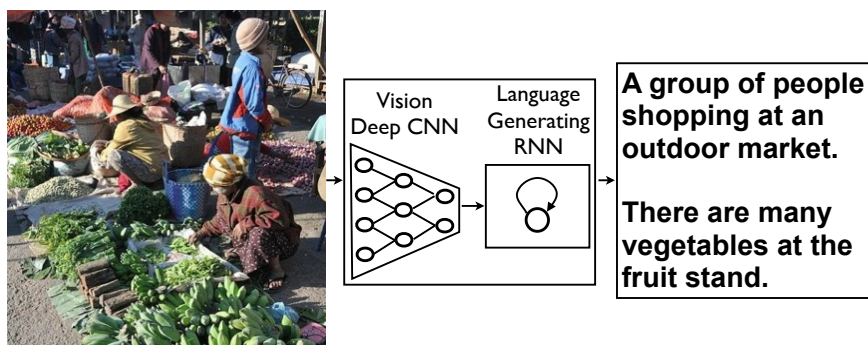


Figure 2.1: A high-level illustration of an encoder-decoder neural network architecture for image captioning (reproduced from [Vinyals et al., 2015]).

Hodosh et al., 2013]. For example, Farhadi et al. [2010] learn to project image and sentence features to a common embedding space, allowing the similarity between images and sentences to be scored. Given a large enough dataset of images and captions, this score can be used to retrieve a caption for a given image, or to obtain images that illustrate a given caption. Naturally, the chances of finding an appropriate caption depend strongly on the size and visual diversity of the underlying dataset, although with large datasets retrieval-based approaches can provide surprisingly strong baselines [Devlin et al., 2015b].

Although models that are capable of generating novel captions also have a long history [Barnard et al., 2003], it was the recent application of deep neural networks to this problem that led a resurgence of interest in this research area. Inspired by advances in automatic machine translation that used encoder-decoder architectures based on recurrent neural networks (RNNs) [Sutskever et al., 2014; Bahdanau et al., 2015a], a number of research groups concurrently proposed encoder-decoder neural network architectures for image captioning [Mao et al., 2015; Vinyals et al., 2015; Karpathy et al., 2014; Xu et al., 2015; Fang et al., 2015; Donahue et al., 2015]. Broadly, these models consist of convolutional neural network (CNN) based image encoders, which are used to extract feature representations from an image, combined with RNN-based language decoders to generate captions, as illustrated in Figure 2.1. We provide additional technical background related to the usage of CNNs as image encoders, and RNNs as language decoders, in Sections 2.3 and 2.4, respectively.

Since the first introduction of encoder-decoder architectures for image captioning, these models have been greatly refined. Many papers have studied different approaches for incorporating visual representations into the language decoder [Wu et al., 2016a; Yao et al., 2017b]. Other works have focused on using reinforcement learning to directly optimise caption quality metrics [Rennie et al., 2017; Liu et al.,

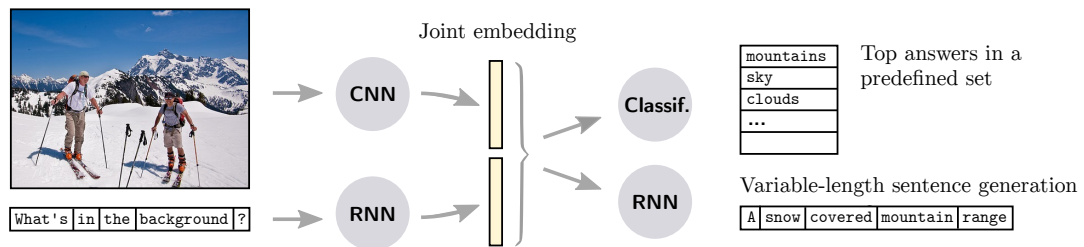


Figure 2.2: A high-level illustration of the joint-embedding approach to VQA, in which CNN and RNN encoders are used to encode the image and the question, respectively (reproduced from [Wu et al., 2017]).

2017a], developing attention mechanisms [Xu et al., 2015; You et al., 2016; Lu et al., 2017] (which we address further in Chapter 5), and improving the diversity and ‘naturalness’ of the captions produced [Dai et al., 2017b]. To support these efforts, a number of image captioning datasets have been collected, of increasing size and complexity [Rashtchian et al., 2010; Hodosh et al., 2013; Young et al., 2014; Lin et al., 2014b]. The main image captioning dataset used in recent work is COCO [Lin et al., 2014b], which is discussed further in Section 3.2.

2.2 Related Vision and Language Tasks

The task of Visual Question Answering (VQA) has also seen substantial interest from the research community, beginning with fairly restricted (sometimes synthetic) settings and small datasets [Bigham et al., 2010; Malinowski and Fritz, 2014; Geman et al., 2015]. More recently, much larger datasets have been introduced [Antol et al., 2015; Goyal et al., 2017], which are discussed further in Section 3.3. As illustrated in Figure 2.2, a common approach to this task involves mapping both the input image and the question to a joint embedding space using CNN and RNN encoders, with the output stage of the model taking the form of a classifier over a set of candidate answers, or an RNN decoder [Wu et al., 2017].

Consistent with this general model structure, much of the recent work in VQA has been focused on improving the performance of the multimodal pooling operation in these models [Fukui et al., 2016], investigating compositional models [Andreas et al., 2016], incorporating visual and/or linguistic attention mechanisms [Yang et al., 2016a; Kazemi and Elqursh, 2017; Lu et al., 2016], and incorporating additional external information into the model from knowledge bases or other sources [Zhu et al., 2015; Wu et al., 2016b]. More broadly, the general problem of reasoning jointly

over visual and linguistic information has been investigated through work on visual grounding [Rohrbach et al., 2016], referring expression generation and comprehension [Kazemzadeh et al., 2014; Mao et al., 2016], video captioning [Venugopalan et al., 2015; Donahue et al., 2015] and visual dialogue [Das et al., 2017], to mention but a few papers from a large body of literature that is largely beyond the scope of this thesis.

Having introduced some of the existing literature relating to image captioning and VQA, we now provide a brief introduction to CNNs and RNNs, which function as the encoder and decoder ‘building blocks’ in these models. For further details we recommend consulting a recent textbook on artificial neural networks such as Goodfellow et al. [2016] and the provided references. Readers who are familiar with these models may skip Sections 2.3 and 2.4.

2.3 Convolutional Neural Networks

Convolutional neural networks (CNNs) [LeCun et al., 1989] are a class of artificial neural network used for processing gridlike data, such as the pixels of an image. More specifically, a CNN is a network that contains one or convolutional layers. These are computational units that produce an output feature map by convolving an input representation with linear kernels containing learned parameters. Each kernel operates independently to produce a 2D response map by sliding across the width and height of the input tensor. The convolution layer output is the concatenation of the individual response maps of multiple kernels, forming a 3D tensor.

Convolutions have several properties that are particularly desirable for image processing [Goodfellow et al., 2016]. First, convolutions are invariant to translation, meaning that objects and other features can be recognised regardless of their position in the image. Second, convolutions achieve efficient parameter sharing since the same kernels are used at every position in the input. Last, interactions are sparse (and therefore efficient) in a convolutional layer because only tensor elements that are nearby will interact through the kernel¹.

To allow information in the individual filter responses of a convolutional layer to be intermingled, a typical CNN contains multiple stacked convolutional layers as illustrated in Figure 2.3. This allows a CNN to learn concepts at an increasing level of abstraction as information is processed through the network, and usually involves interspersing convolutional layers with non-linear activation functions and pooling operations. While various non-linear activation functions have been investigated, the rectified linear unit (ReLU) [Nair and Hinton, 2010] defined by $f(x) = \max(x, 0)$ is

¹This assumes that the kernel is smaller than the input.

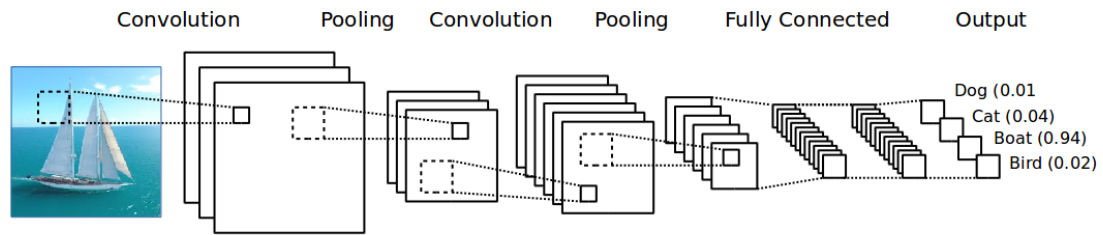


Figure 2.3: An example of a convolutional neural network (CNN) for image classification containing two convolutional layers, two pooling layers, and two fully connected layers. A non-linear activation function (not shown) is typically applied after each convolution layer.

frequently used as it is non-saturating, which can result in faster training [Krizhevsky et al., 2012]. While activation functions operate in an elementwise fashion, pooling functions replace the output of a network at a certain location with a summary statistic—such as the mean or maximum—calculated over a rectangular neighbourhood. This helps to make CNN feature maps approximately invariant to small translations of the input. Pooling functions can also be used to reduce the size of the output representation, increasing the receptive field of the convolutional filters in subsequent layers Goodfellow et al. [2016].

As with most artificial neural networks, an entire CNN can express a single differentiable function. During training, kernel parameters are adapted by backpropagating error gradients from an appropriate loss function defined over the network output [LeCun et al., 1989]. Although the details of CNN loss functions are task specific, it has been observed many times that CNN parameters trained for one task can function as effective visual feature detectors for many other tasks, for example by simply removing the final prediction layer(s) [Huh et al., 2016]. Consistent with this observation, in this thesis we make extensive use of CNNs that have been pre-trained for ImageNet image classification [Russakovsky et al., 2015] as generic image encoders for vision and language tasks such as image captioning.

2.4 Recurrent Neural Networks

Recurrent neural networks (RNNs) are a class of artificial neural network used for processing (either encoding or decoding) sequential data, such as speech or text [Sutskever et al., 2011]. Unlike feedforward neural networks such as CNNs, RNNs contain one or more feedback loops and an internal state (memory) that is updated

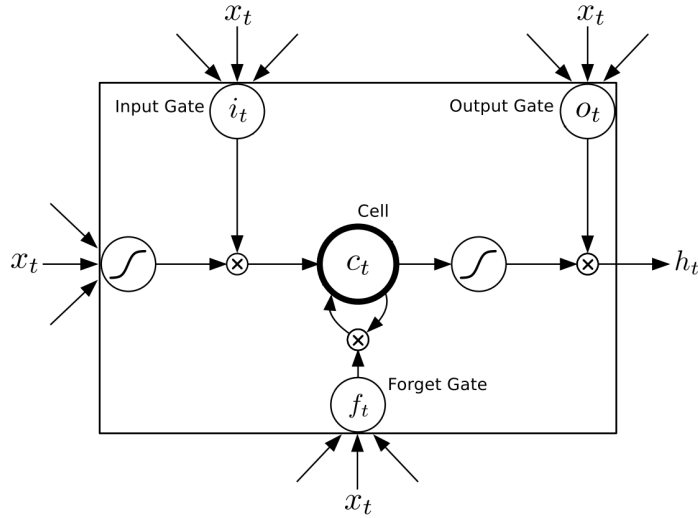


Figure 2.4: Long Short-Term Memory (LSTM) cell, illustrating the operation of the input gate, output gate and forget gate (reproduced from Graves et al. [2013]).

as each element in an input or output sequence is processed. This structure allows the network to process and remember signals, allowing the model to learn sequential dependencies in data.

2.4.1 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) networks [Hochreiter and Schmidhuber, 1997] are a particular RNN implementation that is capable of learning long-term dependencies. In this thesis, we make extensive use of LSTM networks as language encoders or decoders within larger models in Chapters 5, 6 and 7. In the chapters that follow we will refer to the operation of the LSTM over a single time step using the following notation:

$$\mathbf{h}_t = \text{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1}) \quad (2.1)$$

where \mathbf{x}_t is an input vector to the LSTM, representing one element from a sequence of input vectors $\{\dots, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \dots\}$, and \mathbf{h}_t is the LSTM output vector which will also be one element from an output sequence $\{\dots, \mathbf{h}_{t-1}, \mathbf{h}_t, \mathbf{h}_{t+1}, \dots\}$.

The full feed-forward operation and hidden state update of each LSTM layer is illustrated in Figure 2.4, and can be described as follows. Assuming N hidden units within each LSTM layer, an N -dimensional input gate \mathbf{i}_t , forget gate \mathbf{f}_t , output gate

\mathbf{o}_t , and input modulation gate \mathbf{g}_t at timestep t are updated as:

$$\mathbf{i}_t = \text{sigm}(W_{xi}\mathbf{x}_t + W_{hi}\mathbf{h}_{t-1} + \mathbf{b}_i) \quad (2.2)$$

$$\mathbf{f}_t = \text{sigm}(W_{xf}\mathbf{x}_t + W_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f) \quad (2.3)$$

$$\mathbf{o}_t = \text{sigm}(W_{xo}\mathbf{x}_t + W_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o) \quad (2.4)$$

$$\mathbf{g}_t = \text{tanh}(W_{xc}\mathbf{x}_t + W_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \quad (2.5)$$

where $x_t \in \mathbb{R}^K$ is the input vector, $h_t \in \mathbb{R}^N$ is the LSTM output, W 's and \mathbf{b} 's are learned weights and biases, and $\text{sigm}(\cdot)$ and $\text{tanh}(\cdot)$ are the sigmoid and hyperbolic tangent functions, respectively, applied element-wise. The above gates control the memory cell activation vector $\mathbf{c}_t \in \mathbb{R}^N$ and output $\mathbf{h}_t \in \mathbb{R}^N$ of the LSTM as follows:

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \quad (2.6)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \text{tanh}(\mathbf{c}_t) \quad (2.7)$$

where \odot represents element-wise multiplication. Note that both the previous LSTM output vector \mathbf{h}_{t-1} and the previous LSTM memory cells \mathbf{c}_{t-1} are taken as inputs in the current time step. However, in Equation 2.1 we have neglected the propagation of memory cells for notational convenience.

2.4.2 LSTM Encoders

When functioning as an encoder, the encoded representation of some input sequence $\{x_0, \dots, x_T\}$ is usually taken to be the final LSTM output vector \mathbf{h}_T , or the entire sequence of outputs $\{\mathbf{h}_0, \dots, \mathbf{h}_T\}$ if an attention mechanism is used (refer Section 5.1). As the LSTM input x_t is a vector, when LSTMs are required to process discrete input tokens such as words or characters from text, an input word embedding matrix can be used. In this case, the LSTM input vector will be a—possibly learned—encoding of a discrete token, given by:

$$\mathbf{x}_t = W_e \Pi_t \quad (2.8)$$

where W_e is a word embedding matrix, and Π_t is a one-hot column vector identifying the input token at timestep t .

2.4.3 LSTM Decoders

When functioning as a decoder, for example as a language decoder generating discrete output sequences $\mathbf{y} = (y_1, \dots, y_T)$ containing words or other tokens from vocab-

ulary Σ , an LSTM can be augmented with an output projection layer and a softmax. For example, in Section 5.2.2 at each time step t the conditional distribution over possible output words is given by:

$$p(y_t | y_{1:t-1}) = \text{softmax}(W_p \mathbf{h}_t + \mathbf{b}_p) \quad (2.9)$$

where $W_p \in \mathbb{R}^{|\Sigma| \times N}$ and $\mathbf{b}_p \in \mathbb{R}^{|\Sigma|}$ are learned weights and biases, and the softmax function is defined as:

$$\text{softmax}(\mathbf{x})_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad (2.10)$$

Denoting the full set of model parameters by θ , the distribution over complete output sequences can be calculated as the product of conditional distributions:

$$p_\theta(\mathbf{y}) = \prod_{t=1}^T p_\theta(y_t | y_{1:t-1}) \quad (2.11)$$

Typically, in a decoder formulation the LSTM input x_t will be defined to include the previously generated output y_{t-1} , but in general the formulation of the inputs x and the first hidden state \mathbf{h}_0 is application specific. In recent years a large variety of sequence prediction problems have been formulated using LSTM decoders by appropriately defining x and \mathbf{h}_0 , including dependency parsing [Kiperwasser and Goldberg, 2016], language modelling [Sutskever et al., 2011], machine translation [Sutskever et al., 2014], image captioning [Vinyals et al., 2015] and visual question answering (VQA) [Zhu et al., 2016].

As with CNNs, RNNs used for generating sequences can be trained in a supervised fashion, typically using cross-entropy loss. Given a set of training sequences D , the cross-entropy loss maximises the probability of the observed data by minimising:

$$\mathcal{L}_{XE}(\theta) = -\frac{1}{|D|} \sum_{\mathbf{y}^* \in D} \log p_\theta(\mathbf{y}^*) \quad (2.12)$$

$$= -\frac{1}{|D|} \sum_{\mathbf{y}^* \in D} \sum_{t=1}^T \log p_\theta(y_t^* | y_{1:t-1}^*) \quad (2.13)$$

As illustrated in Equation 2.13, this formulation treats the sequence prediction task as a single-step supervised learning problem. The RNN is trained to predict the next token following a given a partial sequence, and every decision is weighted equally.

Given an RNN modelling a probability distribution over output sequences, we may wish to find the output sequence with the maximum log-probability. This is

known as the RNN decoding problem. As in Equation 2.13 the log probability of any partial sequence \mathbf{y}_t of length t is typically given by:

$$\log p_\theta(\mathbf{y}_t) = \sum_{j=1}^t \log p_\theta(y_j | y_{1:j-1}) \quad (2.14)$$

As it is computationally infeasible to solve this problem exactly, beam search [Koehn, 2010] is widely used to find an approximate solution. At each decoding time step t , beam search stores only the the b most likely partial sequences, where b is known as the beam size. We will denote the set of all partial solutions held at the start of time t by $B_{t-1} = \{\mathbf{y}_{t-1,1}, \dots, \mathbf{y}_{t-1,b}\}$. At each time step t , a candidate set E_t is generated by considering all possible next word extensions:

$$E_t = \{(\mathbf{y}_{t-1}, w) \mid \mathbf{y}_{t-1} \in B_{t-1}, w \in \Sigma\} \quad (2.15)$$

The beam B_t is updated by retaining only the b most likely sequences in E_t . This can be trivially implemented by sorting the partial sequences in E_t by their log-probabilities and retaining the top b . Initialisation is performed by inserting an empty sequence into the beam, i.e. $B_0 := \{\epsilon\}$ such that $E_1 = \Sigma$. The algorithm terminates when the beam contains a completed sequence (e.g., containing an end marker) with higher log probability than all incomplete sequences.

Datasets

In computer vision and natural language processing, high-quality datasets—along with well-specified tasks and evaluation protocols—have been crucial to advancing the state of the art. In this section, we introduce the main existing vision and language datasets that are used in this thesis. Our proposed dataset for visually-grounded natural language navigation in real buildings—the Room-to-Room (R2R) dataset—is discussed in Chapter 7.

3.1 ImageNet

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC or ImageNet) [Russakovsky et al., 2015] is a large-scale benchmark dataset and challenge for image classification and object detection. The dataset contains photographs collected from Flickr¹ and other search engines, manually annotated by workers from the Amazon Mechanical Turk² (AMT) crowd-sourcing platform. The challenge has been run each year from 2010 until its final year in 2017, with the size and format of the dataset expanding over time. Various annotations are available, as follows:

- **Image Classification:** The classification dataset consists of 1.4M images manually labelled with the presence of one of 1000 fine-grained object categories, including a huge variety of both natural and man-made visual phenomena. As illustrated in Figure 3.1, each image contains one ground-truth label. The final ILSVRC2012 version of this dataset consists of 1,281K training images, 50K validation images, and 100K test images.
- **Single-Object Localisation:** For single-object localisation, 524K training images plus the validation and test images from the classification dataset are additionally annotated with an axis-aligned bounding box for every instance of

¹www.flickr.com

²www.mturk.com



Figure 3.1: Examples from the ImageNet image classification dataset, illustrating the variety of fine-grained object classes.

the ground-truth object category. The object classes are the same as the object classes for the image classification task.

- **Object Detection:** The object detection dataset contains additional images collected from Flickr using scene-level (rather than object-level) queries. Each image is annotated with a list of object categories present in the image, along with an axis-aligned bounding box indicating the position and scale of each instance of each object category. There are 200 object classes, representing basic-level object categories, and 457K training images, 21K validation images and 40K test images.

In this thesis, we will make extensive use of convolutional neural networks (CNNs) that have been pretrained on the ImageNet image classification task. Learning good general-purpose image feature representations on ImageNet and then transferring these models to other tasks has become a common practice [Huh et al., 2016].

3.2 COCO

COCO [Lin et al., 2014b], short for Common Objects in COntext, is a large-scale object detection, segmentation and captioning dataset. The dataset contains Flickr photographs depicting examples of 91 preselected object categories, such as ‘person’, ‘dog’, ‘cow’, ‘train’, ‘car’, ‘motorbike’, ‘chair’, ‘sofa’ and ‘bottle’. Unlike previous object detection datasets such as PASCAL VOC [Everingham et al., 2010], images were deliberately selected to contain multiple objects viewed from non-canonical perspectives. In total, 328K images were collected, of which 204K have been annotated using AMT.

In the 2014/15 release, the dataset was split into 83K training images, 41K validation images, 41K test 2014 images, 81K test 2015 images, and 123K unlabelled images. However, based on community feedback, in 2017 the 80K/40K train/val split was revised to 115K/5K for train/val. Test images are divided equally into test-dev, test-standard, test-challenge, and test-reserve subsets.

Several different annotation formats are available, as described below and illustrated in Figure 3.2. In each case annotations were collected using AMT.

- **Object Segmentations:** Images in the training and validation sets, as well as the 2015 test set are annotated with per-instance object segmentations for 80 of the 91 object categories. Segmentations were not collected for 11 categories for which instances were too common, too rare, or too easily confused. On average each image contains 3.5 object categories and 7.7 object instances. In total, 1.5M objects were annotated. To support object detection pipelines based on image bounding boxes, tight-fitting bounding boxes were also obtained from the annotated object segmentation masks.
- **Image Captions:** Images in the training and validation sets, as well as the 2014 test set, are annotated with image captions. To collect the captions, AMT workers were instructed to provide a sentence that describes all of the important parts of the scene, using at least eight words. Five captions were collected for each image, except for a random subset of 5K test images for which 40 captions were collected per image (comprising the C40 test set, as differentiated from the C5 test set). Finally, for each testing image, one additional caption was collected to quantify human performance when comparing scores with machine generated captions. In total 1M captions were collected [Chen et al., 2015].
- **Other:** More recently, several additional annotations have been added to the dataset including person keypoints and ‘stuff’ segmentations identifying amorphous background regions such as grass and sky.



Figure 3.2: Examples of annotated images in the COCO dataset. Segmentations are illustrated only for the indicated class, although typically multiple objects are annotated in each image. Captions are numbered 1–5.

To ensure consistency in the evaluation of captioning agents, access to the test sets is restricted and a test evaluation server is maintained by the COCO organisers, providing standard automatic evaluation metrics. In the case of image captioning, the various evaluation metrics used are discussed in Chapter 4. Finally, we note that in the case of image captioning, an alternative partition of the official training and validation split has been extensively reported in prior work for ablation studies and offline testing. Known as the ‘Karpathy’ split after its proposers [Karpathy and Fei-Fei, 2015], this split contains 113K training images, and 5K images respectively for validation and testing.

3.3 VQA

3.3.1 Version 1.0

The Visual Question Answering (VQA) dataset [Antol et al., 2015] is a large-scale dataset containing free-form, open-ended, natural language questions and answers about images, as illustrated by the examples in Figure 3.3. The dataset (now referred to as v1.0) contains 614K questions and 6.1M answers associated with 205K images. The images used are the 123K training and validation images and 81K test images from the COCO 2014/15 dataset [Lin et al., 2014b], following the same train/val/test split strategy.

To create the dataset, questions and answers were collected in separate stages using AMT. To bias against generic image-independent questions, workers were instructed to ask questions that require the image to answer. In order to encourage the submission of interesting and diverse questions, AMT workers were instructed to ask a question that a smart robot ‘probably can not answer, but any human can easily answer while looking at the scene in the image’. For each question, 10 ground-truth answers were collected from different AMT workers. Since most questions are quite specific, most answers consist of simple one to three word phrases.

Two modalities are offered for answering questions, multiple choice and open-ended. To evaluate the generated answers for the open-ended task, answers are preprocessed to standardise case, numbers and punctuation, then the following accuracy metric is used:

$$\text{accuracy} = \min\left(\frac{\# \text{ humans that provided that answer}}{3}, 1\right) \quad (3.1)$$

i.e., an answer is deemed to be 100% accurate if at least three out of 10 AMT workers provided the same answer.

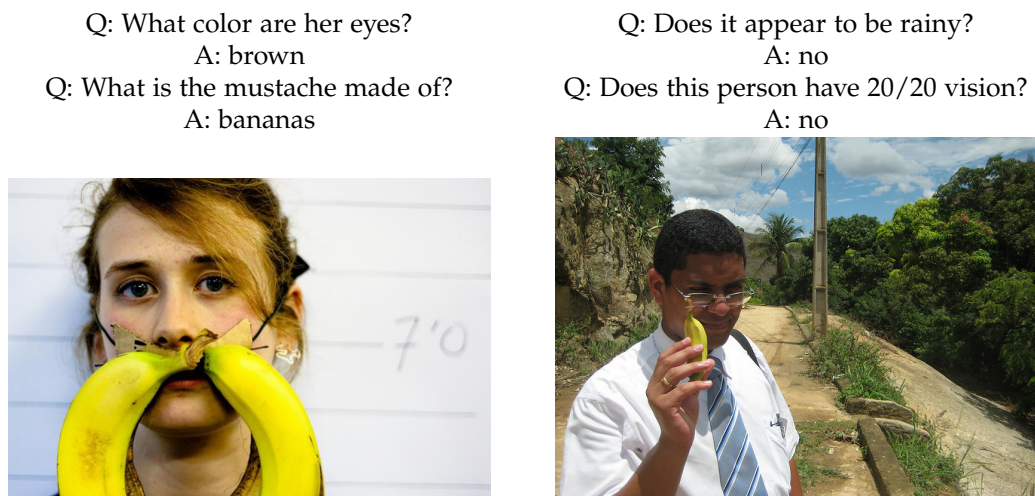


Figure 3.3: Examples of questions (Q) and answers (A) from the VQA v1.0 dataset. Although the questions are primarily visual in nature, some also require ‘common-sense’ knowledge, e.g. ‘Does this person have 20/20 vision?’.

3.3.2 Version 2.0

Several works pointed out that questions in the v1.0 dataset provide strong priors that can result in good performance, even if the visual content of the image is ignored [Agrawal et al., 2016; Jabri et al., 2016; Zhang et al., 2016a]. The Visual Question Answering (VQA) dataset v2.0 [Goyal et al., 2017] subsumes and extends the v1.0 dataset in order to reduce these biases in the answer distributions. Specifically, the VQA v2.0 dataset was created in the following way—given an (image, question, answer) triplet from the VQA v1.0 dataset, AMT workers were asked to identify a new image that is similar to the original but results in a different correct answer. This balances the answer distributions in the dataset such that each question in VQA v2.0 is associated with not just a single image, but rather a pair of similar images that result in two different answers to the question, as illustrated in Figure 3.4. In total the VQA v2.0 dataset, which was used for the 2017 VQA Challenge³, contains 1.1M questions and 11.1M answers associated with the same 205K images as the v1.0 dataset.

3.4 Visual Genome

The Visual Genome dataset [Krishna et al., 2016] contains 108K images annotated with scene graphs as well as 5.4M region descriptions and 1.7M visual question

³www.visualqa.org/challenge.html

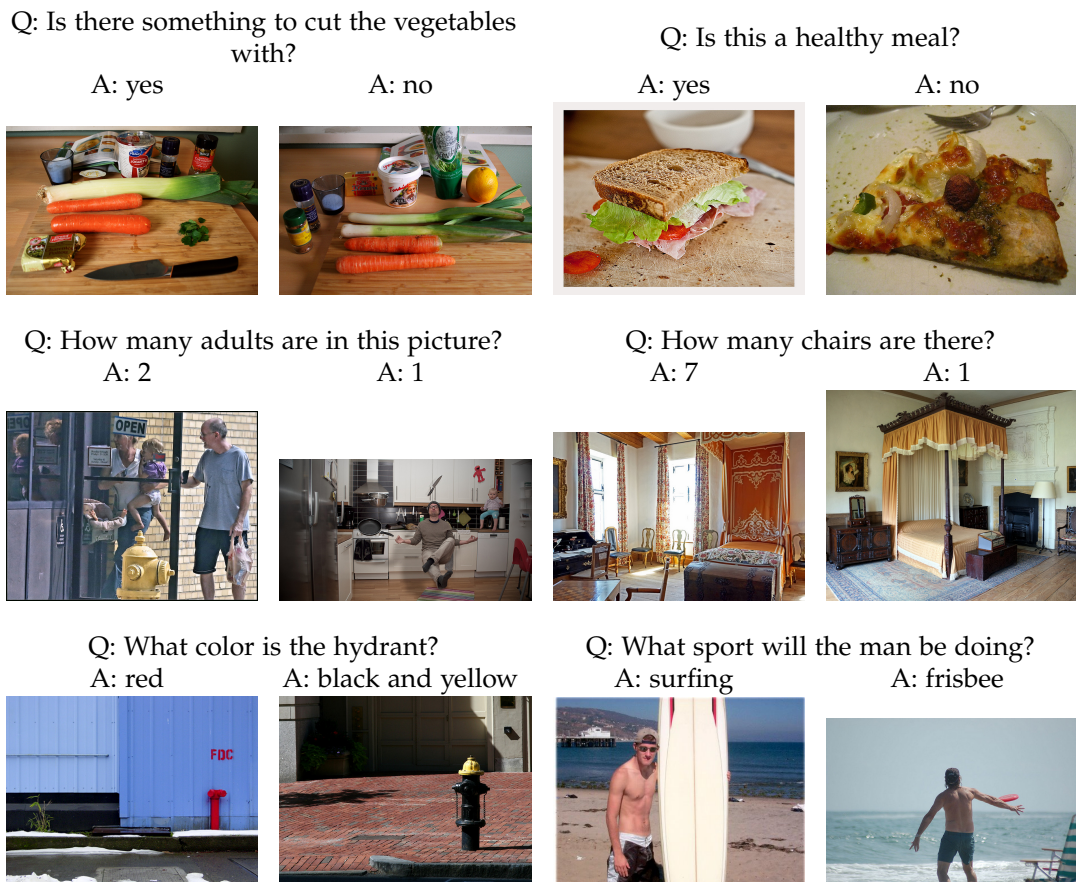


Figure 3.4: Examples of questions (Q) and answers (A) from the VQA v2.0 dataset illustrating images with different answers to the same question, as well as the different question types, i.e. Yes/No (top), Number (middle), and Other (bottom).

answers. Scene graph annotations consist of grounded objects, attributes associated with objects, and pairwise relationships between objects. Each image has an average of 35 objects, 26 attributes and 21 relationships. An example is illustrated in Figure 3.5.

To create the dataset, all creative commons images were selected from the intersection of COCO’s [Lin et al., 2014b] 328K images and the 100M images in the YFCC100M dataset [Thomee et al., 2016]. As a result, approximately 51K of COCO’s 123K training and validation images are found in Visual Genome. All annotations were collected using AMT, starting with region descriptions from which objects, attributes and relations were extracted. Note that unlike object detection datasets, objects and attribute annotations in Visual Genome are freely annotated strings, rather than class labels. However, effort has been made to map annotations to synsets in the WordNet [Miller, 1995] ontology.

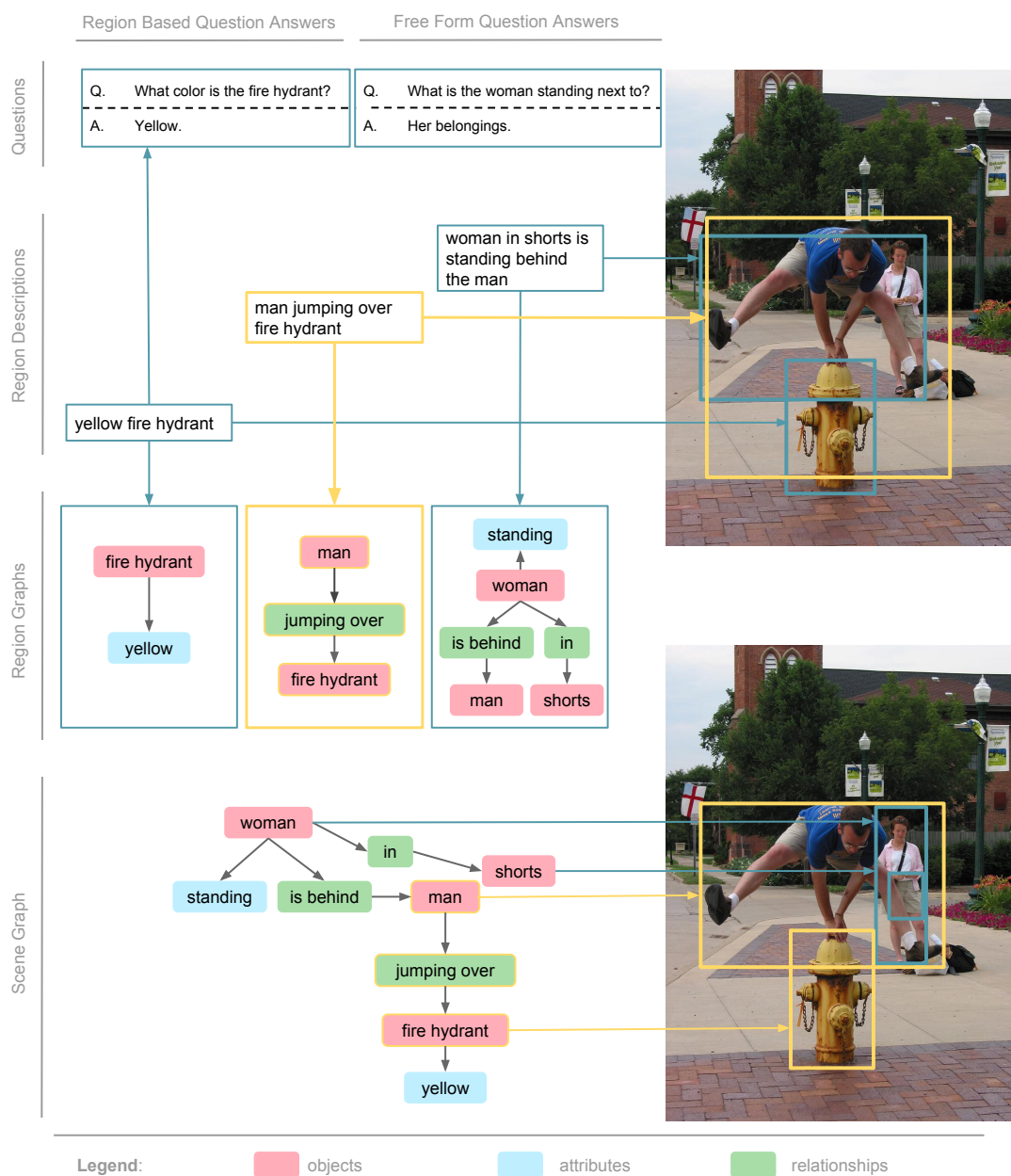


Figure 3.5: An example image annotation from the Visual Genome dataset (reproduced from Krishna et al. [2016]), illustrating questions and answers, region descriptions and grounded scene graphs containing objects, attributes and relations.

SPICE: Semantic Propositional Image Caption Evaluation

In this chapter, we address the problem of automatically evaluating image captions. While new datasets often spur considerable innovation—as has been the case with the COCO Captioning Challenge [Chen et al., 2015]—benchmark datasets also require fast, accurate and inexpensive evaluation metrics to facilitate rapid progress. Although numerous automatic evaluation metrics have already been applied to the task of evaluating image captions, existing metrics have proven to be inadequate substitutes for human judgement [Kulkarni et al., 2013; Hodosh et al., 2013; Elliott and Keller, 2014]. As such, there is an urgent need to develop new automated evaluation metrics for this task [Elliott and Keller, 2014; Bernardi et al., 2016]. To address this problem, we present a novel automatic image caption evaluation metric that measures the quality of generated captions by analysing their semantic content. Our method more closely resembles human judgement while offering the additional advantage that the performance of any model can be analysed in greater detail than with other automated metrics.

4.1 Image Caption Evaluation

Given a candidate caption y and a set of high-quality reference captions $S = \{s_1, \dots, s_m\}$ associated with an image I , our goal is to evaluate the quality of y given I . Typically, the source image I is disregarded during this evaluation, which is performed by computing a score that captures the similarity between y and S . This approach poses caption evaluation as a purely linguistic task that is similar to evaluating text summaries or machine translations (MT). As such, a number of textual similarity metrics designed for other tasks have been applied to image caption evaluation, including Bleu [Papineni et al., 2002], METEOR [Denkowski and Lavie, 2014] and ROUGE [Lin,

Metric	Proposed Application	Principle
Bleu	Machine Translation	n-gram precision
ROUGE	Document Summarization	n-gram recall
METEOR	Machine Translation	n-gram with synonym matching
CIDEr	Image Captions	tf-idf weighted n-gram cosine similarity
SPICE (ours)	Image Captions	scene graph synonym matching

Table 4.1: A summary of textual similarity metrics.

2004]. More recently, CIDEr [Vedantam et al., 2015] has been proposed specifically for the image captioning setting.

As summarised in Table 4.1, existing metrics rely on matching or aligning n-grams¹ between the candidate and reference texts. Bleu [Papineni et al., 2002] is an n-gram precision metric with a sentence-brevity penalty, calculated as a weighted geometric mean over different length n-grams. ROUGE [Lin, 2004] is a package of a measures for automatic evaluation of text summaries. ROUGE-L, the version commonly used in the image captioning community, is an F-measure based on words in the longest common subsequence between candidate and reference sentences. METEOR [Denkowski and Lavie, 2014] works by first performing an alignment between words in the candidate and reference sentences, taking into account exact matches as well as soft-similarity based on stem, synonym and paraphrase matches. The METEOR score is then computed as a parameterised harmonic mean of precision and recall with an added alignment fragmentation penalty. CIDEr [Vedantam et al., 2015] is based on a term frequency-inverse document frequency (tf-idf) weighting of each n-gram in the candidate and reference sentences, which are then compared by summing their cosine similarity across n-grams.

Several studies have investigated the validity of these metrics when used for image caption evaluation by reporting correlation measures between these metrics and human judgements of caption quality. On the PASCAL 1K dataset, Bleu-1 was found to exhibit weak or no correlation with human judgements (Pearson’s r of -0.17 and 0.05 for captions generated by a language model and a template-based model respectively) [Kulkarni et al., 2013]. Using the Flickr 8K [Hodosh et al., 2013] dataset, Elliott and Keller [2014] found that METEOR exhibited moderate correlation (Spearman’s ρ of 0.524) with human judgements, outperforming ROUGE SU-4 (0.435) and Bleu-1 (0.345). Finally, using the PASCAL-50S and ABSTRACT-50S datasets, Vedantam et al. [2015] demonstrated that CIDEr and METEOR have a greater agreement with human consensus than Bleu and ROUGE.

¹An n-gram is defined as a contiguous sequence of n words.

4.2 SPICE Metric

One of the problems with using metrics based on n-grams such as Bleu, ROUGE, CIDEr or METEOR to evaluate captions, is that *n-gram overlap is neither necessary nor sufficient for two sentences to convey the same meaning* [Giménez and Màrquez, 2007]. To illustrate the limitations of n-gram comparisons, consider the following two captions (a,b) from the COCO dataset [Chen et al., 2015] (emphasis added):

- (a) A young girl **standing on top of a tennis court**.
- (b) A giraffe **standing on top of a green field**.

The captions describe two very different images. However, comparing these captions using any of the previously mentioned n-gram metrics produces a high similarity score due to the presence of the long 5-gram phrase ‘standing on top of a’ in both captions. Now consider the captions (c,d) that describe the same image:

- (c) A shiny metal pot filled with some diced veggies.
- (d) The pan on the stove has chopped vegetables in it.

These captions convey almost the same meaning, but exhibit low n-gram similarity as they have no words in common. N-gram approaches suffer from serious deficiencies, particular when applied at the sentence level rather than the corpus level.

To help overcome these limitations, we re-frame the caption evaluation problem in terms of natural language semantics. As image captions are almost always assertions rather than performatives, i.e., they describe or report some state of affairs [Austin, 1962], they ought to be truth-conditionally verifiable. We therefore adopt a truth-conditional approach to semantics, reducing the meaning of a sentence to the truth-conditions of atomic propositions. In other words, we hypothesise that when people evaluate the quality of image captions, they *primarily consider the truth-value of the claims about the image contained therein*. For example, given an image with the caption ‘A young girl standing on top of a tennis court’, a conscientious human evaluator would interpret it as asserting the following atomic propositions²: (1) there is a girl, (2) the girl is young, (3) the girl is standing, (4) there is a court, (5) the court is for tennis, and (6) the girl is on top of the court. If each of these propositions is true—in the sense that it is true in the situation depicted by the image—we argue that many people would consider the caption to be acceptable³.

²Note that these propositions can be more formally represented by a predicate combined with an appropriate number of arguments, or equivalently with a tuple.

³In making this argument we don’t consider the *saliency* of the propositions contained in the caption. However, in the actual SPICE metric saliency is implicitly captured by recall, i.e. caption candidates are penalised if they do not mention relevant propositions contained in the reference captions.

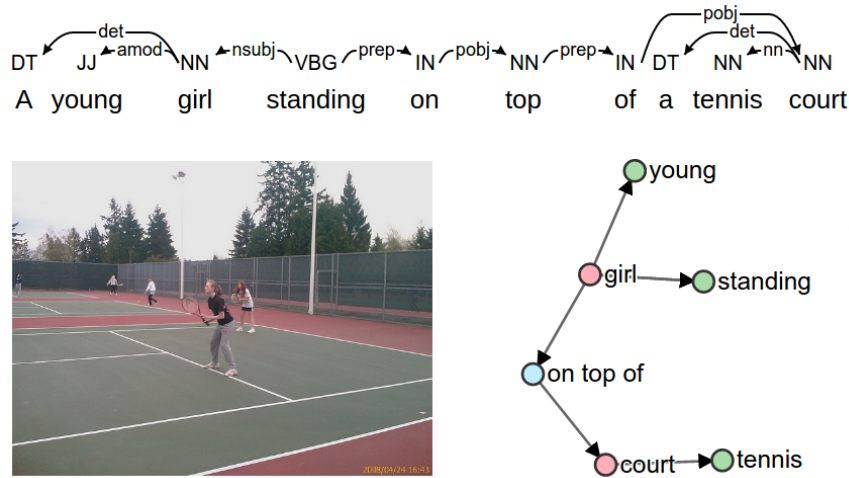


Figure 4.1: SPICE uses semantic propositional content to assess the quality of image captions. Both reference and candidate captions are mapped through dependency parse trees (top) to semantic *scene graphs* (right)—encoding the objects (red), attributes (green), and relations (blue) present. Caption quality is determined using an F-score calculated over tuples in the candidate and reference scene graphs

Taking this main idea as motivation, we estimate caption quality by transforming both candidate and reference captions into a semantic representation. Our choice of semantic representation is the *scene graph*. As illustrated in Figure 4.1, our scene graph implementation explicitly encodes the objects, attributes and relations found in image captions, abstracting away most of the lexical and syntactic idiosyncrasies of natural language in the process. While we are aware that there are other components of linguistic meaning—such as figure-ground relationships—that are almost certainly relevant to caption quality, in this work we focus exclusively on semantic meaning. We choose to use a scene graph representation as scene graphs and similar semantic structures have been used in a number of recent works within the context of image and video retrieval systems to improve performance on complex queries [Lin et al., 2014a; Johnson et al., 2015; Schuster et al., 2015]. Several of these papers have demonstrated that semantic graphs can be parsed from natural language descriptions [Lin et al., 2014a; Schuster et al., 2015].

Given candidate and reference scene graphs, our metric computes an F-score defined over the conjunction of logical tuples representing propositions in the scene graph. We dub this approach SPICE for *Semantic Propositional Image Caption Evaluation*. We now describe each component of our approach in detail.

4.2.1 Semantic Parsing from Captions to Scene Graphs

Following Schuster et al. [2015], we define the subtask of parsing captions to scene graphs as follows. Given a set of object classes \mathcal{C} , a set of attribute types \mathcal{A} , a set of relation types \mathcal{R} , and a caption \mathbf{y} , parse \mathbf{y} to a scene graph G defined by:

$$G = (O, E) \quad (4.1)$$

where $O = \{o_1, \dots, o_n\}$ is the set of objects mentioned in \mathbf{y} , and each object is represented as a pair $o_i = (c_i, A_i)$ where $c_i \in \mathcal{C}$ is the object class and $A_i \subseteq \mathcal{A}$ are the attributes of o_i . $E \subseteq O \times \mathcal{R} \times O$ is the set of relations between pairs of objects in the graph. Note that in practice, \mathcal{C} , \mathcal{A} and \mathcal{R} are open-world sets that expand as new object, relation and attribute types are identified, placing no restriction on the types of objects, relation and attributes that can be represented. We allow both count noun objects, such as ‘chair’ and ‘bottle’, as well as mass noun objects such as ‘grass’ and ‘sky’.

Before introducing our approach to parsing, we note that any scene graph can be equivalently represented by a conjunction of logical propositions, or tuples. We define the invertible function \mathcal{T} that returns tuples from a scene graph as:

$$\mathcal{T}(G) = \text{obj}(G) \cup \text{attr}(G) \cup \text{rel}(G) \quad (4.2)$$

$$\text{obj}(G) = \{(c_i) \mid o_i \in O\} \quad (4.3)$$

$$\text{attr}(G) = \{(c_i, a) \mid o_i \in O, a \in A_i\} \quad (4.4)$$

$$\text{rel}(G) = \{(c_i, r, c_j) \mid (o_i, r, o_j) \in E\} \quad (4.5)$$

where $\text{obj}(G)$, $\text{attr}(G)$ and $\text{rel}(G)$ are functions that return object tuples containing one element, object-attribute tuples containing two elements, and object-relation-object tuples containing three elements, respectively. For example, the scene graph in Figure 4.1 maps to the following tuples:

$$\{ (\text{girl}), (\text{court}), (\text{girl}, \text{young}), (\text{girl}, \text{standing}), \\ (\text{court}, \text{tennis}), (\text{girl}, \text{on top of}, \text{court}) \}$$

As illustrated by this example, object classes, attribute types and relation types may be single words or short phrases.

To parse a caption to a scene graph, we first establish syntactic dependencies between words in the caption using a Probabilistic Context-Free Grammar (PCFG) dependency parser [Klein and Manning, 2003] pretrained on a large, independent dataset. An example of the resulting dependency syntax tree, using Universal Dependency relations [De Marneffe et al., 2014], is shown in Figure 4.1 top. To map

from dependency trees to scene graphs (Figure 4.1 right), we adopt a variant of the rule-based version of the Stanford Scene Graph Parser Schuster et al. [2015] that first modifies the dependency tree and then directly extracts scene graph tuples.

In its original implementation, the Stanford Scene Graph Parser performs three post-processing steps to modify the dependency tree. These steps simplify quantificational modifiers such as ‘a lot of’, resolve pronouns such as ‘it’ and apply a distributive reading to all plural nouns such as ‘three men’, creating multiple copies of these nodes according to the value of their numeric modifier. The resulting graph structure is then parsed according to nine simple linguistic patterns to extract object mentions, object-attribute pairs and object-relation-object tuples, which are added to the scene graph. As an example, one of the linguistic patterns captures adjectival modifiers, such as *young* $\xleftarrow{\text{amod}}$ *girl*, generating the object mention ‘girl’ with attribute ‘young’ that is illustrated in Figure 4.1. Full details of the pipeline can be found in the original paper.

Our implementation differs from the original parser in several respects. The most significant difference occurs in regard to the treatment of plural nouns such as ‘three men’. We do not duplicate these nodes in the dependency graph or the resulting scene graph. In previous work, duplication of object instances was desirable to enable scene graphs to be grounded to image regions in an image retrieval setting Johnson et al. [2015]; Schuster et al. [2015]. In our work, we choose to encode numeric modifiers (object counts) in the scene graph as attributes of a single object. While this approach is like the original in that it does not distinguish collective and distributive readings, it simplifies scene graph alignment and ensures that each incorrect numeric modifier in a caption will only be counted as a single error.

In addition to the changed treatment of plural nouns, we also add three additional linguistic patterns to the parser that reflect other differences between the original image retrieval setting and ours. First, to extract as much information as possible, we add an additional linguistic rule that ensures that nouns will always appear as objects in the scene graph, even if no associated relations can be identified. While disconnected graph nodes may have been problematic in the image retrieval setting, they are easily handled by our semantic proposition F-score calculation. Second, to improve scene graph matching between captions, we encode compound nouns such as ‘tennis court’ as objects (‘court’) with attributes (‘tennis’). This prevents the scenario in which ‘court’ and ‘tennis court’ are treated as different objects and a caption unfairly penalised. Third, we add a pattern that applies a distributive reading to nouns separated by a conjunction. For example, given the caption ‘a man and a woman eating chips’, this pattern ensures that in the resulting scene graph both the man and the woman will have an ‘eating’ relation with the chips.

Notwithstanding the use of the Stanford Scene Graph Parser, our proposed SPICE metric is not tied to this particular parsing pipeline. In fact, it is our hope that ongoing advances in syntactic and semantic parsing will allow SPICE to be further improved in future releases. We also note that because SPICE operates on scene graphs, in principle it could be used to evaluate generated captions for images in datasets that contain reference scene graphs [Schuster et al., 2015; Johnson et al., 2015; Plummer et al., 2015; Krishna et al., 2016] even in the absence of any actual reference captions. However, we leave evaluation of SPICE under these circumstances to future work.

4.2.2 F-score Calculation

Having presented our approach to parsing a caption to a scene graph, we now describe the calculation of the SPICE metric. Given a candidate scene graph G_y and a reference scene graph G_S , we define SPICE as a balanced F-score (equally weighting precision P and recall R) calculated over matching tuples in G_y and G_S :

$$SPICE(G_y, G_S) = F_1(T_y, T_S) = \frac{2 \cdot P(T_y, T_S) \cdot R(T_y, T_S)}{P(T_y, T_S) + R(T_y, T_S)} \quad (4.6)$$

where $T_i = \mathcal{T}(G_i)$. Two questions remain. First, given the parser described in Section 4.2.1, how can we best parse the set of reference captions S to a single unified scene graph G_S , in order to pool the available information? Second, how do we best define precision P and recall R to account for captions that refer to the same objects, attributes or relations using synonyms?

To provide context to these questions, consider the following example in which two reference captions $S = \{s_1, s_2\}$ will be used to evaluate candidate caption y . We also provide the correctly parsed scene graph tuple representation for each caption:

s_1 : A young girl standing on top of a tennis court.

$$T_{s_1} = \{ (girl), (court), (girl, young), (girl, standing), (court, tennis), (girl, on top of, court) \}$$

s_2 : A woman standing on top of a tennis court holding a tennis racquet.

$$T_{s_2} = \{ (woman), (racquet), (court), (woman, standing), (racquet, tennis), (court, tennis), (woman, on top of, court), (woman, holding, racquet) \}$$

y : A young lady holding a tennis racket.

$$T_y = \{ (lady), (racket), (lady, young), (racket, tennis), (lady, holding, racket) \}$$

Suppose we unify the reference scene graph by naively taking the union of all tuples

$T_S = T_{s_1} \cup T_{s_2}$, such that:

$$T_S = \{ (\text{girl}), (\text{woman}), (\text{racquet}), (\mathbf{court}), (\text{girl, young}), (\text{racquet, tennis}), \\ (\mathbf{girl, standing}), (\mathbf{woman, standing}), (\mathbf{court, tennis}), (\mathbf{girl, on top of, court}), \\ (\mathbf{woman, on top of, court}), (\text{woman, holding, racquet}) \}$$

Note that in the context of reference captions s_1 and s_2 , candidate caption y appears to be quite accurate. Yet, if we now define precision and recall in the standard way, i.e., $P(T_y, T_S) = \frac{|T_y \cap T_S|}{|T_y|}$ and $R(T_y, T_S) = \frac{|T_y \cap T_S|}{|T_S|}$, the candidate caption y scores zero for both precision and recall, and therefore zero for the SPICE metric. This occurs because there are no exact matches between T_y and T_{s_1} or T_{s_2} .

Motivated by this problem, we introduce a notion of synonymy. Using the METEOR implementation [Denkowski and Lavie, 2014], we consider two words or phrases to be synonymous if their lemmatised word forms are equal (allowing terms with different inflectional forms to match), if they share membership in any synonym set according to WordNet [Miller, 1995], or if they are listed as paraphrases in an appropriate paraphrase table. To apply synonym matching to tuples, we define the binary matching operator \otimes as the function that returns the scene graph tuples in a set T_A that match one or more tuples in a second set T_B :

$$T_A \otimes T_B = \{ \mathbf{u} \mid \mathbf{u} \in T_A, \mathbf{v} \in T_B, \mathbf{u} \sim \mathbf{v} \} \quad (4.7)$$

where $\mathbf{u} \sim \mathbf{v}$ means that tuples \mathbf{u} and \mathbf{v} have the same length and all corresponding elements are synonyms. We make no allowance for partial credit if one or more elements of a tuple are incorrect. In the domain of image captions, many relations (such as 'in' and 'on') are so common they arguably deserve no credit when applied to the wrong objects. For example, the relation tuple (girl, on top of, horse) should receive no credit if the correct relation tuple is (girl, on top of, court)⁴.

For the purpose of calculating the SPICE metric in Figure 4.6, we now define precision P , recall R as:

$$P(T_y, T_S) = \frac{|T_y \otimes T_S|}{|T_y|} \quad (4.8)$$

$$R(T_y, T_S) = \frac{|T_y \otimes T_S|}{|T_S|} \quad (4.9)$$

Re-evaluating our example using this definition, if we consider 'girl', 'woman' and 'lady' to be synonyms, and 'racket' and 'racquet' to be synonyms, $P(T_y, T_S) = \frac{5}{5}$ and $R(T_y, T_S) = \frac{5}{12}$ with unmatched tuples in T_S emphasised (above). This is a

⁴We naturally assume that horse and court are not synonyms

Runtime (s)	Bleu	METEOR	ROUGE-L	CIDEr	SPICE	SPICE (caching)
Min	10.1	32.9	7.7	31.4	1,101.3	40.3
Max	10.9	33.8	8.8	32.8	1,123.8	1,066.6
Average	10.5	33.3	8.1	31.9	1,110.5	149.9

Table 4.2: Runtime of SPICE in comparison to other commonly used image captioning metrics when evaluating 40.5K candidate captions on the COCO validation set. Results represent wall clock elapsed time on a desktop machine calculated using the outputs of 12 different image captioning models used in the 2015 COCO Captioning Challenge. With caching, SPICE runtime improves substantially. This is because after the first model evaluation, the validation captions (and a substantial number of the candidate captions) are found in the cache and do not need to be parsed again.

substantial improvement, but one problem remains. In the presence of synonym matching, many of the tuples in T_S are redundant as they are synonymous with other tuples in T_S . This means that recall may be understated, particularly when many reference captions are available such as in the COCO C40 test set.

We address this issue by merging synonymous object, attribute and relation references in the reference scene graph G_S . First, we merge object nodes with synonymous class labels, retaining both labels. Second, we merge synonymous attributes belonging to the same object node. Third, we merge synonymous relations linking the same object nodes. In general, this is a greedy process as there may be multiple merge opportunities at each step. Applying this approach to our example problem, now $R(T_y, T_S) = \frac{5}{9}$ based on the following merged set of reference tuples:

$$T_S = \{ (\text{girl/woman}), (\text{racquet}), (\mathbf{court}), (\text{girl/woman, young}), (\text{racquet, tennis}), (\mathbf{girl/woman, standing}), (\mathbf{court, tennis}), (\mathbf{girl/woman, on top of, court}), (\text{girl/woman, holding, racquet}) \}$$

Again, unmatched tuples are emphasised.

Being an F-score, SPICE is simple to understand and easily interpretable as it is naturally bounded between 0 and 1. Unlike CIDEr, SPICE does not use cross-dataset statistics—such as corpus word frequencies—and is therefore equally applicable to both small and large datasets. We provide an end-to-end example of the SPICE calculation in Figure 4.2, and further examples at the end of the chapter.

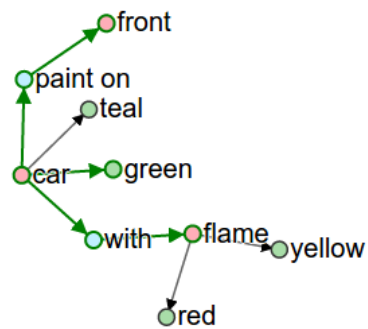
4.2.3 Running Time

We have released a Java implementation of the SPICE metric based on the Stanford CoreNLP software library [Manning et al., 2014]. In Table 4.2 we benchmark the running time of SPICE in comparison to other commonly used metrics for image



Candidate caption y :
A teal green car with yellow and red flames painted on the front.

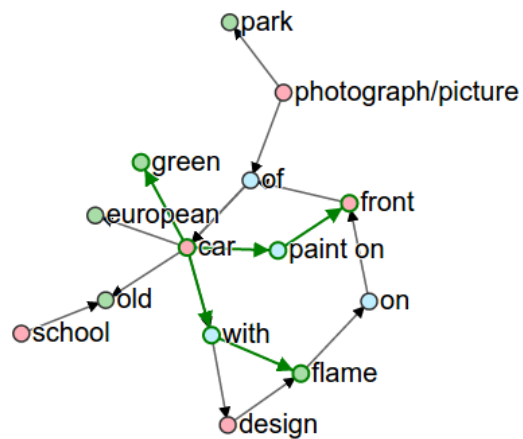
Candidate scene graph G_y :



Reference captions S :

- (1) An old green car with a flame design painted on the front of it.
- (2) A photograph of a european car.
- (3) An old school car with flames.
- (4) A picture of a car parked.
- (5) A car is painted with flames on the front.

Reference scene graph G_S :



SPICE score: 0.444 (Precision: 0.667,
Recall: 0.333)

Figure 4.2: Full example of the SPICE calculation. Candidate caption y and reference captions S are parsed to scene graphs G_y and G_S , respectively. SPICE is calculated as an F-score over matching tuples in G_y and G_S , which are highlighted in green: (car), (car, green), (front), (flame), (car, with, flame), (car, paint on, front). Note that no credit is given for correctly mentioning the colour of the flames ('yellow and red') as this information is not present in the reference scene graph G_S .

captioning. The running time of SPICE is dominated by the cost of the underlying PCFG dependency parser [Klein and Manning, 2003], which is reflected in longer running times in comparison to n-gram metrics that do not parse captions. For this reason we also implement a simple caching mechanism for storing caption parses using an LMDB database. As illustrated in Table 4.2, this significantly speeds up repeated evaluations against the same set of reference captions since, after the first evaluation, the reference captions (and a substantial number of the candidate captions) are found in the cache.

4.2.4 Limitations

One of the limitations of SPICE is that the restriction to (positive) atomic propositions means that there are certain kinds of assertions in captions that we cannot evaluate. For example, we cannot directly evaluate quantificational claims (e.g., ‘Most of the balls are red’) or negated claims (e.g., ‘The dog isn’t on the sofa’). Although previous analysis by van Miltenburg et al. [2016] found that negations are rarely present in image captions (0.56% of captions in the Flickr 30K [Young et al., 2014] and 0.54% of captions in COCO [Lin et al., 2014b]), this is nonetheless an important direction for future work.

More generally, whenever the focus of research is reduced to a single benchmark number, there are risks of unintended side-effects [Torralba and Efros, 2011]. For example, algorithms optimised for performance against a certain metric may produce high scores, while losing sight of the human judgement that the metric was supposed to represent. SPICE measures how well caption generators recover objects, attributes and the relations between them. However the metric neglects fluency, implicitly assuming that captions are well-formed. If this assumption is untrue in a particular application, SPICE may assign high scores to captions that represent only objects, attributes and relations, while ignoring other important aspects of grammar and syntax. In this scenario a fluency metric, such as *surprisal* [Hale, 2001; Levy, 2008], could be included in the evaluation. However, by default we have not included any fluency adjustments as conceptually we favour simpler, more easily interpretable metrics. To model human judgement in a particular task as closely as possible, a carefully tuned ensemble of metrics including SPICE capturing various dimensions of correctness would most likely be the best. ‘SPIDeR’, a linear combination of SPICE and CIDEr, is one such ensemble that appears to work well in practice [Liu et al., 2017a], as does a combination of SPICE, METEOR and Word Mover’s Distance (WMD) [Kilickaya et al., 2017].

4.2.5 Alternatives to Scene Graph Representations

The task of transforming a sentence into its meaning representation has also received considerable attention within the computational linguistics community. As such, there are several possible alternatives to scene graphs that could be considered in the context of image caption evaluation. Recent work has proposed a common framework for semantic graphs called an abstract meaning representation (AMR) [Banarescu et al., 2012], for which a number of parsers [Flanigan et al., 2014; Werling et al., 2015; Wang et al., 2015] and the Smatch evaluation metric [Cai and Knight, 2013] have been developed. However, in initial experiments, we found that AMR representations using Smatch similarity performed poorly as image caption representations. This may be because AMR parsers are not yet reliable, or because the AMR representation, being based on PropBank framesets [Kingsbury and Palmer, 2002; Palmer et al., 2005], is verb-oriented. Verbs in image captions are frequently absent or uninformative, e.g. ‘a very tall building with a train **sitting** next to it’. Because we exploit the semantic structure of scene descriptions and give primacy to nouns, our approach is apparently better suited to evaluating image captions.

Within the context of automatic MT evaluation, a number of other papers have proposed the use of shallow-semantic information such as semantic role labels (SRLs) [Giménez and Màrquez, 2007]. For example, in the MEANT metric [Lo et al., 2012], SRLs are used to try to capture the basic event structure of sentences – ‘*who* did *what* to *whom*, *when*, *where* and *why*’ [Pradhan et al., 2004]. Using this approach, sentence similarity is calculated by first matching semantic frames across sentences by starting with the verbs at their head. However, as previously noted, verb-oriented approaches may not be well suited to evaluating image captions. Conceptually, the closest work to ours is probably the bag of aggregated semantic tuples (BAST) metric [Ellebracht et al., 2015] for image captions. However, this work required the collection of a purpose-built dataset in order to learn to identify semantic tuples, and the proposed metric was not evaluated against human judgements or existing metrics.

4.3 Experiments

In this section, we compare SPICE to existing caption evaluation metrics. We study both system-level and caption-level correlation between automated metrics and human judgements, but we focus on system-level correlation as identification of the best captioning models is our primary motivation. Data for the evaluation is drawn from four datasets containing captions and human judgements. As the data was collected in multiple previous studies, a large variety of captioning models are represented.

Depending on the dataset, human judgements may consist of either pairwise rankings or graded scores, as described further below.

Our choice of correlation coefficients is consistent with an emerging consensus from the WMT Metrics Shared Task [Machacek and Bojar, 2014; Stanojević et al., 2015] for scoring machine translation metrics. To evaluate system-level correlation, we use the Pearson correlation coefficient. Although Pearson’s ρ measures linear association, it is smoother than rank-based correlation coefficients when the number of data points is small and systems have scores that are very close together. For caption-level correlation, we evaluate using Kendall’s τ rank correlation coefficient, which evaluates the similarity of pairwise rankings. Where human judgements consist of graded scores rather than pairwise rankings, we generate pairwise rankings by comparing scores over all pairs in the dataset. In datasets containing multiple independent judgements over the same caption pairs, we also report inter-human correlation.

4.3.1 Human Judgements

We now describe the datasets of image captions and human judgements that were used as the basis for comparing caption evaluation metrics.

4.3.1.1 COCO

The COCO 2014 dataset [Chen et al., 2015] consists of 123K images. 5K randomly selected test images are annotated with 40 captions each (C40 data). All other images are annotated with 5 captions each (C5 data). For further details regarding the underlying dataset refer to Section 3.2. As part of the 2015 COCO Captioning Challenge, human judgements were collected using Amazon Mechanical Turk (AMT) for the 15 learned captioning models that were submitted as competition entries, as well as two additional entries representing human captions and randomly selected reference captions. All AMT workers were native English speakers located in the USA that had been white-listed from previous work. A total of 255,000 human judgements were collected, representing three independent answers to five different questions for each entry. The questions capture the dimensions of overall caption quality (M1 - M2), correctness (M3), detailedness (M4), and saliency (M5), as detailed in Table 4.3. For pairwise rankings (M1, M2 and M5), each entry was evaluated using the same subset of 1000 images from the C40 test set. Scores for each evaluation metric were obtained from the COCO organisers, who used our code to calculate SPICE. The SPICE methodology was fixed before evaluating on COCO. At no stage were we given access to the COCO test captions.

4.3.1.2 Flickr 8K.

The Flickr 8K dataset [Hodosh et al., 2013] contains 8K images annotated with five human-generated reference captions each. The images were manually selected to focus mainly on people and animals performing actions. The dataset also contains graded human quality scores for 5K captions, with scores ranging from 1 (‘the selected caption is unrelated to the image’) to 4 (‘the selected caption describes the image without any errors’). Each caption was scored by three expert human evaluators sourced from a pool of native speakers. All evaluated captions were sourced from the dataset, but association to images was performed using an image retrieval system. In our evaluation we exclude 158 correct image-caption pairs where the candidate caption appears in the reference set. This reduces all correlation scores but does not disproportionately impact any metric.

4.3.1.3 Composite Dataset.

We refer to an additional dataset of 12K human judgements over Flickr 8K, Flickr 30K [Young et al., 2014] and COCO captions as the composite dataset [Aditya et al., 2015]. In this dataset, captions were scored using AMT on a graded correctness scale from 1 (‘The description has no relevance to the image’) to 5 (‘The description relates perfectly to the image’). Candidate captions were sourced from the human reference captions and two recent captioning models [Karpathy and Fei-Fei, 2015; Aditya et al., 2015].

4.3.1.4 PASCAL-50S

To create the PASCAL-50S dataset [Vedantam et al., 2015], 1K images from the UIUC PASCAL Sentence Dataset [Rashtchian et al., 2010]—originally containing five captions per image—were annotated with 50 captions each using AMT. The selected images represent 20 classes including people, animals, vehicles and household objects. The dataset also includes human judgements over 4K candidate sentence pairs. However, unlike in previous studies, AMT workers were not asked to evaluate captions against images. Instead, they were asked to evaluate caption triples by identifying ‘Which of the sentences, B or C, is more similar to sentence A?’, where sentence A is a reference caption, and B and C are candidates. If reference captions vary in quality, this approach may inject more noise into the evaluation process, however the differences between this approach and the previous approaches to human evaluations have not been studied. For each candidate sentence pair (B,C) evaluations were collected against 48 of the 50 possible reference captions. Candidate sentence

	M1		M2		M3		M4		M5	
	ρ	p-value	ρ	p-value	ρ	p-value	ρ	p-value	ρ	p-value
Bleu-1	0.24	(0.37)	0.29	(0.27)	0.72	(0.00)	-0.54	(0.03)	0.44	(0.09)
Bleu-4	0.05	(0.86)	0.10	(0.70)	0.58	(0.02)	-0.63	(0.01)	0.30	(0.27)
ROUGE-L	0.15	(0.59)	0.20	(0.47)	0.65	(0.01)	-0.55	(0.03)	0.38	(0.14)
METEOR	0.53	(0.04)	0.57	(0.02)	0.86	(0.00)	-0.10	(0.71)	0.74	(0.00)
CIDEr	0.43	(0.10)	0.47	(0.07)	0.81	(0.00)	-0.21	(0.43)	0.65	(0.01)
SPICE-exact	0.84	(0.00)	0.86	(0.00)	0.90	(0.00)	0.39	(0.00)	0.95	(0.00)
SPICE	0.88	(0.00)	0.89	(0.00)	0.89	(0.00)	0.46	(0.07)	0.97	(0.00)

M1: Percentage of captions evaluated as better or equal to human caption.

M2: Percentage of captions that pass the Turing Test.

M3: Average correctness of the captions on a scale 1–5 (incorrect - correct).

M4: Average detail of the captions from 1–5 (lacking details - very detailed).

M5: Percentage of captions that are similar to human description.

Table 4.3: System-level Pearson’s ρ correlation between automatic evaluations and human judgements for the 15 competition entries plus human captions in the 2015 COCO Captioning Challenge [Chen et al., 2015]. SPICE more accurately reflects human judgement overall (M1–M2), and across each dimension of quality (M3–M5, representing correctness, detailedness and saliency)

pairs were generated from both human and model captions, paired in four ways: Human-Correct (two correct human captions), Human-Incorrect (two human captions where one is from a different image), Human-Model (a human caption and a model-generated caption), and Model-Model (two model-generated captions).

4.3.2 System-Level Correlation

In Table 4.3 we report system-level correlations between metrics and human judgements over entries in the 2015 COCO Captioning Challenge [Chen et al., 2015]. Each entry is evaluated using the same 1000 image subset of the COCO C40 test set. SPICE significantly outperforms existing metrics, reaching a correlation coefficient of 0.88 with human quality judgements (M1), compared to 0.43 for CIDEr and 0.53 for METEOR. As illustrated in Table 4.3, SPICE more accurately reflects human judgement overall (M1 - M2), and across each dimension of quality (M3 - M5, representing correctness, detailedness and saliency). Importantly, only SPICE rewards captions that are more detailed (indicated by positive correlation with M4). Bleu and ROUGE-L appear to penalise detailedness, while the results for CIDEr and METEOR are not statistically significant.

As illustrated in Figure 4.3, SPICE is the only metric to correctly rank human-generated captions first. CIDEr and METEOR rank human captions 7th and 4th,

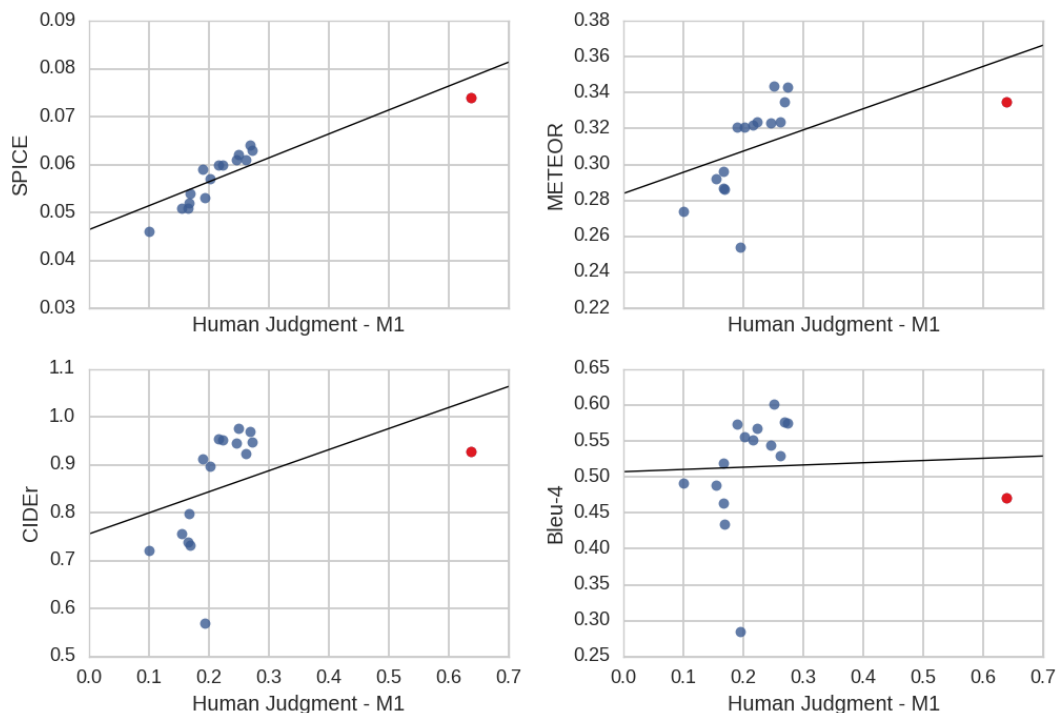


Figure 4.3: Automatic evaluation scores vs. human judgements for the 15 entries in the 2015 COCO Captioning Challenge. Each blue data point represents a single model. Human-generated captions are marked in red. Only SPICE scores human-generated captions significantly higher than challenge entries, which is consistent with human judgement.

respectively. SPICE is also the only metric to correctly select the top-5 non-human entries. To help understand the importance of synonym-matching when calculating SPICE scores, we also evaluated SPICE using exact tuple matching in the F-score calculation. In this case, $u \sim v$ in Equation 4.7 is redefined to mean that tuples u and v have the same length and all corresponding elements *have the same lemma*. These results are reported as SPICE-exact in Table 4.3. Performance degraded only marginally, although we expect synonym-matching to become more important when fewer reference captions are available.

4.3.3 Colour Perception, Counting and Other Questions

While system-level scores are very important, we would also like to know the answers to questions such as ‘which caption-generator best understands colours?’ and ‘can caption generators count?’. Existing n-gram evaluation metrics have little to offer in terms of understanding the relative strengths and weaknesses, or error modes,

	M1	SPICE	Object	Relation	Attribute	Color	Count	Size
Human	0.638	0.074	0.190	0.023	0.054	0.055	0.095	0.026
MSR	0.268	0.064	0.176	0.018	0.039	0.063	0.033	0.019
Google	0.273	0.063	0.173	0.018	0.039	0.060	0.005	0.009
MSR Captivator	0.250	0.062	0.174	0.019	0.032	0.054	0.008	0.009
Berkeley LRCN	0.246	0.061	0.170	0.023	0.026	0.030	0.015	0.010
Montreal/Toronto	0.262	0.061	0.171	0.023	0.026	0.023	0.002	0.010
m-RNN	0.223	0.060	0.170	0.021	0.026	0.038	0.007	0.004
N. Neighbour	0.216	0.060	0.168	0.022	0.026	0.027	0.014	0.013
m-RNN (Baidu)	0.190	0.059	0.170	0.022	0.022	0.031	0.002	0.005
PicSOM	0.202	0.057	0.162	0.018	0.027	0.025	0.000	0.012
MIL	0.168	0.054	0.157	0.017	0.023	0.036	0.007	0.009
Brno	0.194	0.053	0.144	0.012	0.036	0.055	0.029	0.025
MLBL	0.167	0.052	0.152	0.017	0.021	0.015	0.000	0.004
NeuralTalk	0.166	0.051	0.153	0.018	0.016	0.013	0.000	0.007
ACVT	0.154	0.051	0.152	0.015	0.021	0.019	0.001	0.008
Tsinghua Bigeye	0.100	0.046	0.138	0.013	0.017	0.017	0.000	0.009
Random	0.007	0.008	0.029	0.000	0.000	0.000	0.004	0.000

Table 4.4: M1 (percentage of captions that are evaluated as better or equal to human caption) vs. SPICE scores by semantic proposition subcategory. Although the MSR model outperforms the human baseline in terms of identifying object colour attributes, none of the model-based entries exhibits a convincing ability to count (being well below human performance).

of various models. However, SPICE has the useful property that it is defined over tuples that are easy to subdivide into meaningful categories. For example, SPICE F-scores can be quantified separately for object, attribute and relation tuples by re-defining Equation 4.2, the mapping from scene graphs to tuples, using Equation 4.3, Equation 4.4, or Equation 4.5, respectively. In general, scores can be analysed to any arbitrary level by filtering the tuples generated by Equation 4.2.

To demonstrate this capability, in Table 4.4 we review the performance of 2015 COCO Captioning Challenge submissions in terms of overall SPICE score, detection of objects, relations and attributes, as well as *colour perception*, *counting ability*, and understanding of *size attributes*. Here, colour, counting and size performance is quantified by using word lists to identify subsets of attribute tuples that contain colours, the numbers from one to ten, and size-related adjectives, respectively. This affords us some insight, for example, into whether caption generators actually understand colour, and how good they are at counting. As shown in Table 4.4, the MSR entry [Fang et al., 2015]—incorporating specifically trained visual detectors for nouns, verbs and adjectives—exceeds the human F-score baseline for tuples contain-

	Flickr 8K [Hodosh et al., 2013]	Composite [Aditya et al., 2015]
Bleu-1	0.32	0.26
Bleu-4	0.14	0.18
ROUGE-L	0.32	0.28
METEOR	0.42	0.35
CIDEr	0.44	0.36
SPICE	0.45	0.39
Inter-human	0.73	-

Table 4.5: Caption-level Kendall’s τ correlation between automatic evaluations and human quality scores. At the caption-level SPICE modestly outperforms existing metrics. All p-values (not shown) are less than 0.001.

ing colour attributes. However, there is little evidence that any of these models have learned to count objects, as these scores are far below human level performance.

4.3.4 Caption-Level Correlation

In Table 4.5 we report caption-level correlations between automated metrics and human judgements on Flickr 8K [Hodosh et al., 2013] and the composite dataset [Aditya et al., 2015]. At the caption level, SPICE achieves a rank correlation coefficient of 0.45 with Flickr 8K human scores, compared to 0.44 for CIDEr and 0.42 for METEOR. Relative to the correlation between human scores of 0.73, this represents only a modest improvement over existing metrics. However, as reported in Section 4.3.2, SPICE more closely correlates with human judgement at the system level, which is more important for comparing and evaluating models. Results are similar on the composite dataset, with SPICE achieving a rank correlation coefficient of 0.39, compared to 0.36 for CIDEr and 0.35 for METEOR. As this dataset only includes one score per image-caption pair, inter-human agreement cannot be established.

4.3.5 Pairwise Accuracy

For consistency with previous evaluations on the PASCAL-50S dataset [Vedantam et al., 2015], instead of reporting rank correlations we evaluate on this dataset using accuracy. A metric is considered accurate if it gives an equal or higher score to the caption in each candidate pair most commonly preferred by human evaluators. To help quantify the impact of reference captions on performance, the number of reference captions available to the metrics is varied from 1 to 48. This approach follows the original work on this dataset [Vedantam et al., 2015], although our results differ

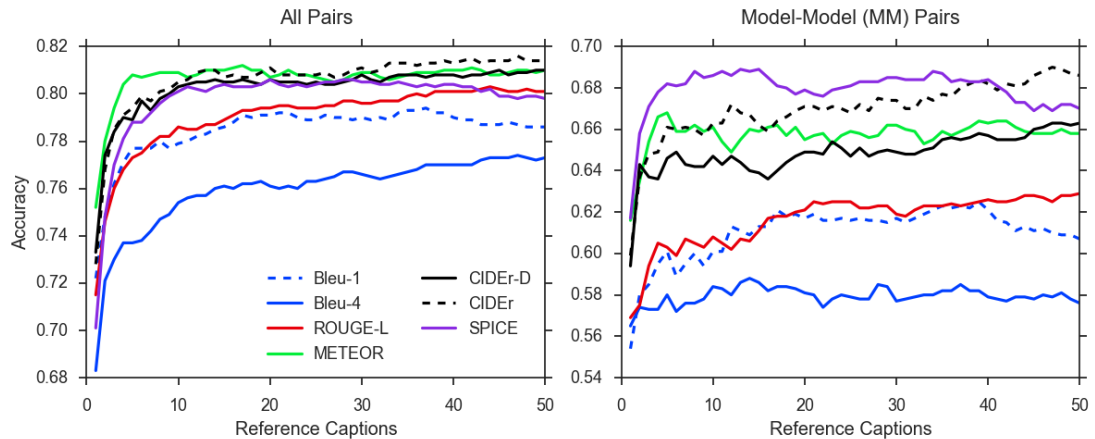


Figure 4.4: Pairwise classification accuracy of automated metrics at matching human judgement on PASCAL-50S with 1-50 reference captions

	Human-Correct	Human-Incorrect	Human-Model	Model-Model	All
Bleu-1	64.9	95.2	90.7	60.1	77.7
Bleu-2	56.6	93.0	87.2	58.0	73.7
ROUGE-L	61.7	95.3	91.7	60.3	77.3
METEOR	64.0	98.1	94.2	66.8	80.8
CIDEr	61.9	98.0	91.0	64.6	78.9
SPICE	63.3	96.3	87.5	68.2	78.8

Table 4.6: Caption-level classification accuracy of evaluation metrics at matching human judgement on PASCAL-50S with 5 reference captions. SPICE is best at matching human judgements on pairs of model-generated captions (Model-Model). METEOR is best at differentiating human and model captions (Human-Model) and human captions where one is incorrect (Human-Incorrect). Bleu-1 performs best given two correct human captions (Human-Correct)

slightly which may be due to randomness in the choice of reference caption subsets, or differences in metric implementations (we use the COCO evaluation code).

On PASCAL-50S, there is little difference in overall performance between SPICE, METEOR and CIDEr, as shown in Figure 4.4 left. However, of the four kinds of captions pairs, SPICE performs best in terms of distinguishing between two model-generated captions (Model-Model pairs) as illustrated in Table 4.6 and Figure 4.4 right. This is important as distinguishing better performing algorithms is the primary motivation for this work.

4.4 Chapter Summary

The interaction between vision and language is a promising area for research that is essential to unlocking numerous practical applications in robotics and artificial intelligence (AI). However, with the increasing focus on combining visual and linguistic learning comes an increasing need for methods that can automatically evaluate the language-based outputs of these models. In this chapter, we present SPICE, a novel semantic evaluation metric that measures how effectively image captions recover objects, attributes and the relations between them.

Using a range of datasets and human evaluations, we show that SPICE outperforms existing methods for evaluating captions in terms of agreement with human evaluations. Most importantly, on COCO—the largest image caption dataset—SPICE is the most effective metric for automatically ranking image captioning models. We further demonstrate that SPICE performance can be decomposed to answer questions such as ‘which caption-generator best understands colours?’ and ‘can caption generators count?’, allowing the performance of any model can be analysed in greater detail than with other automated metrics.

Since its release, the SPICE metric has been readily adopted by the image captioning community, including on the official COCO test server. A year and a half since its first publication, SPICE is still the only automatic evaluation metric that (correctly) judges human image descriptions to be superior to those from all proposed image captioning models⁵. We are aware that significant challenges still remain in semantic parsing, but unlike n-gram metrics, SPICE offers scope for further improvements to the extent that semantic parsing techniques continue to improve.

Conceptually, SPICE differs from existing textual similarity metrics in two key respects. First, SPICE uses a common meaning representation—the scene graph—to establish connections between the visual and linguistic modalities. Second, the SPICE metric incorporates learning, notably within the dependency parser that constitutes the first stage of the scene graph parser. However, in the difficult yet important task of evaluating visually-grounded language there remains much to be explored. Recent work from Wang et al. [2018] focuses on the problem of parsing text to scene graphs, proposing further improvements to the SPICE parser. Yet, the reason why abstract meaning representations (AMRs) [Banarescu et al., 2012] perform poorly as image caption representations in initial experiments is not well understood. Given the large ongoing effort to establish AMR corpora and parsers, in future work we would like to further investigate the use of AMRs for image caption evaluation. Inspired by the

⁵Based on COCO C40 test leaderboard as at March 2018 (refer <http://cocodataset.org/#captions-leaderboard>).

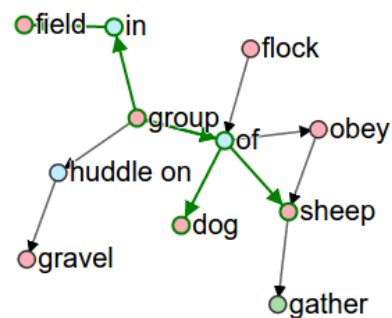
adversarial training methods used by Generative Adversarial Networks [Goodfellow et al., 2014], we also hope to see more research into learned caption evaluation metrics [Cui et al., 2018]. We provide further examples of the SPICE calculation in Figures 4.5 and 4.6.

Image I:

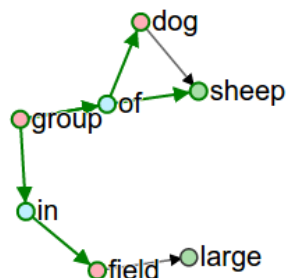


Reference captions S:

- (1) A group of sheep is in a field.
- (2) Sheep gathered together.
- (3) A group of sheep and a dog.
- (4) A flock of sheep obey the dog.
- (5) A group of sheep are huddled together on the gravel.

Reference scene graph G_S :

Candidate caption y :
A group of sheep and a sheep dog in a large field.

Candidate scene graph G_y :

SPICE score: 0.609 (Precision: 0.778,
Recall: 0.5)

Figure 4.5: Additional example of the SPICE calculation. This candidate caption scores very highly, particularly on precision, as 7 out of 9 tuples in the candidate scene graph are also found in the reference scene graph. The only missing tuples are (dog, sheep) and (field, large), since the reference captions do not mention what type of dog it is or the size of the field.

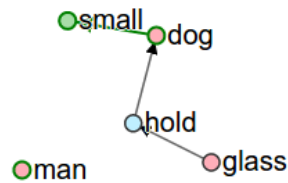
Image I :



Candidate caption y :

A man in glasses holding a small dog.

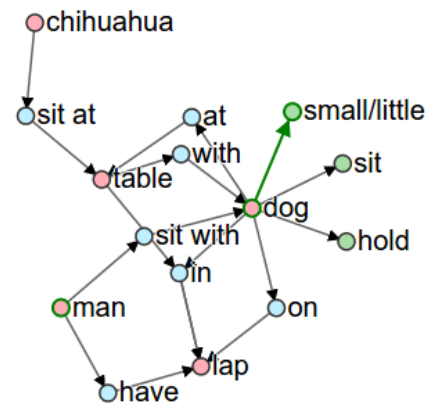
Candidate scene graph G_y :



Reference captions S :

- (1) A man holding small dog sitting at a table.
- (2) A man sits with a little dog on his lap.
- (3) The dog is in the man's lap.
- (4) A man sitting at the table with a small dog on his lap.
- (5) A chihuahua is sitting at a table in a man's lap.

Reference scene graph G_S :



SPICE score: 0.286 (Precision: 0.6, Recall: 0.188)

Figure 4.6: Additional example of the SPICE calculation. In this case, several parsing errors can be seen, for example in the candidate caption ‘man in glasses holding a small dog’ is incorrectly parsed to (glass, hold, dog) rather than (man, hold, dog) and (man, in, glasses). However, the SPICE metric still rewards the candidate caption for identifying (man), (dog), and (dog, small).

Bottom-Up and Top-Down Visual Attention for Image Captioning and VQA

Chapter 4 established SPICE as the most effective metric for automatic image caption evaluation. Armed with this metric and others, we now directly address the challenges of image captioning [Chen et al., 2015] and visual question answering (VQA) [Goyal et al., 2017]. In both these tasks it is often necessary to perform some fine-grained visual processing, or even multiple steps of reasoning to generate high quality outputs. As a result, visual attention mechanisms have been widely adopted in both image captioning [Rennie et al., 2017; Lu et al., 2017; Yang et al., 2016b; Xu et al., 2015] and VQA [Fukui et al., 2016; Lu et al., 2016; Xu and Saenko, 2016; Yang et al., 2016a; Zhu et al., 2016]. These mechanisms improve performance by learning to focus on the regions of the image that are salient and are currently based on deep neural network architectures. In this chapter, we propose a combined bottom-up and top-down attention mechanism that enables attention to be calculated at the level of objects and other salient image regions. We argue that this is the natural basis for attention to be considered. Applying this approach to image captioning and VQA, we achieve state of the art results in both tasks, while simultaneously improving the interpretability of the resulting models.

5.1 Attention Networks

An attention network or attention mechanism is a computational unit in a neural network that restricts a large set of input representations by selectively focusing on the most salient or relevant elements of the set. More formally, an attention mech-

anism can be described as a mapping from a query q and a set of key-value pairs¹ $V = \{(k_1, v_1), \dots, (k_k, v_k)\}$ to an attended output \hat{v} , where the query, keys, values, and output are all vectors.

Attention mechanisms are typically used as components with a larger network, with q and V parametrised by neural network outputs and \hat{v} fed to a downstream network. Their usefulness arises when the query q encodes some task context, and the attended output \hat{v} captures the most salient aspects of the input V given the task at hand. Attention can alleviate the information bottleneck of compressing a large set of input representations into a single fixed-length vector, allowing for variably-sized inputs. For example, in the task of visual question answering (VQA), the task context² q can be provided by a representation of the question, with V containing visual representations from different parts of the image. As described further in Section 5.2, under the influence of the downstream loss function an attention mechanism can learn to focus on the most salient areas of the image when answering questions. Attention mechanisms have been applied to both textual [Bahdanau et al., 2015b; Luong et al., 2014] and visual inputs [Xu et al., 2015; Zhu et al., 2016], in which the elements of V are typically represented by word vectors or spatial locations in a CNN feature map, respectively.

To implement the attention mechanism, various approaches have been considered. So-called ‘hard’ attention mechanisms typically determine \hat{v} by selecting a single value from V [Xu et al., 2015]. However, in this thesis we focus on ‘soft’ attention mechanisms. To calculate the attended output vector \hat{v} , these approaches generate an attention score $a_i \in \mathbb{R}$ for each element of V . The attended feature vector \hat{v} is calculated as a weighted sum over all the values in V after normalizing the attention scores using a softmax:

$$\alpha = \text{softmax}(\mathbf{a}) \quad (5.1)$$

$$\hat{v} = \sum_{i=1}^k \alpha_i v_i \quad (5.2)$$

Unlike hard attention networks, soft attention networks are typically fully differentiable and can be trained using standard backpropagation methods [Xu et al., 2015].

Given set V and task context q , there is no single standard approach to modelling the attention score vector \mathbf{a} . Below, we list three simple alternatives proposed by

¹It is also common to make no distinction between keys and values in V , in which case the input can be specified as $V = \{v_1, \dots, v_k\}$.

²We characterize the query q as the task context. In some other works, the output \hat{v} is referred to as a context vector. The language used to describe attention networks has not been standardized.

Luong et al. [2014]:

$$\textit{Dot} \quad a_i = \mathbf{k}_i^T \mathbf{q} \quad (5.3)$$

$$\textit{General} \quad a_i = \mathbf{k}_i^T W_a \mathbf{q} \quad (5.4)$$

$$\textit{Concat} \quad a_i = \mathbf{w}_a^T \tanh(W_a[\mathbf{k}_i, \mathbf{q}]) \quad (5.5)$$

where W_a and \mathbf{w}_a are learned parameters. In this chapter our attention mechanisms are based on Equation 5.5. However, many more sophisticated approaches have also been considered. For example, in stacked attention [Yang et al., 2016a], multiple attended outputs can be calculated by refining subsequent queries using the attended outputs from previous queries. In contrast, multi-headed attention consists of several attention layers with different parameters running in parallel [Vaswani et al., 2017]. While these works focus on the determination of the attention weights α , in this work we focus on the determination of the attention candidates in V , which has received much less scrutiny.

5.1.1 Bottom-Up vs. Top-Down Attention

In the human visual system, attention can be focused volitionally by top-down signals determined by the current task (e.g., looking for something), and automatically by bottom-up signals associated with unexpected, novel or salient stimuli [Buschman and Miller, 2007; Corbetta and Shulman, 2002]. In this chapter we adopt similar terminology and refer to attention mechanisms driven by contextual information from outside the current image as ‘top-down’, and purely image-driven attention mechanisms as ‘bottom-up’. This definition is consistent with early work in computer vision that described top-down models as ‘goal-driven’, and bottom-up models as ‘image-driven’ [Navalpakkam and Itti, 2006], although to the best of our knowledge the terms ‘bottom-up’ and ‘top-down’ have never been rigorously defined in the computer vision literature.

Under our definition, most conventional visual attention mechanisms used in image captioning and VQA are of the top-down variety. Contextual information is provided by a representation of a partially-completed caption in the case of image captioning [Rennie et al., 2017; Lu et al., 2017; Yang et al., 2016b; Xu et al., 2015], or a representation of the question in the case of VQA [Fukui et al., 2016; Lu et al., 2016; Xu and Saenko, 2016; Yang et al., 2016a; Zhu et al., 2016]. In each case attention is applied to the output of one or more layers of a convolutional neural net (CNN), by predicting a weighting for every spatial location in the (possibly interpolated) CNN output. In these models, the number of candidate image regions is a hyperparame-

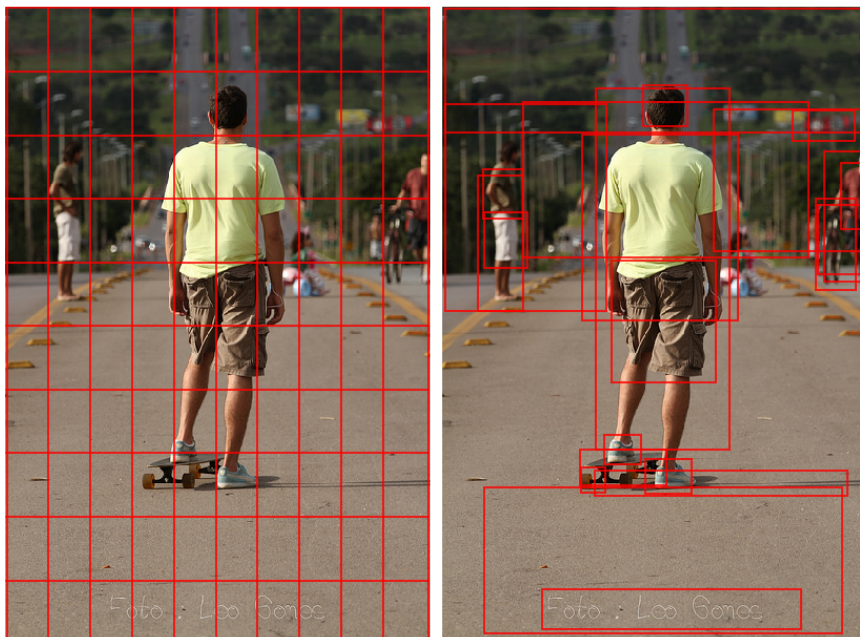


Figure 5.1: Typically, attention models operate on CNN features corresponding to a uniform grid of equally-sized image regions (left). Our approach enables attention to be calculated at the level of objects and other salient image regions (right).

ter. However, determining the optimal number of image regions invariably requires an unwinnable trade-off between coarse and fine levels of detail. Furthermore, the arbitrary positioning of the regions with respect to image content may make it more difficult to detect objects that are poorly aligned to regions and to bind visual concepts associated with the same object. As illustrated conceptually in Figure 5.1, the resulting input regions correspond to a uniform grid of equally sized and shaped neural receptive fields – irrespective of the content of the image. To generate more human-like captions and question answers, objects and other salient image regions are a much more natural basis for attention [Egley et al., 1994; Scholl, 2001].

In this chapter we propose a combined bottom-up and top-down visual attention mechanism. The bottom-up (image-driven) mechanism proposes a set of *salient* image regions, with each region represented by a pooled convolutional feature vector. Practically, we implement bottom-up attention using Faster R-CNN object detector [Ren et al., 2015a], which represents a natural expression of a bottom-up attention mechanism. The top-down (goal-driven) mechanism uses task-specific context to predict an attention distribution over the image regions. The attended feature vector is then computed as a weighted average of image features over all regions.

Comparatively few previous works have considered applying attention to salient image regions. We are aware of two papers. Jin et al. [2015] use selective search [Uijlings et al., 2013] to identify salient image regions, which are filtered with a classifier then resized and CNN-encoded as input to an image captioning model with attention. The Areas of Attention captioning model [Pedersoli et al., 2017] uses either edge boxes [Zitnick and Dollár, 2014] or spatial transformer networks [Jaderberg et al., 2015] to generate image features, which are processed using an attention model based on three bi-linear pairwise interactions [Pedersoli et al., 2017]. In our work, rather than using hand-crafted or differentiable region proposals [Uijlings et al., 2013; Zitnick and Dollár, 2014; Jaderberg et al., 2015], we leverage Faster R-CNN [Ren et al., 2015a], establishing a closer link between vision and language tasks and recent progress in object detection. With this approach we are able to pretrain our region proposals on object detection datasets. Conceptually, the advantages should be similar to pretraining visual representations on ImageNet [Russakovsky et al., 2015] and leveraging significantly more cross-domain knowledge. We additionally apply our method to VQA, establishing the broad applicability of our approach.

5.2 Approach

Given an image I , both our image captioning model and our VQA model take as input a possibly variably-sized set of k image features, $V = \{v_1, \dots, v_k\}$, $v_i \in \mathbb{R}^D$, such that each image feature encodes a salient region of the image. The spatial image features V can be variously defined as the output of our bottom-up attention model, or, following standard practice, as the spatial output layer of a CNN. We describe our approach to implementing a bottom-up attention model in Section 5.2.1. In Section 5.2.2 we outline the architecture of our image captioning model and in Section 5.2.3 we outline the VQA model. We note that for the top-down attention component, both models use simple one-pass attention mechanisms, as opposed to the more complex schemes of recent models such as stacked, multi-headed, or bidirectional attention [Yang et al., 2016a; Jabri et al., 2016; Kazemi and Elqursh, 2017; Lu et al., 2016] that could also be applied.

5.2.1 Bottom-Up Attention Model

The definition of spatial image features V is generic. However, in this work we define spatial regions in terms of bounding boxes and implement bottom-up attention using Faster R-CNN [Ren et al., 2015a]. Faster R-CNN is an object detection model designed to identify instances of objects belonging to certain classes and localise them

with bounding boxes. Other region proposal networks could also be trained as an attentive mechanism [Redmon et al., 2016; Liu et al., 2016].

Faster R-CNN detects objects in two stages. The first stage, described as a Region Proposal Network (RPN), predicts object proposals. A small network is slid over features at an intermediate level of a CNN. At each spatial location the network predicts a class-agnostic objectness score and a bounding box refinement for anchor boxes of multiple scales and aspect ratios. Using greedy non-maximum suppression with an intersection-over-union (IoU) threshold, the top box proposals are selected as input to the second stage. In the second stage, region of interest (RoI) pooling is used to extract a small feature map (e.g. 14×14) for each box proposal. These feature maps are then batched together as input to the final layers of the CNN. The final output of the model consists of a softmax distribution over class labels and class-specific bounding box refinements for each box proposal.

In this work, we use Faster R-CNN in conjunction with the ResNet-101 [He et al., 2016a] CNN. To generate an output set of image features V for use in image captioning or VQA, we take the final output of the model and perform non-maximum suppression for each object class using an IoU threshold. We then select all regions where any class detection probability exceeds a confidence threshold. For each selected region i , v_i is defined as the mean-pooled convolutional feature from this region, such that the dimension D of the image feature vectors is 2048. Used in this fashion, Faster R-CNN effectively functions as a ‘hard’ attention mechanism, as only a relatively small number of image bounding box features are selected from a large number of possible configurations.

To pretrain the bottom-up attention model, we first initialise Faster R-CNN with ResNet-101 pretrained for classification on ImageNet [Russakovsky et al., 2015]. We then train on Visual Genome [Krishna et al., 2016] data. To aid the learning of good feature representations, we add an additional training output for predicting attribute classes such as ‘blue’, ‘muscular’, ‘cooked’ or ‘pointed’ (in addition to object classes such as ‘car’, ‘man’, ‘chicken’ or ‘beard’). To predict attributes for region i , we concatenate the mean pooled convolutional feature v_i with a learned embedding of the ground-truth object class, and feed this into an additional output layer defining a softmax distribution over each attribute class plus a ‘no attributes’ class.

The original Faster R-CNN multi-task loss function contains four components, defined over the classification and bounding box regression outputs for both the RPN and the final object class proposals respectively. We retain these components and add an additional multi-class loss component to train the attribute predictor. In Figure 5.2 we provide some examples of model output. For further implementation details refer to Section 5.2.4.

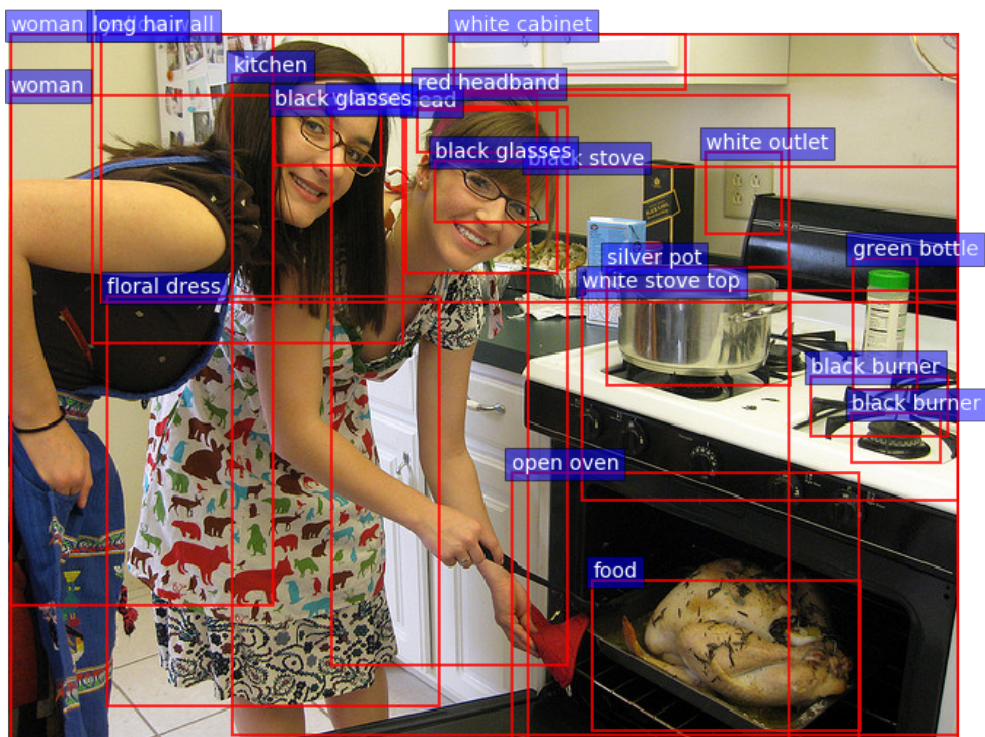
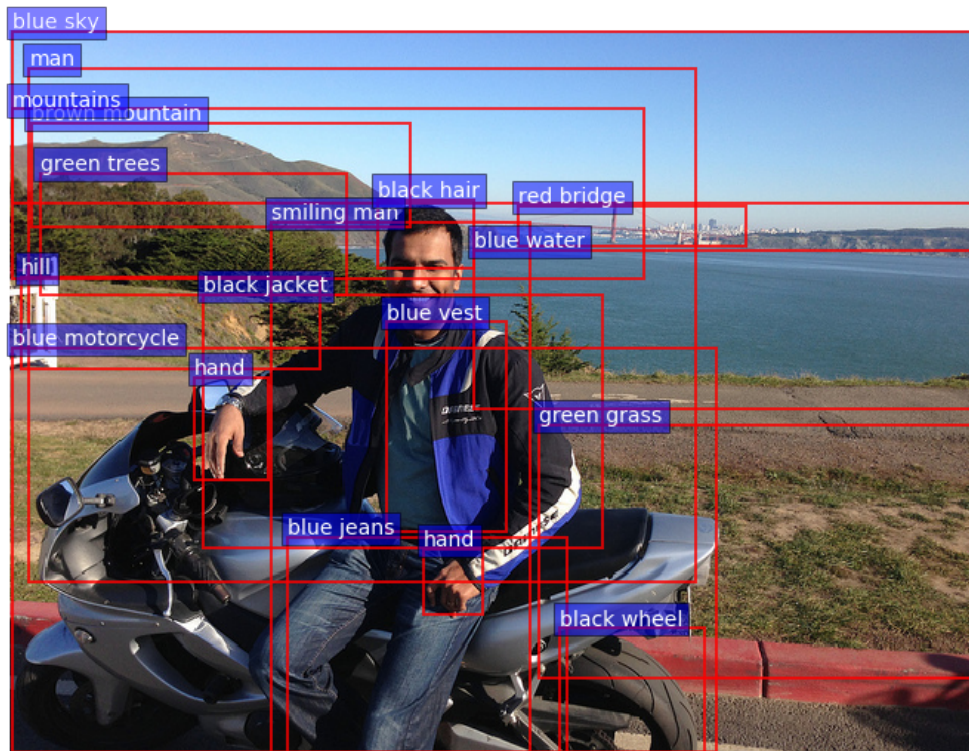


Figure 5.2: Example outputs from our Faster R-CNN bottom-up attention model. Each bounding box is labelled with an attribute class followed by an object class. Note however, that in captioning and VQA we utilize only the feature vectors – not the predicted labels.

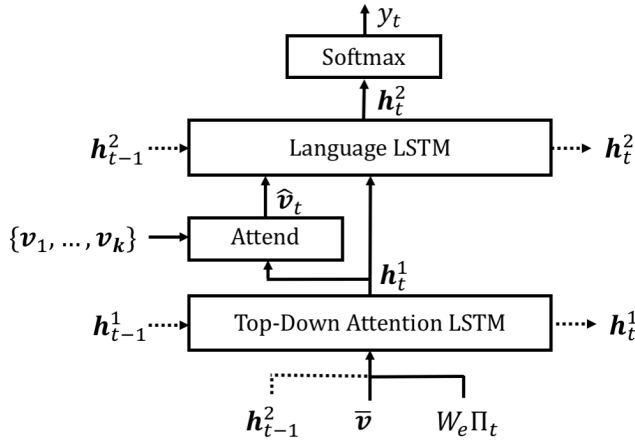


Figure 5.3: Overview of the proposed captioning model. Two LSTM layers are used to selectively attend to spatial image features $\{v_1, \dots, v_k\}$. These features can be defined as the spatial output of a CNN, or following our approach, generated using bottom-up attention.

5.2.2 Captioning Model

Given a set of image features V , our proposed captioning model uses a ‘soft’ top-down attention mechanism to weight each feature during caption generation, using the existing partial output sequence as context. This approach is broadly similar to several previous works [Rennie et al., 2017; Lu et al., 2017; Xu et al., 2015]. However, the particular design choices outlined below make for a relatively simple yet high-performing baseline model. Even without bottom-up attention, our captioning model achieves performance comparable to state of the art on most evaluation metrics (refer Table 5.2).

At a high level, the captioning model is composed of two Long Short-Term Memory (LSTM) layers [Hochreiter and Schmidhuber, 1997]. In the sections that follow we will refer to the operation of the LSTM over a single time step using the following notation:

$$h_t = \text{LSTM}(x_t, h_{t-1}) \quad (5.6)$$

where x_t is the LSTM input vector and h_t is the LSTM output vector. Refer to Section 2.4.1 for further background. We now describe the formulation of the LSTM input vector x_t and the output vector h_t for each layer of the model. The overall captioning model is illustrated in Figure 5.3.

5.2.2.1 Top-Down Attention LSTM

Within the captioning model, we characterise the first LSTM layer as a top-down visual attention model, and the second LSTM layer as a language model, indicating each layer with superscripts in the equations that follow. Note that the bottom-up attention model is described in Section 5.2.1, and in this section its outputs are simply considered as features V . The input vector to the attention LSTM at each time step consists of the previous output of the language LSTM, concatenated with the mean-pooled image feature $\bar{v} = \frac{1}{k} \sum_i v_i$ and an encoding of the previously generated word, given by:

$$\mathbf{x}_t^1 = [\mathbf{h}_{t-1}^2, \bar{v}, W_e \Pi_t] \quad (5.7)$$

where $W_e \in \mathbb{R}^{E \times |\Sigma|}$ is a word embedding matrix for a vocabulary Σ , and Π_t is one-hot encoding of the input word at timestep t . These inputs provide the attention LSTM with maximum context regarding the state of the language LSTM, the overall content of the image, and the partial caption output generated so far, respectively. The word embedding is learned from random initialisation without pretraining.

Given the output \mathbf{h}_t^1 of the attention LSTM, at each time step t we generate a normalised attention weight $\alpha_{i,t}$ for each of the k image features v_i as follows:

$$a_{i,t} = \mathbf{w}_a^T \tanh(W_{va} v_i + W_{ha} \mathbf{h}_t^1) \quad (5.8)$$

$$\boldsymbol{\alpha}_t = \text{softmax}(\mathbf{a}_t) \quad (5.9)$$

where $W_{va} \in \mathbb{R}^{H \times V}$, $W_{ha} \in \mathbb{R}^{H \times M}$ and $\mathbf{w}_a \in \mathbb{R}^H$ are learned parameters. The attended image feature used as input to the language LSTM is calculated as a convex combination of all input features:

$$\hat{v}_t = \sum_{i=1}^k \alpha_{i,t} v_i \quad (5.10)$$

5.2.2.2 Language LSTM

The input to the language model LSTM consists of the attended image feature, concatenated with the output of the attention LSTM, given by:

$$\mathbf{x}_t^2 = [\hat{v}_t, \mathbf{h}_t^1] \quad (5.11)$$

Let $\mathbf{y} = (y_1, \dots, y_T)$ denote an output sequence of length T containing words or other tokens from vocabulary Σ . At each time step t the conditional distribution over

possible output words is given by:

$$p(y_t | y_{1:t-1}) = \text{softmax}(W_p \mathbf{h}_t^2 + \mathbf{b}_p) \quad (5.12)$$

where $W_p \in \mathbb{R}^{|\Sigma| \times M}$ and $\mathbf{b}_p \in \mathbb{R}^{|\Sigma|}$ are learned weights and biases. The distribution over complete output sequences is calculated as the product of conditional distributions:

$$p(\mathbf{y}) = \prod_{t=1}^T p(y_t | y_{1:t-1}) \quad (5.13)$$

5.2.2.3 Objective

Given a target ground truth sequence \mathbf{y}^* and a captioning model with parameters θ , we minimise the following cross entropy loss (refer Section 2.4.3):

$$L_{XE}(\theta) = - \sum_{t=1}^T \log(p_\theta(y_t^* | y_{1:t-1}^*)) \quad (5.14)$$

For fair comparison with recent work [Rennie et al., 2017] we also report results optimised for CIDEr [Vedantam et al., 2015]. Initialising from the cross-entropy trained model, we seek to minimise the negative expected score:

$$L_R(\theta) = -\mathbf{E}_{\mathbf{y} \sim p_\theta}[r(\mathbf{y})] \quad (5.15)$$

where r is the score function (e.g., CIDEr). Following the approach described as Self-Critical Sequence Training [Rennie et al., 2017] (SCST), the gradient of this loss can be approximated:

$$\nabla_\theta L_R(\theta) \approx -(r(\mathbf{y}^s) - r(\hat{\mathbf{y}})) \nabla_\theta \log p_\theta(\mathbf{y}^s) \quad (5.16)$$

where \mathbf{y}^s is a sampled caption and $r(\hat{\mathbf{y}})$ defines the baseline score obtained by greedily decoding the current model. SCST (like other REINFORCE [Williams, 1992] algorithms) explores the space of captions by sampling from the policy during training. This gradient tends to increase the probability of sampled captions that score higher than the score from the current model.

In our experiments, we follow SCST but we speed up the training process by restricting the sampling distribution. Using beam search decoding (refer Section 2.4.3), we sample only from those captions in the decoded beam. Empirically, we have observed when decoding using beam search that the resulting beam typically contains at least one very high scoring caption – although frequently this caption

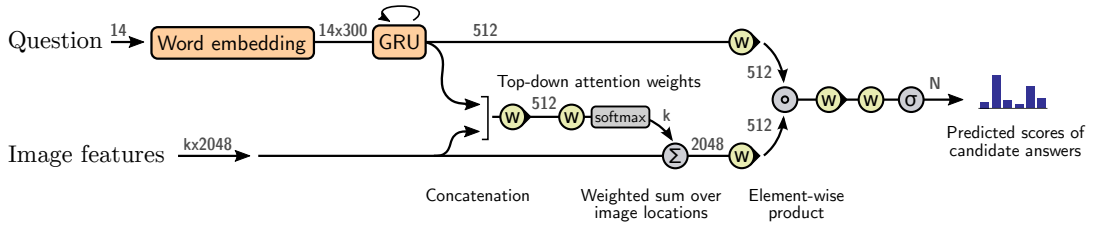


Figure 5.4: Overview of the VQA model used in experiments. A deep neural network implements a joint embedding of the question and image features $\{v_1, \dots, v_k\}$. These features can be defined as the spatial output of a CNN, or following our approach, generated using bottom-up attention. Output is generated by a multi-label classifier operating over a fixed set of candidate answers. Gray numbers indicate the dimensions of the vector representations between layers. Yellow elements use learned parameters.

does not have the highest log-probability of the set. In contrast, we observe that very few unrestricted caption samples score higher than the greedily-decoded caption. Using this approach, we complete CIDEr optimisation in a single epoch.

5.2.3 VQA Model

In this section we outline the VQA model used in experiments to evaluate the proposed bottom-up and top-down attention mechanism. *We note that this model was developed by Damien Teney and is not a contribution of this thesis.* For full specifics of the VQA model including a detailed exploration of architectures and hyperparameters, please refer to Teney et al. [2018].

Given a set of spatial image features V , similarly to our captioning model the VQA model uses a ‘soft’ top-down attention mechanism to weight each feature. In this case, however, a question representation is used as context. As illustrated in Figure 5.4, the proposed model implements the well-known joint multimodal embedding of the question and the image, followed by a prediction of regression of scores over a set of candidate answers. This approach has been the basis of numerous previous models [Jabri et al., 2016; Kazemi and Elqursh, 2017; Teney and van den Hengel, 2016]. However, as with our captioning model, implementation decisions are important to ensure that this relatively simple model delivers high performance.

The learned non-linear transformations within the network are implemented with gated hyperbolic tangent activations [Dauphin et al., 2016]. These are a special case of highway networks Srivastava et al. [2015] that have shown a strong empirical advantage over traditional ReLU or tanh layers. Each of our ‘gated tanh’ layers implements a function $f_a : x \in \mathbb{R}^m \rightarrow y \in \mathbb{R}^n$ with parameters $a = \{W, W', b, b'\}$

defined as follows:

$$\tilde{\mathbf{y}} = \tanh(W\mathbf{x} + \mathbf{b}) \quad (5.17)$$

$$\mathbf{g} = \sigma(W'\mathbf{x} + \mathbf{b}') \quad (5.18)$$

$$\mathbf{y} = \tilde{\mathbf{y}} \circ \mathbf{g} \quad (5.19)$$

where σ is the sigmoid activation function, $W, W' \in \mathbb{R}^{n \times m}$ are learned weights, $\mathbf{b}, \mathbf{b}' \in \mathbb{R}^n$ are learned biases, and \circ is the Hadamard (element-wise) product. The vector \mathbf{g} acts multiplicatively as a gate on the intermediate activation $\tilde{\mathbf{y}}$.

Our proposed approach first encodes each question as the hidden state \mathbf{q} of a gated recurrent unit [Cho et al., 2014] (GRU), with each input word represented using a learned word embedding. Similar to Equation 5.8, given the output \mathbf{q} of the GRU, we generate an unnormalised attention weight a_i for each of the k image features \mathbf{v}_i as follows:

$$a_i = \mathbf{w}_a^T f_a([\mathbf{v}_i, \mathbf{q}]) \quad (5.20)$$

where \mathbf{w}_a^T is a learned parameter vector. Equation 5.9 and Equation 5.10 (neglecting subscripts t) are used to calculate the normalised attention weight and the attended image feature $\hat{\mathbf{v}}$. The distribution over possible output responses y is given by:

$$\mathbf{h} = f_q(\mathbf{q}) \circ f_v(\hat{\mathbf{v}}) \quad (5.21)$$

$$p(y) = \sigma(W_o f_o(\mathbf{h})) \quad (5.22)$$

where \mathbf{h} is a joint representation of the question and the image, and $W_o \in \mathbb{R}^{|\Sigma| \times M}$ are learned weights.

5.2.4 Implementation Details

5.2.4.1 Bottom-Up Attention Model

We use the Visual Genome [Krishna et al., 2016] dataset to pretrain our Faster R-CNN bottom-up attention model. The dataset contains 108K images densely annotated with scene graphs containing objects, attributes and relationships, as well as 1.7M visual question answers. However, for pretraining the bottom-up attention model, only the object and attribute annotations are used. We reserve 5K images for validation, and 5K images for future testing, treating the remaining 98K images as training data. Approximately 51K Visual Genome images are also found in the COCO captions dataset [Lin et al., 2014b], so we are careful to avoid contamination

	Object Detections		Attribute Detections	
	mAP@0.5	w-mAP@0.5	mAP@0.5	w-mAP@0.5
COCO [Ren et al., 2015a]	41.5	-	-	-
Visual Genome	10.2	15.1	7.8	27.8

Table 5.1: Faster R-CNN with ResNet-101 detection scores on the COCO and Visual Genome validation sets. We report both micro-averaged and macro-average mean Average Precision at IoU=0.5 (denoted mAP@0.5 and w-mAP@0.5 respectively), where macro-averaging means we compute mAP@0.5 for each class before averaging over classes. Due to the large number of classes, class imbalance, as well as the presence of many overlapping classes and hard-to-localise classes, Faster R-CNN detection performance on Visual Genome is ostensibly poor. However, the resulting features perform well in downstream tasks.

of our COCO validation and test sets. We ensure that any images found in both datasets are contained in the same split in both datasets.

As the object and attribute annotations consist of freely annotated strings, rather than classes, we perform extensive cleaning and filtering of the training data. Beginning with the most frequent 2,000 object strings and 500 attribute strings (hereafter considered classes), we manually remove abstract classes that exhibit poor detection performance in initial experiments. This leaves a final training set containing 1,600 object classes and 400 attribute classes. Note that we do not merge or remove overlapping classes (e.g. ‘person’, ‘man’, ‘guy’), classes with both singular and plural versions (e.g. ‘tree’, ‘trees’) and classes that are difficult to precisely localise (e.g. ‘sky’, ‘grass’, ‘buildings’). Due to the large number of classes, as well as the presence of overlapping classes and classes that can’t be precisely localised (e.g. ‘street’, ‘field’), the detection performance of the Faster R-CNN model trained in this manner is low by conventional standards (refer Table 5.1). However, our focus is the performance of the resulting features in downstream tasks (e.g. image captioning, VQA) rather than detection performance.

In the Faster R-CNN non-max suppression operations we use an intersection-over-union (IoU) threshold of 0.7 for region proposal suppression, and 0.3 for object class suppression. To select salient image regions for input to the image captioning and VQA models, a class detection confidence threshold of 0.2 is used. This allows the number of regions per image k to vary with the complexity of the image, up to a maximum of 100. However, in initial experiments we find that simply selecting the top 36 features in each image works almost as well in both downstream tasks. Since Visual Genome [Krishna et al., 2016] contains a relatively large number of annotations per image, the model is relatively intensive to train. Using 8 Nvidia M40 GPUs, we

take around 5 days to complete 380K training iterations, although we suspect that faster training regimes could also be effective.

5.2.4.2 Captioning and VQA Models

In the captioning model, we set the number of hidden units M in each LSTM to 1,000, the number of hidden units H in the attention layer to 512, and the size of the input word embedding E to 1,000. In training, we use a simple learning rate schedule, beginning with a learning rate of 0.01 which is reduced to zero on a straight-line basis over 60K iterations using a batch size of 100 and a momentum parameter of 0.9. Training using two Nvidia Titan X GPUs takes around 9 hours (including less than one hour for CIDEr optimisation). During optimisation and decoding we use a beam size of 5. When decoding we also enforce the constraint that a single word cannot be predicted twice in a row. Note that in both our captioning and VQA models, image features are fixed and not finetuned.

In the VQA model, we use 300-dimensional word embeddings, initialised with pretrained GloVe vectors [Pennington et al., 2014], and we use hidden states of dimension 512. We train the VQA model using AdaDelta [Zeiler, 2012] and regularise with early stopping. The training of the model takes in the order of 12–18 hours on a single Nvidia K40 GPU. Refer to Teney et al. [2018] for further details of the VQA model implementation.

5.3 Evaluation

5.3.1 Datasets

5.3.1.1 COCO

To evaluate our proposed captioning model, we use the COCO 2014 captions dataset [Lin et al., 2014b]. For validation of model hyperparameters and offline testing, we use the ‘Karpathy’ splits [Karpathy and Fei-Fei, 2015] that have been used extensively for reporting results in prior work. This split contains 113K training images with five captions each, and 5K images respectively for validation and testing. Our COCO test server submission is trained on the entire COCO 2014 training and validation set (123K images). For further details regarding the COCO dataset refer to Section 3.2.

We follow standard practice and perform only minimal text pre-processing, converting all sentences to lower case, tokenizing on white space, and filtering words that do not occur at least five times, resulting in a model vocabulary of 10,010 words. To evaluate caption quality, we use the standard automatic evaluation metrics, namely

SPICE (refer Chapter 4 and Anderson et al. [2016]), CIDEr [Vedantam et al., 2015], METEOR [Denkowski and Lavie, 2014], ROUGE-L [Lin, 2004] and Bleu [Papineni et al., 2002].

5.3.1.2 VQA v2.0

To evaluate our proposed VQA model, we use the recently introduced VQA v2.0 dataset [Goyal et al., 2017], which attempts to minimise the effectiveness of learning dataset priors by balancing the answers to each question. The dataset, which was used as the basis of the 2017 VQA Challenge³, contains 1.1M questions with 11.1M answers relating to COCO images. We perform standard question text preprocessing and tokenization. Questions are trimmed to a maximum of 14 words for computational efficiency. The set of candidate answers is restricted to correct answers in the training set that appear more than 8 times, resulting in an output vocabulary size of 3,129.

When training the VQA model, we augment the VQA v2.0 training data with Visual Genome [Krishna et al., 2016] question and answer pairs provided the correct answer is already present in model’s answer vocabulary. This represents about 30% of the available Visual Genome data, or 485K questions. Our VQA test server submissions are also trained on the VQA v2.0 validation set. To evaluate answer quality, we report accuracies using the standard VQA metric [Antol et al., 2015], which takes into account the occasional disagreement between annotators for the ground truth answers. For further details regarding the VQA v2.0 and Visual Genome datasets refer to Section 3.3 and Section 3.4 respectively.

5.3.2 ResNet Baseline

To quantify the impact of bottom-up attention, in both our captioning and VQA experiments we evaluate our full model (*Up-Down*) against prior work as well as an ablated baseline. In each case, the baseline (*ResNet*), uses a ResNet [He et al., 2016a] CNN pretrained on ImageNet [Russakovsky et al., 2015] to encode each image in place of the bottom-up attention mechanism.

In image captioning experiments, similarly to previous work [Rennie et al., 2017] we encode the full-sized input image with the final convolutional layer of Resnet-101, and use bilinear interpolation to resize the output to a fixed size spatial representation of 10×10 . This is equivalent to the maximum number of spatial regions used in our full model. In VQA experiments, we encode the resized input image with ResNet-200 [He et al., 2016b]. In separate experiments we use evaluate the effect of

³<http://www.visualqa.org/challenge.html>

	Bleu-1	Bleu-4	METEOR	ROUGE-L	CIDEr	SPICE
Cross-Entropy Optimisation:						
SCST:Att2in	-	31.3	26.0	54.3	101.3	-
SCST:Att2all	-	30.0	25.9	53.4	99.4	-
Ours: ResNet	74.5	33.4	26.1	54.4	105.4	19.2
Ours: Up-Down	77.2	36.2	27.0	56.4	113.5	20.3
Ours: Relative Improvement	4%	8%	3%	4%	8%	6%
CIDEr Optimisation:						
SCST:Att2in	-	33.3	26.3	55.3	111.4	-
SCST:Att2all	-	34.2	26.7	55.7	114.0	-
Ours: ResNet	76.6	34.0	26.5	54.9	111.1	20.2
Ours: Up-Down	79.8	36.3	27.7	56.9	120.1	21.4
Ours: Relative Improvement	4%	7%	5%	4%	8%	6%

Table 5.2: Single-model image captioning performance on the COCO Karpathy test split. Our baseline ResNet model obtains similar results to SCST [Rennie et al., 2017], the existing state of the art on this test set. Illustrating the contribution of bottom-up attention, our Up-Down model achieves 3–8% relative improvements across all metrics, regardless of whether cross-entropy loss or CIDEr-based optimisation methods are used.

	SPICE	Objects	Attributes	Relations	Color	Count	Size
Cross-Entropy Optimisation:							
Ours: ResNet	19.2	35.4	8.6	5.3	12.2	4.1	3.9
Ours: Up-Down	20.3	37.1	9.2	5.8	12.7	6.5	4.5
CIDEr Optimisation:							
Ours: ResNet	20.2	37.0	9.2	6.1	10.6	12.0	4.3
Ours: Up-Down	21.4	39.1	10.0	6.5	11.4	18.4	3.2

Table 5.3: Breakdown of SPICE F-scores over various tuple subcategories on the COCO Karpathy test split. Our Up-Down model outperforms the ResNet baseline at identifying objects, as well as detecting object attributes and the relations between objects.

varying the size of the spatial output from its original size of 14×14 , to 7×7 (using bilinear interpolation) and 1×1 (i.e., mean pooling without attention).

5.3.3 Image Captioning Results

In Table 5.2 we report the performance of our full model and the ResNet baseline in comparison to the existing state of the art Self-critical Sequence Training [Rennie et al., 2017] (SCST) approach on the test portion of the Karpathy splits. For fair comparison, results are reported for models trained with both standard cross-entropy loss, and models optimised for CIDEr score. Note that the SCST approach uses ResNet-101 encoding of full images, similar to our ResNet baseline. All results are reported for a single model with no fine-tuning of the input ResNet / R-CNN model. However, the SCST results are selected from the best of four random initialisations, while our results are outcomes from a single initialisation.

Relative to the SCST models, our ResNet baseline obtains slightly better performance under cross-entropy loss, and slightly worse performance when optimised for CIDEr score. After incorporating bottom-up attention, our full Up-Down model shows significant improvements across all metrics regardless of whether cross-entropy loss or CIDEr optimisation is used. Using just a single model, we obtain the best reported results for the Karpathy test split. As illustrated in Table 5.3, the contribution from bottom-up attention is broadly based, illustrated by improved performance in terms of identifying objects, object attributes and also the relationships between objects (refer to Section 4.3.3 for a discussion of these SPICE subcategories).

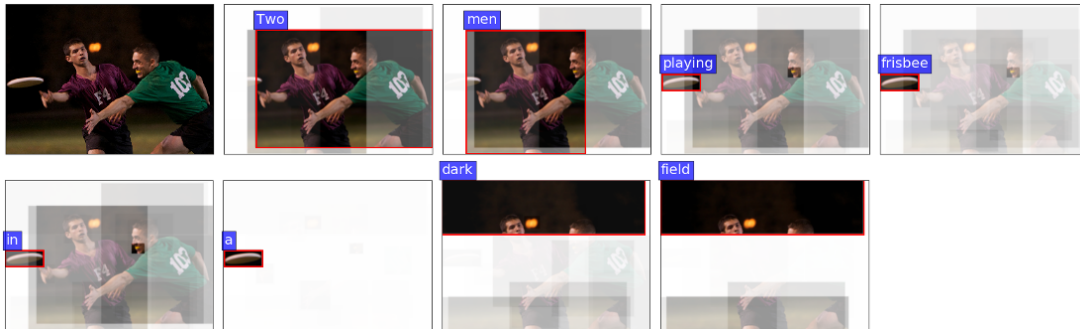
Table 5.4 reports the performance of 4 ensembled models trained with CIDEr optimisation on the official COCO evaluation server, along with the highest ranking previously published results. At the time of submission (18 July 2017), we outperform all other test server submissions on all reported evaluation metrics.

To help qualitatively evaluate our attention methodology, in Figure 5.5 we visualise the attended image regions for different words generated by our Up-Down captioning model. As indicated by this example, our approach is equally capable of focusing on fine details or large image regions. This capability arises because the attention candidates in our model consist of many overlapping regions with varying scales and aspect ratios – each aligned to an object, several related objects, or an otherwise salient image patch.

Unlike conventional approaches, when a candidate attention region corresponds to an object, or several related objects, all the visual concepts associated with those objects appear to be spatially co-located – and are processed together. In other words, our approach is able to consider all of the information pertaining to an object at once.

	Bleu-1		Bleu-2		Bleu-3		Bleu-4	
	c5	c40	c5	c40	c5	c40	c5	c40
Review Net [Yang et al., 2016b]	72.0	90.0	55.0	81.2	41.4	70.5	31.3	59.7
Adaptive [Lu et al., 2017]	74.8	92.0	58.4	84.5	44.4	74.4	33.6	63.7
PG-BCMR [Liu et al., 2017b]	75.4	-	59.1	-	44.5	-	33.2	-
SCST:Att2all [Rennie et al., 2017]	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5
LSTM-A ₃ [Yao et al., 2017b]	78.7	93.7	62.7	86.7	47.6	76.5	35.6	65.2
Ours: Up-Down	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5
	METEOR		ROUGE-L		CIDEr		SPICE	
	c5	c40	c5	c40	c5	c40	c5	c40
Review Net [Yang et al., 2016b]	25.6	34.7	53.3	68.6	96.5	96.9	18.5	64.9
Adaptive [Lu et al., 2017]	26.4	35.9	55.0	70.5	104.2	105.9	19.7	67.3
PG-BCMR [Liu et al., 2017b]	25.7	-	55	-	101.3	-	-	-
SCST:Att2all [Rennie et al., 2017]	27.0	35.5	56.3	70.7	114.7	116.7	20.7	68.9
LSTM-A ₃ [Yao et al., 2017b]	27	35.4	56.4	70.5	116	118	-	-
Ours: Up-Down	27.6	36.7	57.1	72.4	117.9	120.5	21.5	71.5

Table 5.4: Highest ranking published image captioning results on the online COCO test server. Our submission, an ensemble of 4 models optimised for CIDEr with different initialisations, outperformed previously published work on all reported metrics. At the time of submission (18 July 2017), we also outperformed all unpublished test server submissions.



Two men playing frisbee in a dark field.

Figure 5.5: Example of a generated caption showing attended image regions. For each generated word, we visualise the attention weights on individual pixels, outlining the region with the maximum attention weight in red. Avoiding the conventional trade-off between coarse and fine levels of detail, our model focuses on both closely-cropped details, such as the frisbee and the green player’s mouthguard when generating the word ‘playing’, as well as large regions, such as the night sky when generating the word ‘dark’.

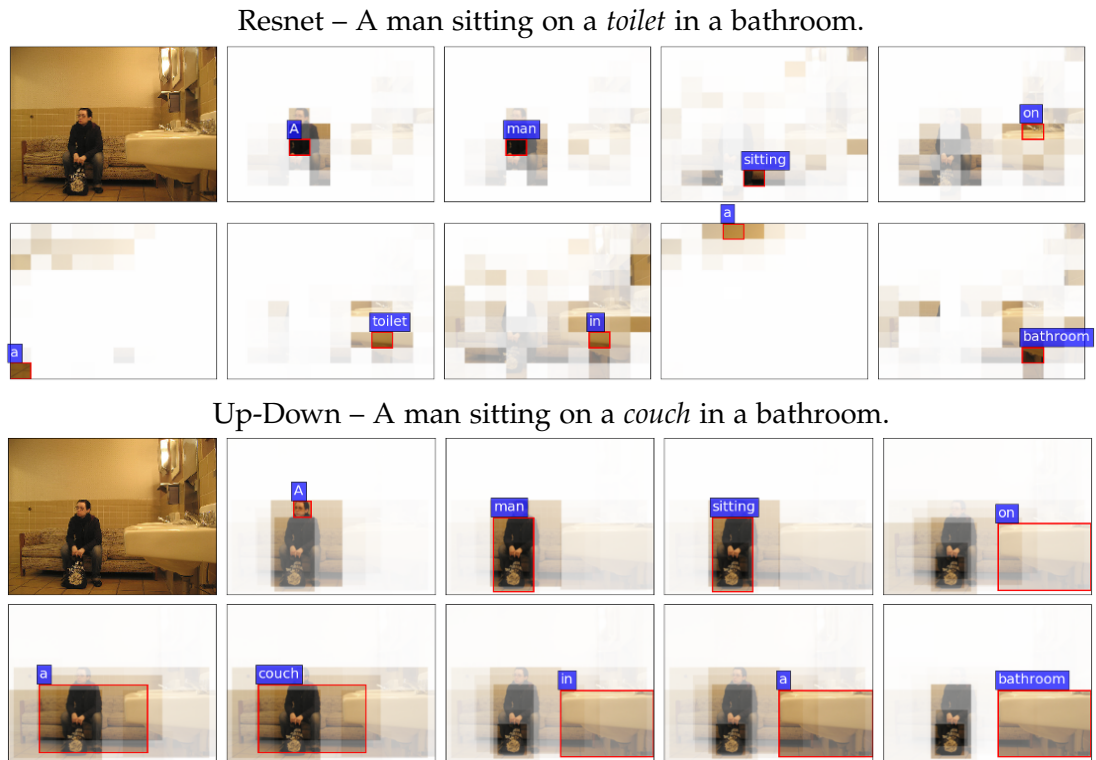


Figure 5.6: Qualitative differences between attention methodologies in caption generation. The selected image is unusual because it depicts a bathroom containing a couch but no toilet. Nevertheless, the baseline ResNet model (top) hallucinates a toilet, presumably from language priors. In contrast, our Up-Down model (bottom) identifies the out-of-context couch and provides more interpretable attention weights.

This is also a natural way for attention to be implemented. In the human visual system, the problem of integrating the separate features of objects in the correct combinations is known as the feature binding problem, and experiments suggest that attention plays a central role in the solution [Treisman and Gelade, 1980; Treisman, 1982]. We illustrate the qualitative differences between attention methodologies in Figure 5.6, demonstrating the improved interpretability of the bottom-up and top-down attention weights.

5.3.4 VQA Results

In Table 5.5 we report the single model performance of our full Up-Down VQA model relative to several ResNet baselines on the VQA v2.0 validation set. The addition of bottom-up attention provides a significant improvement over the best ResNet baseline across all question types, even though the ResNet baseline uses approximately

	Yes/No	Number	Other	Overall
Ours: ResNet (1×1)	76.0	36.5	46.8	56.3
Ours: ResNet (14×14)	76.6	36.2	49.5	57.9
Ours: ResNet (7×7)	77.6	37.7	51.5	59.4
Ours: Up-Down	80.3	42.8	55.8	63.2
Relative Improvement	3%	14%	8%	6%

Table 5.5: Single-model performance on the VQA v2.0 validation set. The use of bottom-up attention in the Up-Down model provides a significant improvement over the best ResNet baseline across all question types, even though the ResNet baselines use almost twice as many convolutional layers.

	Yes/No	Number	Other	Overall
Prior [Goyal et al., 2017]	61.2	0.4	1.2	26.0
Language-only [Goyal et al., 2017]	67.0	31.6	27.4	44.3
d-LSTM+n-I [Lu et al., 2015]	73.5	35.2	41.8	54.2
MCB [Fukui et al., 2016]	78.8	38.3	53.4	62.3
JuneflowerIvaNlpr	81.1	41.6	57.8	65.7
UPMC-LIP6	82.1	41.1	57.1	65.7
LV NUS	81.9	46.3	58.3	66.8
Athena	82.5	44.2	60.0	67.6
HDU-USYD-UNCC	84.5	45.4	59.0	68.1
Ours: Up-Down	86.6	48.6	61.2	70.3

Table 5.6: VQA v2.0 test-standard server accuracy as at 8 August 2017, ranking our submission against the highest ranked published and unpublished work. The result for the d-LSTM+n-I and MCB models are as reported in Goyal et al. [2017]. Our approach, an ensemble of 30 models, outperformed all other leaderboard entries, achieving first place in the 2017 VQA Challenge.

twice as many convolutional layers. Table 5.6 reports the performance of 30 ensemble models on the official VQA 2.0 test-standard evaluation server, along with the previously published baseline results and the highest ranking other entries. At the time of submission (8 August 2017), we outperform all other test server submissions. Our submission also achieved first place in the 2017 VQA Challenge. We include an example of VQA attention in Figure 5.7.

5.4 Chapter Summary

In this chapter we present a novel combined bottom-up and top-down visual attention mechanism. Our approach enables attention to be calculated more naturally



Question: What room are they in? Answer: kitchen

Figure 5.7: VQA example illustrating attention output. Given the question ‘What room are they in?’, the model focuses on the stovetop, generating the correct answer ‘kitchen’.

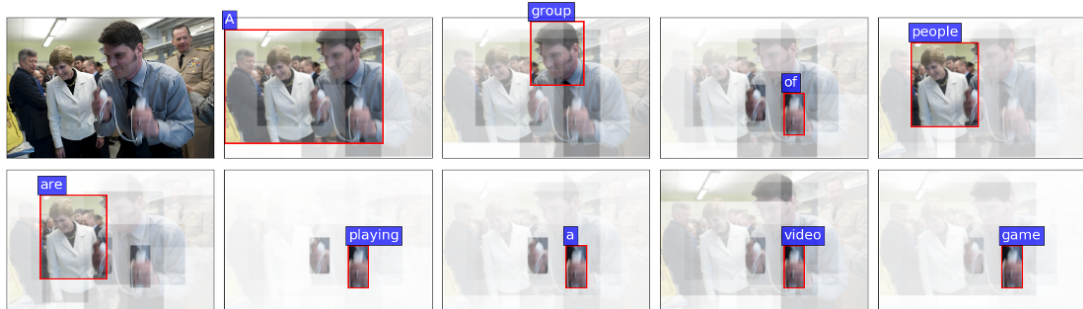
at the level of objects and other salient regions, improving the interpretability of the resulting attention weights. Empirically, we find that the inclusion of bottom-up attention has a significant positive benefit for image captioning. Our results on the COCO test server establish a new state of the art for the task, achieving CIDEr, SPICE, and Bleu-4 scores of 117.9, 21.5, and 36.9, respectively (outperforming all published and unpublished work at the time of test server submission). Demonstrating the broad applicability of the method, we additionally present a VQA model using the same bottom-up attention features. Using this model we obtain first place in the 2017 VQA Challenge, achieving 70.3% overall accuracy on the VQA v2.0 test-standard server (and 69.0% overall accuracy on the VQA v2.0 test-challenge dataset split).

Although visual attention mechanisms have been widely studied, perhaps surprisingly our work is one of the first to more carefully consider the determination of attention candidates (shifting from a uniform grid of neural receptive fields to more natural proposals based on objects and other salient image regions). Given the complexity of biological attention mechanisms, which can be feature-based, object-based, spatial and temporal [Scholl, 2001], we will be surprised if further empirical investigation of visual attention mechanisms does not yield positive results. There are also exciting opportunities for future work in the unsupervised and semi-supervised training of visual attention mechanisms.

At a high level, our work more closely unifies tasks involving visual and linguistic understanding with recent progress in object detection. We hope to see vision and language researchers continue to take advantage of ongoing research into object detection—as we did by using Faster R-CNN [Ren et al., 2015b]—rather than relying

exclusively on convolutional neural networks (CNNs) pretrained for whole-image classification. In our work, the Faster R-CNN model is pretrained and fixed, meaning that input image region locations and their associated feature representations are not influenced by the loss function of the task at hand. Future work could investigate training the Faster R-CNN (including the region proposal network) in an end-to-end fashion. Also, as our model utilizes only object and attributes, incorporating predictions of the relations between pairs of objects into VQA and captioning models may also provide benefits. Nevertheless, while these are important research directions, the immediate benefits of our approach may be captured by simply replacing pretrained CNN features with pretrained bottom-up attention features (i.e., our pretrained Faster R-CNN [Ren et al., 2015a] outputs), which we have made available to the community. We provide further examples in Figures 5.8–5.11.

A group of people are playing a video game.



A brown sheep standing in a field of grass.

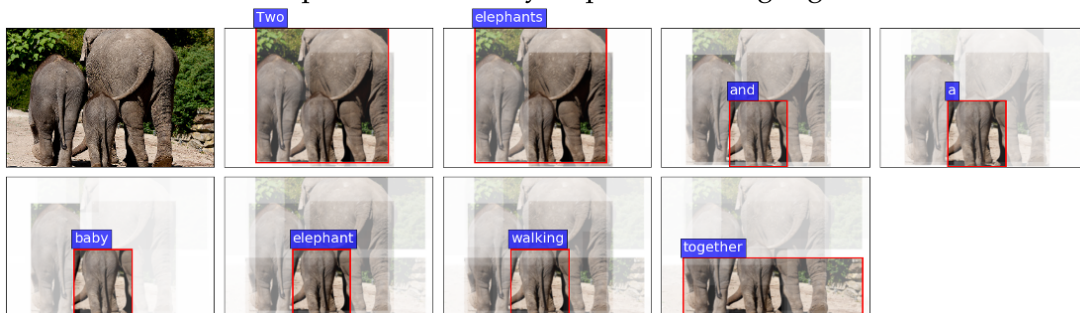


Two hot dogs on a tray with a drink.



Figure 5.8: Further examples of generated captions showing attended image regions. Attention is given to fine details, such as: (1) the man’s hands holding the game controllers in the top image, and (2) the sheep’s legs when generating the word ‘standing’ in the middle image. Our approach can avoid the trade-off between coarse and fine levels of detail.

Two elephants and a baby elephant walking together.



A close up of a sandwich with a stuffed animal.



A dog laying in the grass with a frisbee.



Figure 5.9: Further examples of generated captions showing attended image regions. The first example suggests an understanding of spatial relationships when generating the word ‘together’. The middle image demonstrates the successful captioning of a compositionally novel scene. The bottom example is a failure case. The dog’s pose is mistaken for laying, rather than jumping – possibly due to poor salient region cropping that misses the dog’s head and feet.

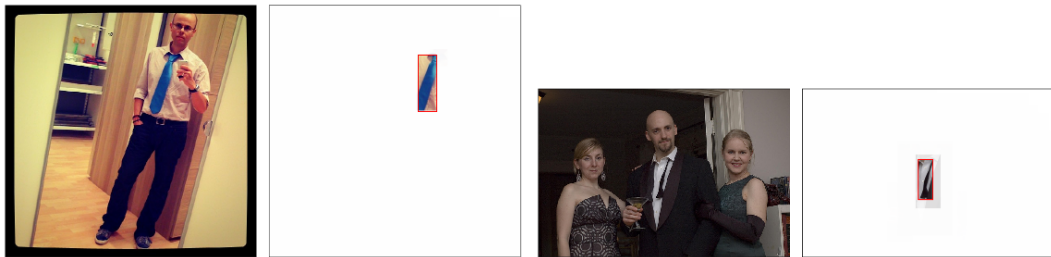
Question: What color is illuminated on the traffic light? Answer left: green. Answer right: red.



Question: What is the man holding? Answer left: phone. Answer right: controller.



Question: What color is his tie? Answer left: blue. Answer right: black.



Question: What sport is shown? Answer left: frisbee. Answer right: skateboarding.

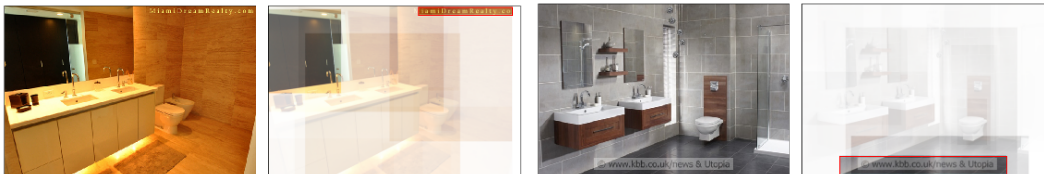


Question: Is this the handlebar of a motorcycle? Answer left: yes. Answer right: no.



Figure 5.10: Further examples of successful visual question answering results, showing attended image regions.

Question: What is the name of the realty company? Answer left: none. Answer right: none.



Question: What is the bus number? Answer left: 2. Answer right: 23.



Question: How many cones have reflective tape? Answer left: 2. Answer right: 1.



Question: How many oranges are on pedestals? Answer left: 2. Answer right: 2.



Figure 5.11: Examples of visual question answering (VQA) failure cases. Although our simple VQA model has limited reading and counting capabilities, the attention maps are often correctly focused.

Guided Image Captioning using Constrained Beam Search

In Chapter 5 we introduced a bottom-up and top-down visual attention mechanism, demonstrating improved performance on standard image captioning and visual question answering (VQA) datasets [Chen et al., 2015; Goyal et al., 2017]. However, as discussed in Chapter 3, the images underlying both these datasets are primarily concerned with only 91 objects and their interactions. As a result, models trained on these datasets do not generalise well to out-of-domain images containing novel scenes or objects [Tran et al., 2016]. This limitation severely hinders the use of these models in real applications. To address this problem, in this chapter we propose a flexible approach to combining the output of image taggers and image captioning models at test time, without any joint training, in order to scale existing image captioning models to many more visual concepts. Our approach—using a novel constraint-based decoding algorithm we describe as *constrained beam search*—achieves state of the art results on a held-out version of the COCO image captioning dataset [Hendricks et al., 2016].

6.1 Test-Time Novel Object Captioning

We wish to caption an image for which some high-confidence image tags (or other text fragments) are available during caption generation (i.e., at test time). These image tags may originate from a predictive model such as an image tagger [Chen et al., 2013; Zhang et al., 2016b] or an object detector [Ren et al., 2015b; Krause et al., 2016]. Alternatively, as many image collections are annotated with semantic attributes and / or object classes, the image tags may reflect the ground-truth. As illustrated in Figure 6.1, our goal is to use these image tags to improve the output of a recurrent neural network (RNN) image captioning model. In particular, we would like to successfully caption images containing novel objects—i.e., objects that are not represented in the

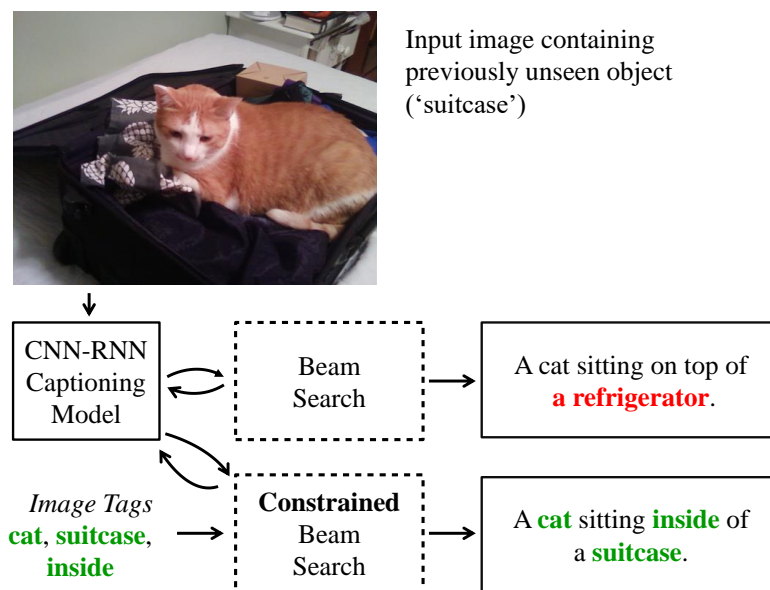


Figure 6.1: We successfully caption images containing previously unseen objects by incorporating image tags during RNN decoding. Actual example from experiments in Section 6.4.2.

available image-caption training data—when given the correct object class as an image tag. This scenario is identical to the problem of novel object captioning posed by Hendricks et al. [2016], except that we will make use of image tags at test time, rather than during training. Our problem formulation is arguably more challenging, as it prevents joint training of the image tagging and image captioning components of the resulting system. However, our loosely-coupled approach also offers several advantages, as it permits the re-use of existing models, which can be combined at test time, and also allows captioning models to take advantage of ground-truth image tags when available.

This scenario poses two key challenges. First, RNNs are generally opaque, and difficult to influence at test time. Second, for previously unseen objects, image tags will include words that are not present in the RNN vocabulary. We address the first challenge (guidance) by proposing *constrained beam search* to guarantee the inclusion of selected words or phrases in the output of an RNN, while leaving the model free to determine the syntax and additional details. Constrained beam search is an approximate search algorithm capable of enforcing any constraints over resulting output sequences that can be expressed in a finite-state machine (FSM). With regard to the second challenge (vocabulary), empirically we demonstrate that an RNN can successfully generalise from similar words if both the input embedding and output

projection layers contain fixed, pretrained word embeddings.

6.2 Related Work

Several papers have proposed models intended to describe images containing objects for which no image-caption training data exists, i.e., novel object captioning. The Deep Compositional Captioner (DCC) [Hendricks et al., 2016] uses a CNN image tagger to predict words that are relevant to an image, combined with an RNN language model to estimate probabilities over word sequences. The tagger and language models are pretrained separately, then finetuned jointly using the available image-caption data.

Building on the DCC approach, the Novel Object Captioner (NOC) [Venugopalan et al., 2017] is contemporary work with ours that also uses pretrained word embeddings in both the input and output layers of the language model. Since the publication of our approach, additional methods have also been proposed using a word copying mechanism [Yao et al., 2017a] and neural slot-filling [Lu et al., 2018]. More generally, the effectiveness of incorporating semantic attributes (i.e., image tags) into caption model training for in-domain data has been established by several works [Fang et al., 2015; Wu et al., 2016a; Elliot and de Vries, 2015].

Overall, our work differs fundamentally from these approaches as we do not attempt to introduce semantic attributes, image tags or other text fragments into the learning algorithm. Instead, we incorporate text fragments during model decoding. To the best of our knowledge we are the first to consider this more loosely-coupled approach, which allows the model to take advantage of information not available at training time, and avoids the need to retrain the captioning model if the source of text fragments is changed.

6.3 Approach

In this section we briefly review beam search before describing in detail the constrained beam search algorithm, its application to image captioning, and our approach to expanding model vocabularies with pretrained word embeddings.

6.3.1 RNN Decoding with Beam Search

While various approaches to image caption generation have been considered, a large body of recent work—including our own work in Chapter 5—is dedicated to neural network approaches [Donahue et al., 2015; Mao et al., 2015; Karpathy and Fei-Fei,

2015; Vinyals et al., 2015; Devlin et al., 2015a]. These approaches typically use a pretrained Convolutional Neural Network (CNN) image encoder, combined with a Recurrent Neural Network (RNN) decoder trained to predict the next output word, conditioned on previous words and the image. In each case the decoding process remains the same—captions are generated by searching over output sequences greedily or with beam search. Similar decoding approaches are typically used for many sequence generation tasks including machine translation [Sutskever et al., 2014].

More formally, let $\mathbf{y} = (y_1, \dots, y_T)$ denote an output sequence of length T containing words or other tokens from vocabulary Σ . Given an RNN modelling a probability distribution over such sequences, the RNN decoding problem is to find the output sequence with the maximum log-probability, where the log probability of any partial sequence \mathbf{y}_t of length t is typically given by $\Theta(\mathbf{y}_t) = \sum_{j=1}^t \log p(y_j | y_1, \dots, y_{j-1})$. As illustrated in Algorithm 1, at each decoding time step t beam search stores only the b most likely partial sequences, where b is known as the beam size. We will denote the set of all partial solutions held at the start of time t by $B_{t-1} = \{\mathbf{y}_{t-1,1}, \dots, \mathbf{y}_{t-1,b}\}$. At each time step t , a candidate set E_t is generated by considering all possible next word extensions:

$$E_t = \{(\mathbf{y}_{t-1}, w) \mid \mathbf{y}_{t-1} \in B_{t-1}, w \in \Sigma\} \quad (6.1)$$

The beam B_t is updated by retaining only the b most likely sequences in E_t . This can be trivially implemented by sorting the partial sequences in E_t by their log-probabilities and retaining the top b . Initialisation is performed by inserting an empty sequence into the beam, i.e. $B_0 := \{\epsilon\}$ such that $E_1 = \Sigma$. The algorithm terminates when the beam contains a completed sequence (e.g., containing an end marker) with higher log probability than all incomplete sequences. Refer to Section 2.4 or Koehn [2010] for further background.

6.3.2 Constrained Beam Search

We introduce constrained beam search, a multiple beam approximate search algorithm that enforces constraints in the sequence generation process. To fix ideas, suppose that we wish to generate word sequences containing at least one word from each disjunctive constraint set $D_1 = \{\text{chair}, \text{chairs}\}$ and $D_2 = \{\text{desk}, \text{table}\}$. As an example, the word sequence ‘a table surrounded by chairs’ would satisfy these constraints.

We first need a method for checking if the constraints are satisfied by a given sequence. For this we can use the finite-state machine (FSM) illustrated in Figure 6.2, with start state s_0 and accepting state s_3 , which *recognises* sequences satisfying

these constraints. More generally, any set of constraints that can be represented with a regular expression can also be expressed as an FSM (either deterministic or non-deterministic) that recognises sequences satisfying those constraints [Sipser, 2012]. Therefore, we will assume that constraints are provided in the form of a FSM that accepts sequences satisfying those constraints, and that the structure of the FSM is determined by the problem setting (in our work, the FSM is determined by the detected image tags, as described in Section 6.3.3). Formally, an FSM is defined by $(\Sigma, S, s_0, \delta, F)$ where Σ is the model vocabulary, S is the set of states, $s_0 \in S$ is the initial state, $\delta : S \times \Sigma \rightarrow S$ is the state-transition function that maps states and words to states, and $F \subseteq S$ is the set of final or accepting states.

To decode output sequences under these constraints, a naive approach might impose the constraints on sequences produced at the end of beam search. This can be implemented by performing beam search decoding, then returning the highest log-probability sequence that is accepted by the FSM. However, if the constraints are only satisfied by relatively low probability output sequences, it is likely that an infeasibly large beam would be required in order to produce sequences that satisfy the constraints (and there is no guarantee of generating a satisfying sequence with a given beam size).

An alternative approach has been investigated in the context of RNN poetry generation [Ghazvininejad et al., 2016]. Under this approach, all partial sequences generated by Equation 6.1 must satisfy the FSM *during each step of beam search*. We note that similar ideas have been explored in the context of machine translation using n-gram language models [Allauzen et al., 2014]. However, while this approach is effective for ensuring that generated poems obey formal sonnet constraints, it cannot be applied in our example as our FSM cannot be satisfied by sequences of arbitrary length—the shortest satisfying sequence must contain at least two words. Instead, we propose a *multiple beam* approach under which only complete sequences are required to satisfy the FSM, and sequences that satisfy different constraint subsets do not compete with each other for membership in a search beam. For example, in Figure 6.1 the constraint word ‘suitcase’ is going to have an extremely low probability as it was never seen in the caption training data, so partial captions containing ‘suitcase’ will always be much lower probability than partial captions that do not contain ‘suitcase’. Our approach maintains separate beams for captions containing ‘suitcase’ and for captions not containing ‘suitcase’, and allows captions to move between beams during generation.

In detail, to generate constrained sequences we take as input an FSM that recognises sequences satisfying the required constraints, and use the following multiple-beam decoding algorithm. For each state $s \in S$ in the FSM, a corresponding search

Algorithm 1 Beam search decoding

```

1: procedure BS( $\Theta, b, T, \Sigma$ ) ▷ With beam size  $b$  and vocabulary  $\Sigma$ 
2:    $B \leftarrow \{\epsilon\}$  ▷  $\epsilon$  is the empty string
3:   for  $t = 1, \dots, T$  do
4:      $E \leftarrow \{(\mathbf{y}, w) \mid \mathbf{y} \in B, w \in \Sigma\}$  ▷ All one-word extensions of sequences in  $B$ 
5:      $B \leftarrow \operatorname{argmax}_{E' \subset E, |E'|=b} \sum_{\mathbf{y} \in E'} \Theta(\mathbf{y})$  ▷ The  $b$  most probable extensions in  $E$ 
6:   return  $\operatorname{argmax}_{\mathbf{y} \in B} \Theta(\mathbf{y})$  ▷ The most probable sequence

```

Algorithm 2 Constrained beam search decoding

```

1: procedure CBS( $\Theta, b, T, (\Sigma, S, s_0, \delta, F)$ ) ▷ With finite state machine
2:   for  $s \in S$  do
3:      $B^s \leftarrow \{\epsilon\}$  if  $s = s_0$  else  $\emptyset$  ▷ Each state  $s$  has a beam  $B^s$ 
4:   for  $t = 1, \dots, T$  do
5:     for  $s \in S$  do ▷ Extend sequences through state-transition function  $\delta$ 
6:        $E^s \leftarrow \cup_{s' \in S} \{(\mathbf{y}, w) \mid \mathbf{y} \in B^{s'}, w \in \Sigma, \delta(s', w) = s\}$ 
7:        $B^s \leftarrow \operatorname{argmax}_{E' \subset E^s, |E'|=b} \sum_{\mathbf{y} \in E'} \Theta(\mathbf{y})$  ▷ The  $b$  most probable extensions
      in  $E^s$ 
8:   return  $\operatorname{argmax}_{\mathbf{y} \in \cup_{s \in F} B^s} \Theta(\mathbf{y})$  ▷ The most probable accepted sequence

```

beam B^s is maintained. As in beam search, each B^s is a set containing at most b output sequences, where b is the beam size. At each time step t , a candidate set E_t^s is generated for each beam, as follows:

$$E_t^s = \bigcup_{s' \in S} \{(\mathbf{y}_{t-1}, w) \mid \mathbf{y}_{t-1} \in B_{t-1}^{s'}, w \in \Sigma, \delta(s', w) = s\} \quad (6.2)$$

where $\delta : S \times \Sigma \mapsto S$ is the FSM state-transition function that maps states and words to states. In effect, each candidate set E_t^s includes next word extensions from every beam, provided the resulting sequence satisfies the corresponding state s . In other words, the FSM state-transition function determines the appropriate candidate set for each possible extension of a partial sequence. This ensures that sequences in accepting states must satisfy all constraints as they have been recognised by the FSM during the decoding process. Similarly to beam search, each B_t^s is updated by retaining the b most likely sequences in its candidate set E_t^s . This ensures that only partial sequences that satisfy the same constraints will compete with each other.

Initialisation is performed by inserting an empty sequence into the beam associated with the start state s_0 , so $B_0^0 := \{\epsilon\}$ and $B_0^{i \neq 0} := \emptyset$. The algorithm terminates when an accepting state contains a completed sequence (e.g., containing an end marker) with higher log probability than all incomplete sequences. In the example contained in Figure 6.2, on termination captions in Beam 0 will not contain any

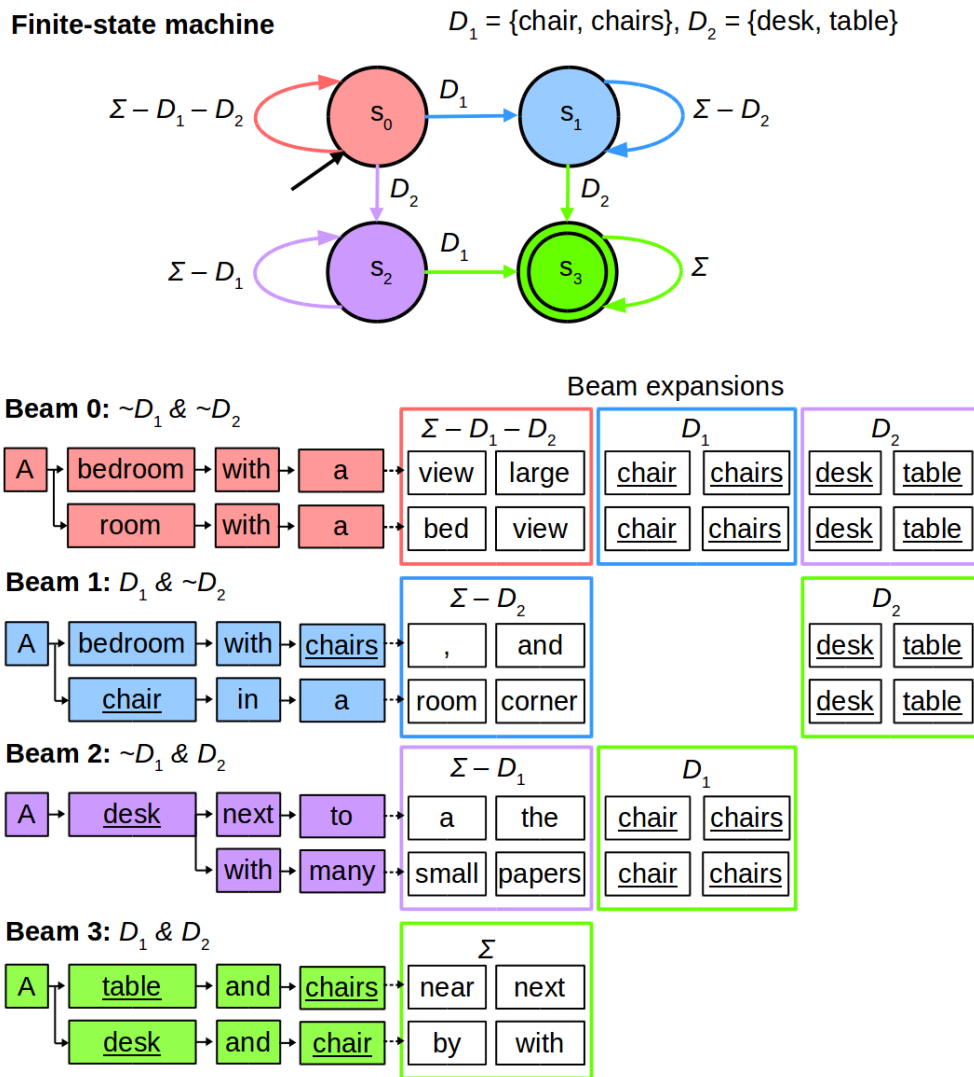


Figure 6.2: Example of constrained beam search decoding. Each output sequence must include the words ‘chair’ or ‘chairs’, and ‘desk’ or ‘table’ from vocabulary Σ . A finite-state machine (FSM) that recognises valid sequences is illustrated at top. Each state in the FSM corresponds to a beam in the search algorithm (bottom). FSM state transitions determine the destination beam for each possible sequence extension. Valid sequences are found in Beam 3, corresponding to FSM accepting state s_3 .

words from D_1 or D_2 , captions in Beam 1 will contain a word from D_1 but not D_2 , captions in Beam 2 will contain a word from D_2 but not D_1 , and captions in Beam 3 will contain a word from both D_1 and D_2 .

To summarise, in Algorithms 1 and 2 we provide an overview of the constrained beam search algorithm, contrasting it directly with beam search. Both algorithms take as inputs a scoring function which we define by $\Theta(\mathbf{y}) = \log p_\theta(\mathbf{y})$, a beam size b , the maximum sequence length T and the model vocabulary Σ . However, the constrained beam search algorithm additionally takes an FSM as input, and guarantees that the sequence returned will be accepted by the FSM.

Computational complexity Compared to beam search with the same beam size, constrained beam search performs additional computation since it maintains multiple beams during decode (one for each FSM state). Specifically, if γ is the computational cost of a single forward pass through an unrolled recurrent neural network (e.g., the cost of decoding a single sequence), with beam size b the cost of beam search decoding is given by $b \cdot \gamma$. The cost of constrained beam search decoding is given by $|S| \cdot b \cdot \gamma$, where $|S|$ is the number of FSM states. Although the computational cost of training increases linearly with the number of FSM states, we note that for any particular application FSM construction is a modelling choice and there are many existing FSM compression and state reduction methods available.

6.3.3 Application to Image Captioning

In our experiments applying constrained beam search to image captioning, we implement constraints to ensure that high-confidence text fragments—sourced from a predictive model or ground-truth annotations—are mentioned in the resulting image captions. Given single-word image tags (as in Section 6.4.2), to allow the captioning model freedom to choose word forms we use WordNet [Miller, 1995] to map each image tag to a disjunctive set $D_i = \{w_{i,1}, \dots, w_{i,n_i}\}$ containing the words in vocabulary Σ that share the same lemma. Then, for each image we generate a FSM encoding the conjunction of disjunctions $C = \{D_1, \dots, D_m\}$ containing one disjunctive set for each of m image tags. A sequence \mathbf{y} satisfies constraint C iff for each $D_i \in C$, there exists a $w_{i,j} \in D_i$ such that $w_{i,j} \in \mathbf{y}$. As illustrated with the example in Figure 6.2, which contains two disjunctive sets— D_1 and D_2 —and four states and beams, the algorithm maintains one beam for each of the 2^m subsets of disjunctive constraints D_i . However, in practice $m \leq 4$ is sufficient for the captioning task, and with these values our GPU constrained beam search implementation based on Caffe [Jia et al., 2014] generates 40K captions for COCO in well under an hour. The use of WordNet lemmas

adds minimal complexity to the algorithm, as the number of FSM states, and hence the number of search beams, is not increased by adding words to a disjunctive set. Given sequence-based constraints (as in Section 6.4.3), we do not use disjunctive sets. In this case, the number of FSM states, and the number of search beams, is linear in the length of the subsequence (the number of states is equal to number of words in a phrase plus one).

Our approach to test-time novel object image captioning could be applied to any existing CNN-RNN captioning model that can be decoded using beam search, e.g., [Donahue et al., 2015; Mao et al., 2015; Karpathy and Fei-Fei, 2015; Vinyals et al., 2015; Devlin et al., 2015a]. However, for empirical evaluation we use the Long-term Recurrent Convolutional Network [Donahue et al., 2015] (LRCN) as our base model. The LRCN consists of a CNN visual feature extractor followed by two LSTM layers [Hochreiter and Schmidhuber, 1997], each with 1K hidden units. The model is factored such that the bottom LSTM layer receives only language input, consisting of the embedded previous word. At test time the previous word is the predicted model output, but during training the ground-truth preceding word is used. The top LSTM layer receives the output of the bottom LSTM layer, as well as a per-timestep static copy of the CNN features extracted from the input image.

Given the LSTM operation $h_t = \text{LSTM}(x_t, h_{t-1})$ described in Section 2.4, and using superscripts to represent the LSTM layer index, the input vector for the bottom LSTM is an encoding of the previous word, given by:

$$x_t^1 = W_e \Pi_t \quad (6.3)$$

where W_e is a word embedding matrix, and Π_t is a one-hot column vector identifying the input word at timestep t . The top LSTM input vector comprises the concatenated output of the bottom LSTM and the CNN feature descriptor of the image I , given by:

$$x_t^2 = (h_t^1, \text{CNN}_\theta(I)) \quad (6.4)$$

For the CNN component of the model, we evaluate using the 16-layer VGG [Simonyan and Zisserman, 2015] model and the 50-layer Residual Net [He et al., 2016a], pretrained on ImageNet [Russakovsky et al., 2015] in both cases. Unlike Donahue et al. [2015], we do not fix the CNN weights during initial training, as we find that performance improves if all training is conducted end-to-end. In training, we use only very basic data augmentation. All images are resized to 256×256 pixels and the model is trained on random 224×224 crops and horizontal flips using stochastic gradient descent (SGD) with hand-tuned learning rates.

6.3.4 Vocabulary Expansion

When captioning images containing objects previously unseen by the captioning model, the text fragments used as constraints may contain words that are not actually present in the captioning model’s vocabulary. To tackle this issue, we leverage pre-trained word embeddings, specifically the 300 dimension GloVe [Pennington et al., 2014] embeddings trained on 42B tokens of external text corpora. These embeddings are introduced at both the word input and word output layers of the captioning model and fixed throughout training. Concretely, the i th column of the W_e input embedding matrix is initialised with the GloVe vector associated with vocabulary word i . This entails reducing the dimension of the original LRCN input embedding from 1K to 300. The model output is then:

$$\mathbf{v}_t = \tanh(W_v \mathbf{h}_t^2 + \mathbf{b}_v) \quad (6.5)$$

$$p(y_t | y_{t-1}, \dots, y_1, I) = \text{softmax}(W_e^T \mathbf{v}_t) \quad (6.6)$$

where \mathbf{v}_t represents the top LSTM output projected to 300 dimensions, W_e^T contains GloVe embeddings as row vectors, and $p(y_t | y_{t-1}, \dots, y_1, I)$ represents the normalised probability distribution over the predicted output word y_t at timestep t , given the previous output words and the image. The model is trained with the conventional softmax cross-entropy loss function, and learns to predict \mathbf{v}_t vectors that have a high dot-product similarity with the GloVe embedding of the correct output word.

Given these modifications—which could be applied to other similar captioning models—the process of expanding the model’s vocabulary at test time is straightforward. To introduce an additional vocabulary word, the GloVe embedding for the new word is simply concatenated with W_e as an additional column, increasing the dimension of both Π_t and p_t by one. In total there are 1.9M words in our selected GloVe embedding, which for practical purposes represents an open vocabulary. Since GloVe embeddings capture semantic and syntactic similarities [Pennington et al., 2014], intuitively the captioning model will generalise from similar words in order to understand how the new word can be used.

6.4 Experiments

To evaluate our approach, we conduct two experiments. In the first experiment we use a held-out version of the COCO dataset. Leveraging image tag predictions from an existing model [Hendricks et al., 2016] as constraints, we demonstrate state of the art performance for out-of-domain image captioning, while simultaneously improv-

ing the performance of the base model on in-domain data. Perhaps surprisingly, our results significantly outperform approaches that incorporate the same tag predictions into the learning algorithm [Hendricks et al., 2016; Venugopalan et al., 2017]. In the second experiment we attempt the extremely challenging task of using a model trained on COCO to caption the ImageNet classification dataset [Russakovsky et al., 2015] (which contains hundreds of unseen object classes). Human evaluations indicate that by leveraging ground truth image labels as constraints, the proportion of captions meeting or exceeding human quality doubles.

6.4.1 Dataset Pre-processing

Refer to Section 3.2 for a detailed overview of the COCO captions dataset [Lin et al., 2014b]. In our experiments in this chapter we follow standard practice and perform only minimal text pre-processing, converting all sentences to lower case and tokenizing on white space. It is common practice to filter vocabulary words that occur less than five times in the training set. However, since our model does not learn word embeddings, vocabulary filtering is not necessary. Avoiding filtering increases our vocabulary from around 8,800 words to 21,689, allowing the model to potentially extract a useful training signal even from rare words and spelling mistakes (which are generally close to the correctly spelled word in embedding space). In all experiments we use a beam size of 5, and we also enforce the constraint that a single word cannot be predicted twice in a row.

6.4.2 Out-of-Domain Image Captioning

To evaluate the ability of our approach to perform novel object image captioning, we replicate an existing experimental design [Hendricks et al., 2016] using COCO. Following this approach, all images with captions that mention one of eight selected objects (or their synonyms) are excluded from the image caption training set. This reduces the size of the caption training set from 82,783 images to 70,194 images. However, the complete caption training set is tokenized as a bag of words per image, and made available as image tag training data. As such, the selected objects are unseen in the image caption training data, but not the image tag training data. The excluded objects, selected by Hendricks et al. [2016] from the 80 main object categories in COCO, are: ‘bottle’, ‘bus’, ‘couch’, ‘microwave’, ‘pizza’, ‘racket’, ‘suitcase’ and ‘zebra’.

For validation and testing on this task, we use the same splits as in prior work [Hendricks et al., 2016; Venugopalan et al., 2017], with half of the original COCO validation set used for validation, and half for testing. We use the validation set to



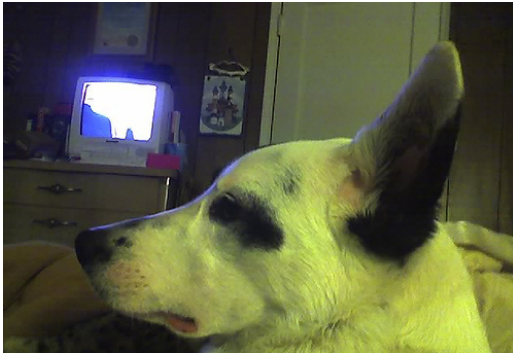
Base: A woman is playing tennis on a tennis court. **Tags:** tennis, player, ball, racket. **Base+T4:** A tennis player swinging a racket at a ball.



Base: A man standing next to a yellow train. **Tags:** bus, yellow, next, street. **Base+T4:** A man standing next to a yellow bus on the street.



Base: A close up of a cow on a dirt ground. **Tags:** zebra, zoo, enclosure, standing. **Base+T4:** A zebra standing in front of a zoo enclosure.



Base: A dog is sitting in front of a tv. **Tags:** dog, head, television, cat. **Base+T4:** A dog with a cat on its head watching television.



Base: A group of people playing a game of tennis. **Tags:** pink, tennis, crowd, ball. **Base+T4:** A crowd of people standing around a pink tennis ball.

Figure 6.3: Examples of captions generated on the COCO dataset using the base model (Base), and the base model constrained to include four predicted image tags (Base+T4). Words never seen in training captions are underlined. The bottom row contains some failure cases.

determine hyperparameters and for early-stopping, and report all results on the test set. For evaluation the test set is split into in-domain and out-of-domain subsets, with the out-of-domain designation given to any test image that contains a mention of an excluded object in at least one reference caption.

To evaluate generated caption quality, we use the SPICE metric (refer Chapter 4 and [Anderson et al., 2016]), which has been shown to correlate well with human judgement on the COCO dataset, as well as the METEOR [Denkowski and Lavie,

Model	CNN	Out-of-Domain Test Data				In-Domain Test Data		
		SPICE	METEOR	CIDEr	F1	SPICE	METEOR	CIDEr
DCC	VGG-16	13.4	21.0	59.1	39.8	15.9	23.0	77.2
NOC	VGG-16	-	21.4	-	49.1	-	-	-
Base	VGG-16	12.4	20.4	57.7	0	17.6	24.9	93.0
Base+T1	VGG-16	13.6	21.7	68.9	27.2	17.9	25.0	93.4
Base+T2	VGG-16	14.8	22.6	75.4	38.7	18.2	25.0	92.8
Base+T3	VGG-16	15.5	23.0	77.5	48.4	18.2	24.8	90.4
Base+T4	VGG-16	15.9	23.3	77.9	54.0	18.0	24.5	86.3
Base+T3*	VGG-16	18.7	27.1	119.6	54.5	22.0	29.4	135.5
Base All	VGG-16	17.8	25.2	93.8	59.4	17.4	24.5	91.7
Base	ResNet-50	12.6	20.5	56.8	0	18.2	24.9	93.2
Base+T1	ResNet-50	14.2	21.7	68.1	27.3	18.5	25.2	94.6
Base+T2	ResNet-50	15.3	22.7	74.7	38.5	18.7	25.3	94.1
Base+T3	ResNet-50	16.0	23.3	77.8	48.2	18.7	25.2	92.3
Base+T4	ResNet-50	16.4	23.6	77.6	53.3	18.4	24.9	88.0
Base+T3*	ResNet-50	19.2	27.3	117.9	54.5	22.3	29.4	133.7
Base All	ResNet-50	18.6	26.0	96.9	60.0	18.0	25.0	93.8

Table 6.1: Evaluation of captions generated using constrained beam search with 1–4 predicted image tags used as constraints (Base+T1–4). Our approach significantly outperforms both the DCC [Hendricks et al., 2016] and NOC [Venugopalan et al., 2017] models, despite reusing the image tag predictions of the DCC model. Importantly, performance on in-domain data is not degraded but can also improve.

2014] and CIDEr [Vedantam et al., 2015] metrics. For consistency with previously reported results, scores on out-of-domain test data are macro-averaged across the eight excluded object classes. To improve the comparability of CIDEr scores, the inverse document frequency statistics used by this metric are determined across the entire test set, rather than within subsets. On out-of-domain test data, we also report the F1 metric for mentions of excluded objects. To calculate the F1 metric, the model is considered to have predicted condition positive if the generated caption contains at least one mention of the excluded object, and negative otherwise. The ground truth is considered to be positive for an image if the excluded object in question is mentioned in any of the reference captions, and negative otherwise.

As illustrated in Table 6.1, on the out-of-domain test data, our base model trained only with image captions (Base) receives an F1 score of 0, as it is incapable of mentioned objects that do not appear in the training captions. In terms of SPICE, METEOR and CIDEr scores, our base model performs slightly worse than the DCC model on out-of-domain data, but significantly better on in-domain data. This may suggest that the DCC model achieves improvements in out-of-domain performance

Model	bottle	bus	couch	microwave	pizza	racket	suitcase	zebra	Avg
DCC	4.6	29.8	45.9	28.1	64.6	52.2	13.2	79.9	39.8
NOC	17.8	68.8	25.6	24.7	69.3	68.1	39.9	89.0	49.1
Base+T4	16.3	67.8	48.2	29.7	77.2	57.1	49.9	85.7	54.0

Table 6.2: F1 scores for mentions of objects not seen during caption training. Our approach (Base+T4) reuses the top 4 image tag predictions from the DCC [Hendricks et al., 2016] model but generates higher F1 scores by interpreting tag predictions as constraints. All results based on use of the VGG-16 CNN.

at the expense of in-domain scores (in-domain scores for the NOC model were not available at the time of submission).

Results marked with ‘+’ in Table 6.1 indicate that our base model has been decoded with constraints in the form of predicted image tags. However, for the fairest comparison, and because re-using existing image taggers at test time is one of the motivations for this work, we did not train an image tagger from scratch. Instead, in results T1–4 we use the top 1–4 tag predictions respectively from the VGG-16 CNN-based image tagger used in the DCC model. This model was trained by Hendricks et al. [2016] to predict 471 COCO visual concepts including adjectives, verbs and nouns. Examples of generated captions, including failure cases, are presented in Figure 6.3.

As indicated in Table 6.1, using similar model capacity, the constrained beam search approach with predicted tags significantly outperforms prior work in terms SPICE, METEOR and CIDEr scores, across both out-of-domain and in-domain test data, utilising varying numbers of tag predictions. Overall these results suggest that, perhaps surprisingly, it may be better to incorporate image tags into captioning models during decoding rather than during training. It also appears that, while introducing image tags improves performance on both out-of-domain and in-domain evaluations, it is beneficial to introduce more tag constraints when the test data is likely to contain previously unseen objects. This reflects the trading-off of influence between the image tags and the captioning model. For example, we noted that when using two tag constraints, 36% of generated captions were identical to the base model, but when using four tags this proportion dropped to only 3%.

To establish performance upper bounds, we train the base model on the complete COCO training set (Base All). We also evaluate captions generated using our approach combined with an ‘oracle’ image tagger consisting of the top 3 ground-truth image tags (T3*). These were determined by selecting the 3 most frequently mentioned words in the reference captions for each test image (after eliminating stop words). The very high scores recorded for this approach may motivate the use of

	Better	Equally Good	Equally Poor	Worse
Base v. Human	0.05	0.06	0.04	0.86
Base+Syn v. Human	0.12	0.10	0.05	0.73
Base+Syn v. Base	0.39	0.06	0.42	0.13

Table 6.3: Human evaluations comparing ImageNet captions. Our approach leveraging ground-truth synset labels (Base+Syn) improves significantly over the base model (Base) in both direct comparison and in comparison to human-generated captions.

more powerful image taggers in future work. Finally, replacing VGG-16 with the more powerful ResNet-50 [He et al., 2016a] CNN leads to modest improvements as indicated in the lower half of Table 6.1.

Evaluating F1 scores for object mentions (see Table 6.2), we note that while our approach outperforms prior work when four image tags are used, a significant increase in this score should not be expected as the underlying image tagger is the same.

6.4.3 Captioning ImageNet

Consistent with our observation that many image collections contain useful annotations, and that we should seek to use this information, in this section we caption a 5K image subset of the ImageNet [Russakovsky et al., 2015] ILSVRC 2012 classification dataset for assessment. The dataset contains 1.2M images classified into 1K object categories, from which we randomly select five images from each category.

For this task we use the ResNet-50 [He et al., 2016a] CNN, and train the base model on a combined training set containing 155k images comprised of the COCO [Chen et al., 2015] training and validation datasets, and the full Flickr 30k [Young et al., 2014] captions dataset. We use constrained beam search and vocabulary expansion to ensure that each generated caption includes a phrase from the WordNet [Fellbaum, 1998] synset representing the ground-truth image category. For synsets that contain multiple entries, we run constrained beam search separately for each phrase and select the predicted caption with the highest log probability overall.

Note that even with the use of ground-truth object labels, the ImageNet captioning task remains extremely challenging as ImageNet contains a wide variety of classes, many of which are not represented in the available image-caption training datasets. Nevertheless, the injection of the ground-truth label frequently improves the overall structure of the caption over the base model in multiple ways. Examples of generated captions, including failure cases, are presented in Figure 6.4.

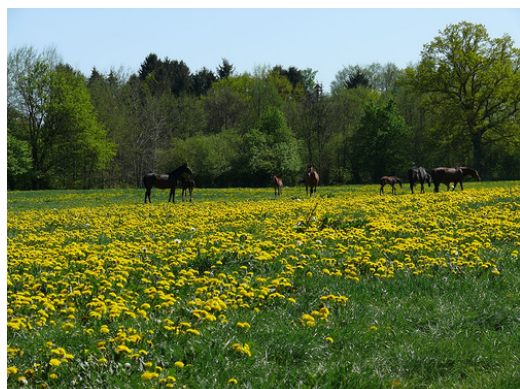
As the ImageNet dataset contains no existing caption annotations, following the



Base: A close up of a pizza on the ground. **Synset:** rock crab. **Base+Synset:** A large rock crab sitting on top of a rock.



Base: A close up shot of an orange. **Synset:** pool table, billiard table, snooker table. **Base+Synset:** A close up of an orange ball on a billiard table.



Base: A herd or horses standing on a lush green field. **Synset:** rapeseed. **Base+Synset:** A group of horses grazing in a field of rapeseed.



Base: A black bird is standing in the grass. **Synset:** oystercatcher, oyster catcher. **Base+Synset:** A black oystercatcher with a red beak standing in the grass.



Base: A man and a woman standing next to each other. **Synset:** colobus, colobus monkey. **Base+Synset:** Two colobus standing next to each other near a fence.



Base: A bird standing on top of a grass covered field. **Synset:** cricket. **Base+Synset:** A bird standing on top of a cricket field.

Figure 6.4: Examples of ImageNet captions generated by the base model (Base), and by the base model constrained to include the ground-truth synset (Base+Synset). Words never seen in the COCO / Flickr 30k caption training set are underlined. The bottom row contains some failure cases.

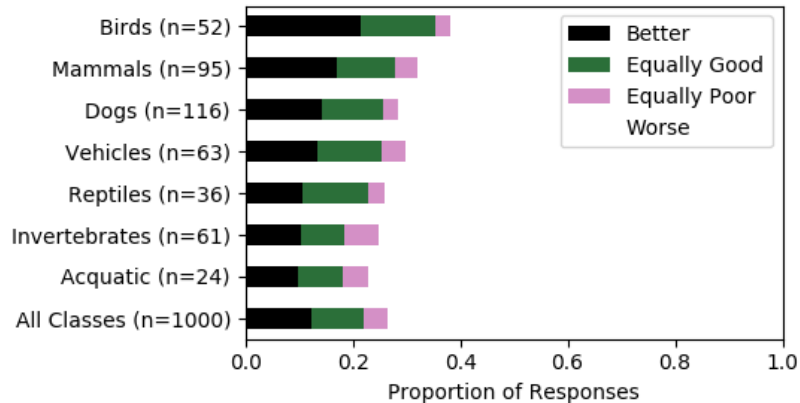


Figure 6.5: Human evaluations of generated (Base+Syn) ImageNet captions versus human captions, by super-category.

human-evaluation protocol established for the COCO 2015 Captioning Challenge [Chen et al., 2015], we used Amazon Mechanical Turk (AMT) to collect a human-generated caption for each sample image. For each of the 5K samples images, three human evaluators were then asked to compare the caption generated using our approach with the human-generated caption (Base+Syn v. Human). Using a smaller sample of 1K images, we also collected evaluations comparing our approach to the base model (Base+Syn v. Base), and comparing the base model with human-generated captions (Base v. Human). We used only US-based AMT workers, screened according to their performance on previous tasks. For both tasks, the user interface and question phrasing was identical to the COCO collection process. The results of these evaluations are summarised in Table 6.3.

Overall, Base+Syn captions were judged to be equally good or better than human-generated captions in 22% of pairwise evaluations (12% ‘better’, 10% ‘equally good’), and equally poor or worse than human-generated captions in the remaining 78% of evaluations. Although still a long way from human performance, this is a significant improvement over the base model with only 11% of captions judged to be equally good or better than human. For context, using the identical evaluation protocol, the top scoring model in the COCO Captioning Challenge (evaluating on in-domain data) received 11% ‘better’, and 17% ‘equally good’ evaluations.

To better understand performance across synsets, in Figure 6.5 we cluster some class labels into super-categories using the WordNet hierarchy, noting particularly strong performances in super-categories that have some representation in the caption training data — such as birds, mammals and dogs. These promising results suggest that fine-grained object labels can be successfully integrated with a general purpose captioning model using our approach.

6.5 Chapter Summary

To approach human performance on vision and language tasks, we must grapple with a very long tail of rare visual concepts. Motivated by this problem, in this chapter we introduce *constrained beam search*, a novel approximate search algorithm capable of enforcing any constraints over generated output sequences that can be expressed in a finite-state machine. Applying this approach to out-of-domain image captioning, we demonstrate that the outputs of an image tagger and an image captioning model can be successfully combined at test time, allowing an image captioning model to scale to many more visual concepts. Our approach achieves state of the art results on a held-out COCO dataset. Furthermore, we show that we can significantly improve the quality of generated ImageNet captions by using the ground-truth labels. Perhaps surprisingly, our approach—which does not rely on any joint training of the image tagger and the captioning model—outperformed existing methods that incorporate tag predictions into the learning algorithm. Given this result we view the incorporation of constrained beam search into the learning algorithm—perhaps using an expectation-maximisation (EM) style algorithm—as an exciting direction for future research.

Vision and Language Navigation in Real Environments

In Chapters 4–6 we focused on the challenges of image captioning [Chen et al., 2015] and visual question answering (VQA) [Goyal et al., 2017], addressing evaluation, attention modelling, and generalisation to more diverse input images, respectively. These tasks are ideal for encouraging and quantifying progress in vision and language understanding. However, these and other datasets based on static images have a serious limitation as they do not allow the model (agent) to move or control the camera, or take any other actions in the environment. This neglects a crucial aspect of many problems, namely, the embodiment of the agent. In this chapter, we address this limitation by proposing a new task and dataset to enable and encourage the application of vision and language methods to problems involving embodied agents.

7.1 Vision and Language Navigation

The idea that we might be able to give general, verbal instructions to a robot and have at least a reasonable probability that it will carry out the required task is one of the long-held goals of robotics, and artificial intelligence (AI). Despite significant progress, there are a number of major technical challenges that need to be overcome before robots will be able to perform general tasks in the real world. One of the primary requirements will be new techniques for linking natural language to vision and action in *unstructured, previously unseen environments*. It is the navigation version of this challenge that we refer to as Vision and Language Navigation (VLN).

Although interpreting natural-language navigation instructions has received significant attention previously [Chaplot et al., 2018; Chen and Mooney, 2011; Guadarrama et al., 2013; Mei et al., 2016; Misra et al., 2017; Tellex et al., 2011], it is the recent success of recurrent neural network methods for the joint interpretation of

images and natural language that motivates the VLN task, and the associated Room-to-Room (R2R) dataset described below. The dataset particularly has been designed to simplify the application of vision and language methods to what might otherwise seem a distant problem.

Previous approaches to natural language command of robots have often neglected the visual information processing aspect of the problem. Using rendered, rather than real images [Beattie et al., 2016; Kempka et al., 2016; Zhu et al., 2017], for example, constrains the set of visible objects to the set of hand-crafted models available to the renderer. This turns the robot’s challenging open-set problem of relating real language to real imagery into a far simpler closed-set classification problem. The natural extension of this process is that adopted in works where the images are replaced by a set of labels [Chen and Mooney, 2011; Tellex et al., 2011]. Limiting the variation in the imagery inevitably limits the variation in the navigation instructions also. What distinguishes the VLN challenge is that the agent is required to interpret a previously *unseen* natural-language navigation command in light of images generated by a previously *unseen* real environment. The task thus more closely models the distinctly open-set nature of the underlying problem.

To enable the reproducible evaluation of VLN methods, we present the Matterport3D Simulator. The simulator is a large-scale interactive reinforcement learning (RL) environment constructed from the Matterport3D dataset [Chang et al., 2017] which contains 10,800 densely-sampled panoramic RGB-D images of 90 real-world building-scale indoor environments. Compared to synthetic RL environments [Beattie et al., 2016; Kempka et al., 2016; Zhu et al., 2017], the use of real-world image data preserves visual and linguistic richness, maximising the potential for trained agents to be transferred to real-world applications.

Based on the Matterport3D environments, we collect the Room-to-Room (R2R) dataset containing 21,567 open-vocabulary, crowd-sourced navigation instructions with an average length of 29 words. Each instruction describes a trajectory traversing typically multiple rooms. As illustrated in Figure 7.1, the associated task requires an agent to follow natural-language instructions to navigate to a goal location in a previously unseen building. We investigate the difficulty of this task, and particularly the difficulty of operating in unseen environments, using several baselines and a sequence-to-sequence model based on methods successfully applied to other vision and language tasks [Antol et al., 2015; Chen et al., 2015; Goyal et al., 2017].



Instruction: Head upstairs and walk past the piano through an archway directly in front. Turn right when the hallway ends at the pictures and table. Wait by the moose antlers hanging on the wall.

Figure 7.1: The Room-to-Room (R2R) navigation task. We focus on executing natural language navigation instructions in previously unseen real-world buildings. The agent’s camera can be rotated freely. Blue discs indicate nearby (discretized) navigation options.

7.2 Related Work

Navigation and language Natural language command of robots in unstructured environments has been a research goal for several decades [Winograd, 1971]. However, many existing approaches abstract away the problem of visual perception to some degree. This is typically achieved either by assuming that the set of all navigation goals, or objects to be acted upon, has been enumerated, and that each will be identified by label [Chen and Mooney, 2011; Tellex et al., 2011], or by operating in visually restricted environments requiring limited perception [Chaplot et al., 2018; Guadarrama et al., 2013; Huang et al., 2010; Kollar et al., 2010; MacMahon et al., 2006; Mei et al., 2016; Vogel and Jurafsky, 2010]. Our work contributes for the first time a navigation benchmark dataset that is both linguistically and visually rich, moving closer to real scenarios while still enabling reproducible evaluations.

Vision and language The development of new benchmark datasets for image captioning [Chen et al., 2015], visual question answering (VQA) [Antol et al., 2015; Goyal et al., 2017] and visual dialog [Das et al., 2017] has spurred considerable progress in

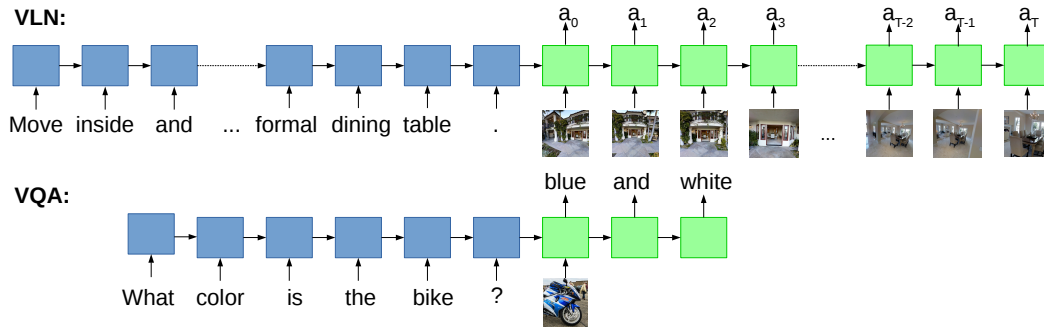


Figure 7.2: Differences between Vision and Language Navigation (VLN) and Visual Question Answering (VQA). Both tasks can be formulated as visually grounded sequence-to-sequence transcoding problems. However, VLN sequences are much longer and, uniquely among vision and language benchmark tasks using real images, the model outputs actions (a_0, a_1, \dots, a_T) that manipulate the camera viewpoint.

vision and language understanding, enabling models to be trained end-to-end on raw pixel data from large datasets of natural images. However, although many tasks combining visual and linguistic reasoning have been motivated by their potential robotic applications [Antol et al., 2015; Das et al., 2017; Kazemzadeh et al., 2014; Mao et al., 2016; Tapaswi et al., 2016], none of these tasks allow an agent to move or control the camera. As illustrated in Figure 7.2, our proposed R2R benchmark addresses this limitation, which also motivates several concurrent works on embodied question answering [Das et al., 2018; Gordon et al., 2018].

Navigation based simulators Our simulator is related to existing 3D RL environments based on game engines, such as ViZDoom [Kempka et al., 2016], DeepMind Lab [Beattie et al., 2016] and AI2-THOR [Kolve et al., 2017], as well as a number of newer environments developed concurrently including HoME [Brodeur et al., 2017], House3D [Wu et al., 2018], MINOS [Savva et al., 2017], CHALET [Yan et al., 2018] and Gibson Env [Zamir et al., 2018b]. The main advantage of our framework over synthetic environments [Kolve et al., 2017; Brodeur et al., 2017; Wu et al., 2018; Yan et al., 2018] is that all pixel observations come from natural images of real scenes, ensuring that almost every coffee mug, pot-plant and wallpaper texture is unique. This visual diversity and richness is hard to replicate using a limited set of 3D assets and textures. Compared to MINOS [Savva et al., 2017], which is also based on Matterport data [Chang et al., 2017], we render from panoramic images rather than textured meshes. Since the meshes have missing geometry—particularly for windows and mirrors—our approach improves visual realism but limits navigation

to discrete locations (refer Section 7.3.2 for details). Our approach is similar to the (much smaller) Active Vision Dataset [Ammirato et al., 2017].

RL in navigation A number of recent papers use reinforcement learning (RL) to train navigational agents [Kulkarni et al., 2016; Tai and Liu, 2016; Tessler et al., 2017; Zhu et al., 2017; Gupta et al., 2017], although these works do not address language instruction. The use of RL for language-based navigation has been studied in Chaplot et al. [2018] and Misra et al. [2017], however, the settings are visually and linguistically less complex. For example, Chaplot et al. [2018] develop an RL model to execute template-based instructions in Doom environments [Kempka et al., 2016]. Misra et al. [2017] study complex language instructions in a fully-observable blocks world. By releasing our simulator and dataset, we hope to encourage further research in more realistic partially-observable settings.

7.3 Matterport3D Simulator

In this section we introduce the Matterport3D Simulator, a new large-scale visual reinforcement learning (RL) simulation environment for the research and development of intelligent agents based on the Matterport3D dataset [Chang et al., 2017]. The Room-to-Room (R2R) navigation dataset is discussed in Section 7.4.

7.3.1 Matterport3D Dataset

Most RGB-D datasets are derived from video sequences; e.g. NYUv2 [Nathan Silberman and Fergus, 2012], SUN RGB-D [Song et al., 2015] and ScanNet [Dai et al., 2017a]. These datasets typically offer only one or two paths through a scene, making them inadequate for simulating robot motion. In contrast to these datasets, the recently released Matterport3D dataset [Chang et al., 2017] contains a comprehensive set of panoramic views. To the best of our knowledge it is also the largest currently available RGB-D research dataset.

In detail, the Matterport3D dataset consists of 10,800 panoramic views constructed from 194,400 RGB-D images of 90 building-scale scenes. On average, panoramic viewpoints are distributed throughout the entire walkable floor plan of each scene at an average separation of 2.25m. Each panoramic view is comprised of 18 RGB-D images captured from a single 3D position at the approximate height of a standing person. Each image is annotated with an accurate 6 DoF camera pose, and collectively the images capture the entire sphere except the poles. The dataset also includes



Figure 7.3: A snapshot of the visual diversity in the Matterport3D dataset [Chang et al., 2017], illustrating one randomly selected panoramic viewpoint per scene.

globally-aligned, textured 3D meshes annotated with class and instance segmentations of regions (rooms) and objects.

In terms of visual diversity, as illustrated in Figure 7.3, the selected Matterport scenes encompass a range of buildings including houses, apartments, hotels, offices and churches of varying size and complexity. These buildings contain enormous visual diversity, posing real challenges to computer vision. Many of the scenes in the dataset can be viewed in the Matterport 3D spaces gallery¹.

7.3.2 Simulator

7.3.2.1 Observations

To construct the simulator, we allow an embodied agent to virtually ‘move’ throughout a scene by adopting poses coinciding with panoramic viewpoints. Agent poses are defined in terms of 3D position $v \in V$, heading $\psi \in [0, 2\pi)$, and camera elevation $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, where V is the set of 3D points associated with panoramic viewpoints in the scene. At each step t , the simulator outputs an RGB image observation o_t corresponding to the agent’s first person camera view. Images are generated from perspective projections of precomputed cube-mapped images at each viewpoint. Future extensions to the simulator will also support depth image observations (RGB-D), and additional instrumentation in the form of rendered object class and object instance segmentations (based on the underlying Matterport3D mesh annotations).

7.3.2.2 Action Space

The main challenge in implementing the simulator is determining the state-dependent action space. Naturally, we wish to prevent agents from teleporting through walls and floors, or traversing other non-navigable regions of space. Therefore, at each step t the simulator also outputs a set of next step reachable viewpoints $W_{t+1} \subseteq V$. Agents interact with the simulator by selecting a new viewpoint $v_{t+1} \in W_{t+1}$, and nominating camera heading ($\Delta\psi_{t+1}$) and elevation ($\Delta\theta_{t+1}$) adjustments. Actions are deterministic.

To determine W_{t+1} , for each scene the simulator includes a weighted, undirected graph over panoramic viewpoints, $G = (V, E)$, such that the presence of an edge signifies a robot-navigable transition between two viewpoints, and the weight of that edge reflects the straight-line distance between them. To construct the graphs, we ray-traced between viewpoints in the Matterport3D scene meshes to detect intervening obstacles. To ensure that motion remains localised, we then removed edges longer

¹<https://matterport.com/gallery/>

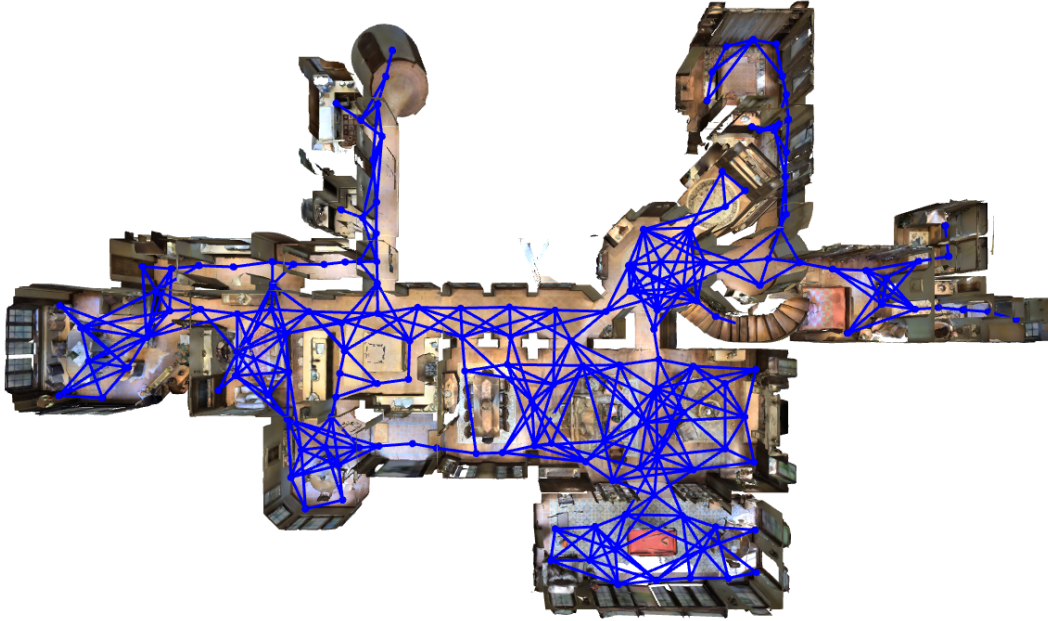


Figure 7.4: An example navigation graph for a partial floor of one building-scale scene in the Matterport3D Simulator. Navigable paths between panoramic view-points are illustrated in blue. Stairs can also be navigated to move between floors.

than 5m. Finally, we manually verified each navigation graph to correct for missing obstacles not captured in the meshes (such as windows and mirrors).

Given navigation graph G , the set of next-step reachable viewpoints is given by:

$$W_{t+1} = \{v_t\} \cup \{v_i \in V \mid (v_t, v_i) \in E \wedge v_i \in P_t\} \quad (7.1)$$

where v_t is the current viewpoint, and P_t is the region of space enclosed by the left and right extents of the camera view frustum at step t . In effect, the agent is permitted to follow any edges in the navigation graph, provided that the destination is within the current field of view, or visible by glancing up or down². Alternatively, the agent always has the choice to remain at the same viewpoint and simply move the camera.

Figure 7.4 illustrates a partial example of a typical navigation graph. On average each graph contains 117 viewpoints, with an average vertex degree of 4.1. This compares favourably with grid-world navigation graphs which, due to walls and obstacles, must have an average degree of less than 4. As such, although agent motion is discretised, this does not constitute a significant limitation in the context of most high-level tasks. Even with a real robot it may not be practical or necessary

²This avoids forcing the agent to look at the floor every time it takes a small step.

to continuously re-plan higher-level objectives with every new RGB-D camera view. Indeed, even agents operating in 3D simulators that notionally support continuous motion typically use discretised action spaces in practice [Zhu et al., 2017; Das et al., 2018; Gordon et al., 2018; Savva et al., 2017].

The simulator does not define or place restrictions on the agent’s goal, reward function, or any additional context (such as natural language navigation instructions). These aspects of the RL environment are task and dataset dependent, for example as described in Section 7.4.

7.3.2.3 Implementation Details

The Matterport3D Simulator is written in C++ using OpenGL. In addition to the C++ API, Python bindings are also provided, allowing the simulator to be easily used with deep learning frameworks such as Caffe [Jia et al., 2014] and TensorFlow [Abadi et al., 2016], or within RL platforms such as ParlAI [Miller et al., 2017] and OpenAI Gym [Brockman et al., 2016]. Various configuration options are offered for parameters such as image resolution and field of view. Separate to the simulator, we have also developed a WebGL browser-based visualisation library for collecting text annotations of navigation trajectories using Amazon Mechanical Turk, which we will make available to other researchers.

7.3.2.4 Biases

We are reluctant to introduce a new dataset (or simulator, in this case) without at least some attempt to address its limitations and biases [Torralba and Efros, 2011]. In the Matterport3D dataset we have observed several selection biases. First, the majority of captured living spaces are scrupulously clean and tidy, and often luxurious. Second, the dataset contains very few people and animals, which are a mainstay of many other vision and language datasets [Chen et al., 2015; Antol et al., 2015]. Finally, we observe some capture bias as selected viewpoints generally offer commanding views of the environment (and are therefore not necessarily in the positions in which a robot might find itself). Alleviating these limitations to some extent, the simulator can be extended by collecting additional building scans. Refer to Stanford 2D-3D-S [Armeni et al., 2017] for a recent example of an academic dataset collected with a Matterport camera.

7.4 Room-to-Room (R2R) Navigation

We now describe the Room-to-Room (R2R) task and dataset, including an outline of the data collection process and analysis of the navigation instructions gathered.

7.4.1 Task

As illustrated in Figure 7.1, the R2R task requires an embodied agent to follow natural language instructions to navigate from a starting pose to a goal location in the Matterport3D Simulator. Formally, at the beginning of each episode the agent is given as input a natural language instruction $\bar{x} = (x_1, x_2, \dots, x_L)$, where L is the length of the instruction and x_i is a single word token. The agent observes an initial RGB image o_0 , determined by the agent’s initial pose comprising a tuple of 3D position, heading and elevation $s_0 = (v_0, \psi_0, \theta_0)$. The agent must execute a sequence of actions $(s_0, a_0, s_1, a_1, \dots, s_T, a_T)$, with each action a_t leading to a new pose $s_{t+1} = (v_{t+1}, \psi_{t+1}, \theta_{t+1})$, and generating a new image observation o_{t+1} . The episode ends when the agent selects the special `stop` action, which is augmented to the simulator action space defined in Section 7.3.2.2. The task is successfully completed if the action sequence delivers the agent close to an intended goal location v^* (refer to Section 7.4.4 for evaluation details).

7.4.2 Data Collection

To generate navigation data, we use the Matterport3D region annotations to sample start pose s_0 and goal location v^* pairs that are (predominantly) in different rooms. For each pair, we find the shortest path $v_0 : v^*$ in the relevant weighted, undirected navigation graph G , discarding paths that are shorter than 5m, and paths that contain less than four or more than six edges. In total we sample 7,189 paths capturing most of the visual diversity in the dataset. The average path length is 10m, as illustrated in Figure 7.6.

For each path, we collect three associated navigation instructions using Amazon Mechanical Turk (AMT). To this end, we provide workers with an interactive 3D WebGL environment depicting the path from the start location to the goal location using coloured markers. Workers can interact with the trajectory as a ‘fly-through’, or pan and tilt the camera at any viewpoint along the path for additional context. We then ask workers to ‘write directions so that a smart robot can find the goal location after starting from the same start location’. Workers are further instructed that it is not necessary to follow exactly the path indicated, merely to reach the goal. A video demonstration is also provided.



Pass the pool and go indoors using the double glass doors. Pass the large table with chairs and turn left and wait by the wine bottles that have grapes by them.

Walk straight through the room and exit out the door on the left. Keep going past the large table and turn left. Walk down the hallway and stop when you reach the 2 entry ways. One in front of you and one to your right. The bar area is to your left.

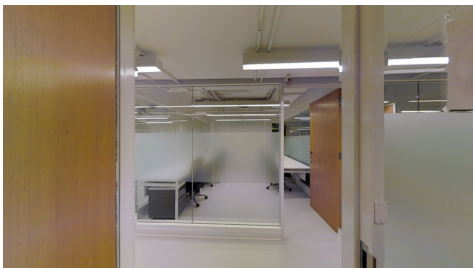
Enter house through double doors, continue straight across dining room, turn left into bar and stop on the circle on the ground.



Standing in front of the family picture, turn left and walk straight through the bathroom past the tub and mirrors. Go through the doorway and stop when the door to the bathroom is on your right and the door to the closet is to your left.

Walk with the family photo on your right. Continue straight into the bathroom. Walk past the bathtub. Stop in the hall between the bathroom and toilet doorways.

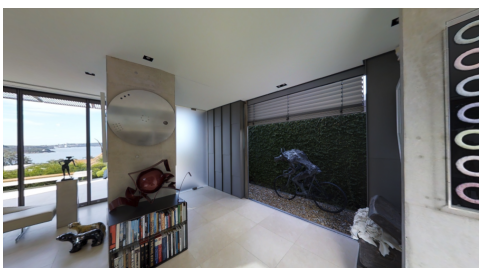
Walk straight passed bathtub and stop with closet on the left and toilet on the right.



Exit the office then turn left and then turn left in the hallway and head down the hallway until you get to a door on your left and go into office 359 then stop.

Go out of the room and take a left. Go into the first room on your left.

Leave the office and take a left. Take the next left at the hallway. Walk down the hall and enter the first office on the left. Stop next to the door to office 359.



Go up the stairs and turn right. Go past the bathroom and stop next to the bed.

Walk all the way up the stairs, and immediately turn right. Pass the bathroom on the left, and enter the bedroom that is right there, and stop there.

Walk up the stairs turn right at the top and walk through the doorway continue straight and stop inside the bedroom.

Figure 7.5: Randomly selected examples of R2R navigation instructions (three per trajectory) showing the view from the starting pose.

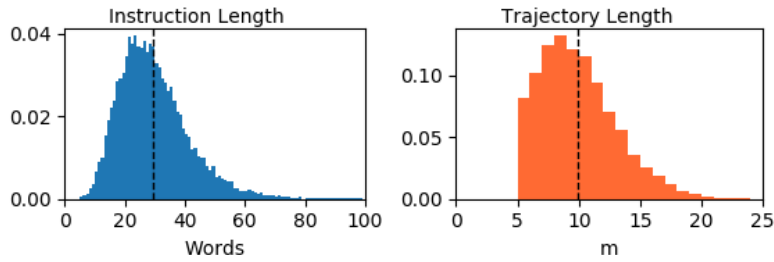


Figure 7.6: Distribution of instruction length and navigation trajectory length in the R2R dataset.

The full collection interface (illustrated in Figure 7.7) was the result of several rounds of experimentation. We used only US-based AMT workers, screened according to their performance on previous tasks. Over 400 workers participated in the data collection, contributing around 1,600 hours of annotation time.

7.4.3 R2R Dataset Analysis

In total, we collected 21,567 navigation instructions with an average length of 29 words. This is considerably longer than visual question answering datasets where most questions range from four to ten words [Antol et al., 2015]. However, given the focused nature of the task, the instruction vocabulary is relatively constrained, consisting of around 3.1k words (approximately 1.2k with five or more mentions). As illustrated by the examples included in Figure 7.5, the level of abstraction in instructions varies widely. This likely reflects differences in people’s mental models of the way a ‘smart robot’ works [Norman, 2002], making the handling of these differences an important aspect of the task. The distribution of navigation instructions based on their first words is depicted in Figure 7.8. Although we use the R2R dataset in conjunction with the Matterport3D Simulator, we see no technical reason why this dataset couldn’t also be used with other simulators based on the Matterport dataset [Chang et al., 2017].

7.4.4 Evaluation Protocol

One of the strengths of the R2R task is that, in contrast to many other vision and language tasks such as image captioning, success is clearly measurable. We define *navigation error* as the shortest path distance in the navigation graph G between the agent’s final position v_T (i.e., disregarding heading and elevation) and the goal location v^* . We consider an episode to be a *success* if the navigation error is less than 3m. This threshold allows for a margin of error of approximately one viewpoint, yet

Instructions: Give A Smart Robot Directions (Click to collapse)

You will see a series of panoramic photos taken while moving from a **start location** to a **goal location** in a building. **Your task is to write directions so that a smart robot can find the goal location after starting from the same start location.** The robot understands language and recognizes objects about as well as a typical person. However, you should assume that the robot is visiting this building for the first time.

For your reference, the path to the goal is indicated by color-coded markers (**green** for start, **red** for goal, and **blue** for intermediate markers).

- You won't see the **green** start marker at the beginning - because it's under your feet.
- You may not see the **red** goal marker until you move (often the goal is in the next room).
- These markers are not visible to the robot**, and should not be mentioned in your directions.

Good directions will ensure that the robot arrives **within a few metres** of the red goal marker. Therefore, we suggest:

- NEW! Spelling and punctuation is important.** Please use full sentences with punctuation (.,) and correct spelling.
- Focus on the goal, not the path.** It's not necessary for the robot to follow the exact path indicated by the markers.
- Try to mention objects or landmarks.** This is clearer than saying 'turn slight left' or 'go forward'.

Mouse Controls:

- Left-click and drag the panoramic image** to look around.
- Right-click on a color-coded marker** to move to that position.
- Press the 'Play / Replay' button** at any time to watch a 15-20 second animated fly-through from the start to the goal.

Before you start, [please watch this short training video](#). It contains examples that will help you complete these tasks efficiently.

Note: This task is not suitable for devices with small screens or touch screen devices. Recommended browsers are Chrome, Firefox and Safari (not Internet Explorer).

These tasks relate to academic research conducted by Peter Anderson through the [Australian Centre for Robotic Vision](#), Brisbane, Australia. We estimate that on average each HIT to take around 1-1.5 minutes to complete. Please send your queries and feedback to bringmeaspoon@gmail.com. We will be continually releasing more HITs for this task.



Left-click and drag the panoramic image to start. Instructions have been updated from the first batch (please re-read).

Play / Replay

Write your Directions here (with correct spelling and punctuation):

Submit

Figure 7.7: AMT data collection interface for the R2R navigation dataset. Here, blue markers can be seen indicating the trajectory to the goal location. However, in many cases the worker must first look around (pan and tilt) to find the markers. Clicking on a marker moves the camera to that location. Workers can also watch a 'fly-through' of the complete trajectory by clicking the Play / Replay button.

it is comfortably below the minimum starting error of 5m. We do not evaluate the agent’s entire trajectory as many instructions do not specify the path that should be taken.

Central to our evaluation is the requirement for the agent to choose to end the episode when the goal location is identified. We consider stopping to be a fundamental aspect of completing the task, demonstrating understanding, but also freeing the agent to potentially undertake further tasks at the goal. However, we acknowledge that this requirement contrasts with recent works in vision-only navigation that do not train the agent to stop [Zhu et al., 2017; Mirowski et al., 2017]. To disentangle the problem of recognising the goal location, we also report success for each agent under an *oracle* stopping rule, i.e. if the agent stopped at the closest point to the goal on its trajectory. Misra et al. [2017] also use this evaluation.

Dataset Splits We follow broadly the same train/val/test split strategy as the Matterport3D dataset [Chang et al., 2017]. The test set consists of 18 scenes, and 4,173 instructions. We reserve an additional 11 scenes and 2,349 instructions for validating in unseen environments (val unseen). The remaining 61 scenes are pooled together, with instructions split 14,025 train / 1,020 val seen. Following best practice, goal locations for the test set will not be released. Instead, we will provide an evaluation server where agent trajectories may be uploaded for scoring.

7.5 Vision and Language Navigation Agents

In this section, we describe a sequence-to-sequence neural network agent and several other baselines that we use to explore the difficulty of the R2R navigation task.

7.5.1 Sequence-to-Sequence Model

We model the agent with a recurrent neural network policy using an LSTM-based [Hochreiter and Schmidhuber, 1997] sequence-to-sequence architecture with an attention mechanism [Bahdanau et al., 2015b]. Recall that the agent begins with a natural language instruction $\bar{x} = (x_1, x_2, \dots, x_L)$, and an initial image observation o_0 . The encoder computes a representation of \bar{x} . At each step t , the decoder observes representations of the current image o_t and the previous action a_{t-1} as input, applies an attention mechanism to the hidden states of the language encoder, and predicts a distribution over the next action a_t . Using this approach, the decoder maintains an internal memory of the agent’s entire preceding history, which is essential for

navigating in a partially observable environment [Wierstra et al., 2007]. We discuss further details in the following sections.

Language instruction encoding Each word x_i in the language instruction is presented sequentially to the encoder LSTM as an embedding vector. We denote the output of the encoder at step i as h_i , such that $h_i = \text{LSTM}_{enc}(x_i, h_{i-1})$. We denote $\bar{h} = \{h_1, h_2, \dots, h_L\}$ as the encoder context, which will be used in the attention mechanism. As with Sutskever et al. [2014], we found it valuable to reverse the order of words in the input language instruction.

Model action space The simulator action space is state-dependent (refer Section 7.3.2.2), allowing agents to make fine-grained choices between different forward trajectories that are presented. However, in this initial work we simplify our model action space to 6 actions corresponding to left, right, up, down, forward and stop. The forward action is defined to always move to the reachable viewpoint that is closest to the centre of the agent’s visual field. The left, right, up and down actions are defined to move the camera by 30 degrees.

Image and action embedding For each image observation o_t , we use a ResNet-152 [He et al., 2016a] CNN pretrained on ImageNet [Russakovsky et al., 2015] to extract a mean-pooled feature vector. Analogously to the embedding of instruction words, an embedding is learned for each action. The encoded image and previous action features are then concatenated together to form a single vector q_t . The decoder LSTM operates as $h'_t = \text{LSTM}_{dec}(q_t, h'_{t-1})$.

Action prediction with attention mechanism To predict a distribution over actions at step t , we first use an attention mechanism to identify the most relevant parts of the navigation instruction. This is achieved by using the global, general alignment function described by Luong et al. [2014] to compute an instruction context $c_t = f(h'_t, \bar{h})$. When then compute an attentional hidden state $\tilde{h}_t = \tanh(W_c[c_t; h'_t])$, and calculate the predictive distribution over the next action as $a_t = \text{softmax}(\tilde{h}_t)$. Although visual attention has also proved highly beneficial in vision and language problems [Yang et al., 2016a; Lu et al., 2016; Anderson et al., 2018a], we leave an investigation of visual attention in Vision and Language Navigation to future work.

7.5.2 Training

We investigate two training regimes, ‘teacher-forcing’ and ‘student-forcing’. In both cases, we use cross entropy loss at each step to maximize the likelihood of the

ground-truth target action a_t^* given the previous state-action sequence $(s_0, a_0, s_1, a_1, \dots, s_t)$. The target output action a_t^* is always defined as the next action in the ground-truth shortest-path trajectory from the agent’s current pose $s_t = (v_t, \psi_t, \theta_t)$ to the target location v^* .

Under the ‘teacher-forcing’ [Lamb et al., 2016] approach, at each step during training the ground-truth target action a_t^* is selected, to be conditioned on for the prediction of later outputs. However, this limits exploration to only states that are in ground-truth shortest-path trajectory, resulting in a changing input distribution between training and testing [Ross et al., 2011; Lamb et al., 2016]. To address this limitation, we also investigate ‘student-forcing’. In this approach, at each step the next action is sampled from the agent’s output probability distribution. Student-forcing is equivalent to an online version of DAGGER [Ross et al., 2011], or the ‘always sampling’ approach in scheduled sampling [Bengio et al., 2015]³.

Implementation Details We perform only minimal text preprocessing, converting all sentences to lower case, tokenizing on white space, and filtering words that do not occur at least five times. We set the simulator image resolution to 640×480 with a vertical field of view of 60 degrees. We set the number of hidden units in each LSTM to 512, the size of the input word embedding to 256, and the size of the input action embedding to 32. Embeddings are learned from random initialisation. We use dropout of 0.5 on embeddings, CNN features and within the attention model.

As we have discretised the agent’s heading and elevation changes in 30 degree increments, for fast training we extract and pre-cache all CNN feature vectors. We train using the Adam optimiser [Kingma and Ba, 2014] with weight decay and a batch size of 100. In all cases we train for a fixed number of iterations. As the evaluation is single-shot, at test time we use greedy decoding [Rennie et al., 2017]. Following standard practice our test set submission is trained on all training and validation data. Our models are implemented in PyTorch.

7.5.3 Additional Baselines

Learning free We report two learning-free baselines which we denote as RANDOM and SHORTEST. The RANDOM agent exploits the characteristics of the dataset by turning to a randomly selected heading, then completing a total of 5 successful forward

³Scheduled sampling has been shown to improve performance on tasks for which it is difficult to exactly determine the best next target output a_t^* for an arbitrary preceding sequence (e.g. language generation [Bengio et al., 2015]). However, in our task we can easily determine the shortest trajectory to the goal location from anywhere, and we found in initial experiments that scheduled sampling performed worse than student-forcing (i.e., always sampling).

	Trajectory Length (m)	Navigation Error (m)	Success (%)	Oracle Success (%)
Val Seen:				
SHORTEST	10.19	0.00	100	100
RANDOM	9.58	9.45	15.9	21.4
Teacher-forcing	10.95	8.01	27.1	36.7
Student-forcing	11.33	6.01	38.6	52.9
Val Unseen:				
SHORTEST	9.48	0.00	100	100
RANDOM	9.77	9.23	16.3	22.0
Teacher-forcing	10.67	8.61	19.6	29.1
Student-forcing	8.39	7.81	21.8	28.4
Test (unseen):				
SHORTEST	9.93	0.00	100	100
RANDOM	9.93	9.77	13.2	18.3
Human	11.90	1.61	86.4	90.2
Student-forcing	8.13	7.85	20.4	26.6

Table 7.1: Average R2R navigation results using evaluation metrics defined in Section 7.4.4. Our seq-2-seq model trained with student-forcing achieves promising results in previously explored environments (Val Seen). Generalisation to previously *unseen* environments (Val Unseen / Test) is far more challenging.

actions (when no forward action is available the agent selects right). The SHORTEST agent always follows the shortest path to the goal.

Human We quantify human performance by collecting human-generated trajectories for one third of the test set (1,390 instructions) using AMT. The collection procedure is similar to the dataset collection procedure described in Section 7.4.2, with two major differences. First, workers are provided with navigation instructions. Second, the entire scene environment is freely navigable in first-person by clicking on nearby viewpoints. In effect, workers are provided with the same information received by an agent in the simulator. To ensure a high standard, we paid workers bonuses for stopping within 3m of the true goal location.

7.6 Results

As illustrated in Table 7.1, our exploitative RANDOM agent achieves an average success rate of 13.2% on the test set (which appears to be slightly more challenging than the validation sets). In comparison, AMT workers achieve 86.4% success on the test set, illustrating the high quality of the dataset instructions. Nevertheless, people are

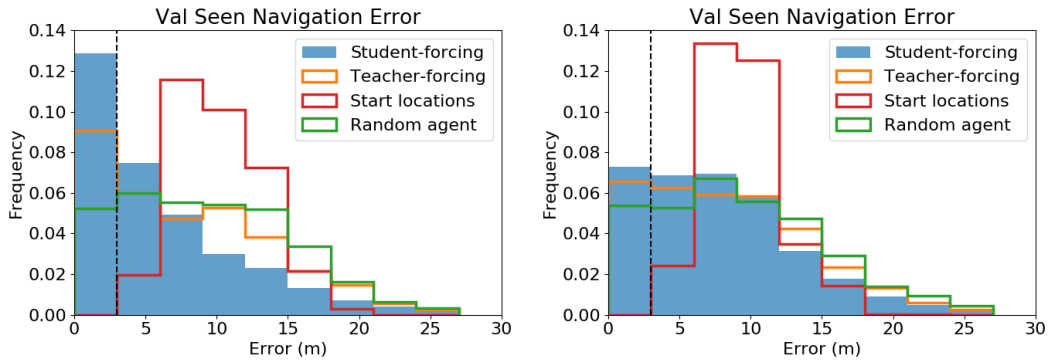


Figure 7.9: Distribution of navigation error in validation environments. In *previously seen* environments student-forcing training achieves 38.6% success ($< 3\text{m}$ navigation error), although this drops to 21.8% in *unseen* validation environments.

not infallible when it comes to navigation. For example, in the dataset we occasionally observe some confusion between right and left (although this is recoverable if the instructions contain enough visually-grounded references). In practice, people also use two additional mechanisms to reduce ambiguity that are not available here, namely gestures and dialogue.

With regard to the sequence-to-sequence model, student-forcing is a more effective training regime than teacher-forcing, although it takes longer to train as it explores more of the environment. Both methods improve significantly over the RANDOM baseline, as illustrated in Figure 7.9. Using the student-forcing approach we establish the first test set leaderboard result achieving a 20.4% success rate.

The most surprising aspect of the results is the significant difference between performance in seen and unseen validation environments (38.6% vs. 21.8% success for student-forcing). To better explain these results, in Figure 7.10 we plot validation performance during training. Even using strong regularisation (dropout and weight decay), performance in unseen environments plateaus quickly, but further training continues to improve performance in the training environments. This suggests that the visual groundings learned may be quite specific to the training environments.

Overall, the results illustrate the significant challenges involved in training agents that can generalise to perform well in previously unseen environments. The techniques and practices used to optimise performance on existing vision and language datasets are unlikely to be sufficient for models that are expected to operate in new environments.

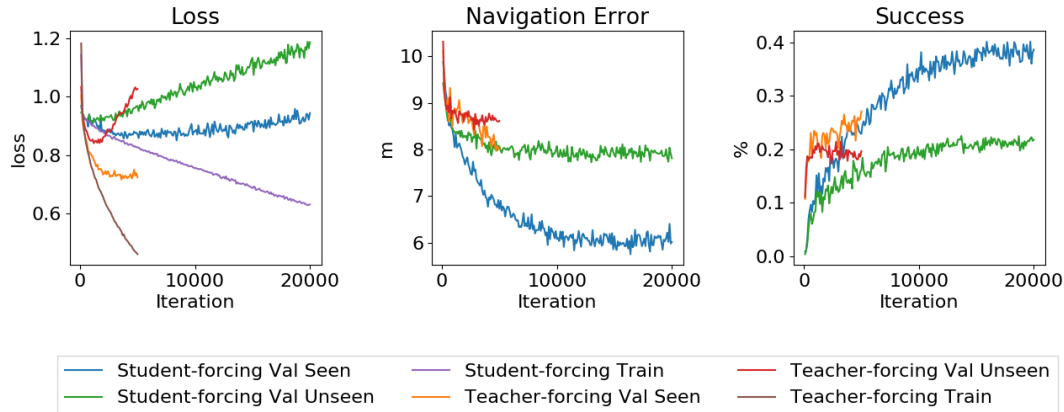


Figure 7.10: Validation loss, navigation error and success rate during training. Our experiments suggest that neural network approaches can strongly overfit to training environments, even with regularisation. This makes generalising to unseen environments challenging.

7.7 Chapter Summary

Vision and Language Navigation (VLN) is important because it represents a significant step towards capabilities that are critical for practical robotics. To further the investigation of VLN, in this chapter we introduced the Matterport3D Simulator constructed from dense RGB-D imagery of 90 real buildings [Chang et al., 2017]. This simulator achieves a unique and desirable trade-off between reproducibility, interactivity, and visual realism. Leveraging these advantages, we collected the Room-to-Room (R2R) dataset. The R2R dataset is the first dataset to evaluate the capability to follow natural language navigation instructions in previously unseen real images at building scale. To explore this task we investigated several baselines and a sequence-to-sequence neural network agent.

From this work we reach three main conclusions. First, VLN is interesting because existing vision and language methods can be successfully applied. To improve on our baseline sequence-to-sequence agent, we anticipate that the agent would benefit from a spatial memory [Gupta et al., 2017], rather than relying exclusively on recurrent neural network (RNN) memory vectors to encode complex geometric context. A visual attention mechanism may also be required. Second, the challenge of generalising to previously *unseen* environments is significant. We anticipate that solutions will require models that go beyond simply learning from the training data. For example, when faced with an instruction containing a novel visual concept such as ‘Egyptian hieroglyphs’, the VLN agent could use internet resources to search for

visual representations of this concept at test time, in order to successfully ground this concept in the environment. Third, crowd-sourced reconstructions of *real* locations are a highly-scalable and underutilised resource. The existing Matterport3D data release constitutes just 90 out of more than 700,000 building scans that have been already been uploaded by users [Matterport, 2017]. The process used to generate R2R is applicable to a host of related vision and language problems, particularly in robotics.

We hope that this simulator will benefit the community by providing a visually-realistic framework to investigate VLN and related problems such as navigation instruction generation, embodied visual question answering, human-robot dialog and transfer learning.

Conclusion and Future Directions

The interaction between vision and language is a promising area for research that is essential to unlocking numerous practical applications in robotics and artificial intelligence (AI). In this final chapter, we summarise the main contributions of this thesis and discuss some open problems and exciting directions for future research.

8.1 Summary

With the increasing focus on combining visual and linguistic learning comes an increasing need for methods that can automatically evaluate the language-based outputs of these models. Focusing on automatic image captioning as proxy for visual and linguistic understanding, in Chapter 4 we presented SPICE, a new metric for automatically evaluating image captions. Conceptually, SPICE differs from existing textual similarity metrics by using a meaning representation—the scene graph—to compare captions rather than using n-grams. Empirically, we demonstrated that SPICE is the most effective metric for automatically ranking image captioning models on the COCO dataset. To the extent that semantic parsing techniques continue to improve, SPICE also offers scope for further improvement.

Equipped with this more effective evaluation metric, in Chapter 5 we addressed the task of image captioning from the modelling side. Inspired by insights from neuroscience and psychology, we proposed a bottom-up and top-down visual attention mechanism. Although visual attention mechanisms have been widely studied, perhaps surprisingly our work is one of the first to more carefully consider how attention candidates are determined. Moving away from existing work that determines attention candidates according to a uniform grid of neural receptive fields, our approach makes objects the basis of attention in the model. Using this approach we achieved state of the art performance in image captioning as well as visual question answering (VQA), while simultaneously helping to improve the interpretability of the resulting systems.

Notwithstanding significant improvements in image captioning models in recent years, one of the major hurdles that remains is learning the long tail of visual concepts that occur rarely, but nevertheless must be understood to approach human performance on vision and language tasks. To begin to address this problem, in Chapter 6 we introduced constrained beam search, a novel constraint-based decoding algorithm, and demonstrated that it could be used to combine the outputs of image taggers and image captioning models at test time. This approach allows an image captioning model to scale to many more visual concepts, particularly if data-driven image tagging approaches are used. Using this approach we achieved state of the art performance on an out-of-domain image captioning task without degrading in-domain performance.

In Chapter 7, the final technical chapter of this thesis, we proposed the task of Vision-and-Language Navigation (VLN). To support this and other tasks involving embodied vision and language agents, we introduced the Matterport3D Simulator constructed from dense RGB-D imagery of 90 real buildings [Chang et al., 2017]. Additionally, we collected the Room-to-Room (R2R) dataset which is the first dataset to evaluate the capability to follow natural language navigation instructions in previously unseen real images at building scale. VLN is important because it represents a significant step towards capabilities critical for practical robotics. However, our investigation of several baselines and a sequence-to-sequence neural network agent indicates that the challenge of generalising to previously unseen environments is significant. More generally, the development of good models to solve problems that involve vision, language and action is still very much an open problem, as we discuss further below.

8.2 Future Directions

This thesis was motivated by the huge potential for intelligent agents to improve our lives via domestic service robots, voice-controlled drones, visually-aware virtual personal assistants, smart buildings and appliances, intelligent surveillance and search systems for querying image and video, language-based image and video editing software, personal navigation systems, etc. We conclude with a discussion of some promising research directions towards realising this potential.

8.2.1 Complete Agents

The traditional artificial intelligence (AI) paradigm defines intelligent agents in terms of their ability to *perceive*, *reason* and *act* [Winston, 1992]. However, in practice it seems

difficult to separate reasoning from perception and action planning. In our view, a ‘complete’ intelligent agent is one that can *see* (in the sense of perceiving and reasoning about the complex visual world), *communicate* (in natural language with humans and other agents), and *act* (by moving to gather new information, or by manipulating other objects or executing instructions). We prefer this more functional characterisation for several reasons. First, it gives much greater primacy to vision as the chief mode of perception for humans and increasingly, for agents operating in human environments. Second, separately identifying *communication*—which may involve perception, i.e. listening, and action, i.e. talking—explicitly captures the sociability required of intelligent agents. This reflects an increasing awareness of the need for AI systems that work cooperatively and collaboratively with people [Guszcza et al., 2017], and increasing evidence that the demands of social life may actually underpin the development of intelligence [Ashton et al., 2018]. Third, this viewpoint much more clearly illustrates the contributions of various AI subfields—such as computer vision, natural language processing (NLP) and robotics—to a complete agent, making a strong case for interdisciplinary research. Russell and Norvig [2003] stated 15 years ago that ‘one consequence of trying to build complete agents is the realization that the previously isolated subfields of AI might need to be reorganized somewhat when their results are to be tied together’. In our view, progress in these subfields is now sufficiently advanced for the areas of intersection to command a greater focus. Rather than continually seeking to separate computer vision, NLP and robotics problems, we see enormous potential in future research that focuses on ‘complete’ agents.

8.2.2 Large-scale 3D Datasets as Simulation Environments

Datasets play a central role in machine learning. For example, it is hard to overstate the impact of the ImageNet dataset [Russakovsky et al., 2015] in terms of its role in the development of computer vision algorithms, the renewed interest in artificial neural networks, the paradigm shift towards large-scale datasets and the focus on benchmark evaluations. What is sometimes forgotten is that it was the abundance of image data on crowd-based photo sharing websites such as Flickr¹ that made ImageNet possible, by providing the photographic raw material for a large annotated dataset. From our work on Vision-and-Language Navigation (VLN), we conclude that the burgeoning availability of 3D reconstructions of real locations at scale may be similarly transformative in terms of providing the raw material to develop datasets and simulation environments for training embodied vision and language agents. For

¹www.flickr.com

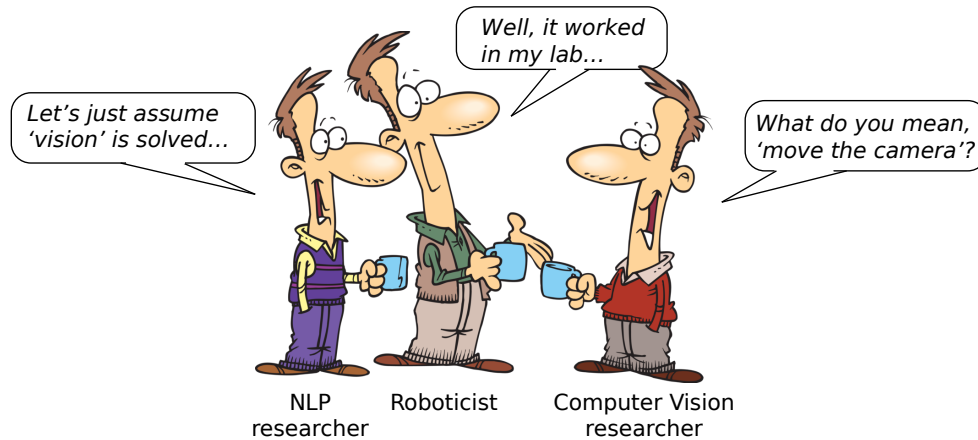


Figure 8.1: A light-hearted look at inter-disciplinary research in AI. Traditionally, research in computer vision, natural language processing (NLP) and robotics has tackled problems in relative isolation, subject to various limitations (e.g. the use of fully instrumented environments in NLP, lacking reproducibility and limited comparative evaluations in robotics, and the use of static image datasets in computer vision). Our challenge now as a research community is to devise environments, tasks and evaluation protocols to train complete agents that *see*, *communicate* and *act* (while still supporting reproducible evaluations).

example, owners of Matterport cameras have already uploaded more than 700,000 indoor building scans to a cloud platform [Matterport, 2017]. Although this data is not yet available to the research community, the ability of untrained users to generate high-quality 3D data and their willingness to do it without payment are promising signs for future data availability. Similarly, the introduction of millions of cars with sophisticated sensor suites to support partial or full autonomy will generate enormous amounts of outdoor 3D data. To take advantage of the opportunity presented by 3D reconstructed environments at scale, as illustrated in Figure 8.1 our challenge as a research community is to devise environments, tasks and evaluation protocols that capture the complexity of seeing, communicating and acting, while still supporting reproducible evaluations.

8.2.3 The Long Tail

One of the most pressing challenges to developing intelligent agents that achieve human performance is dealing with the long tail of visual and linguistic concepts that occur rarely, but are nonetheless tremendously important in practice. In the context of the tasks examined in this thesis, we illustrate various examples of this

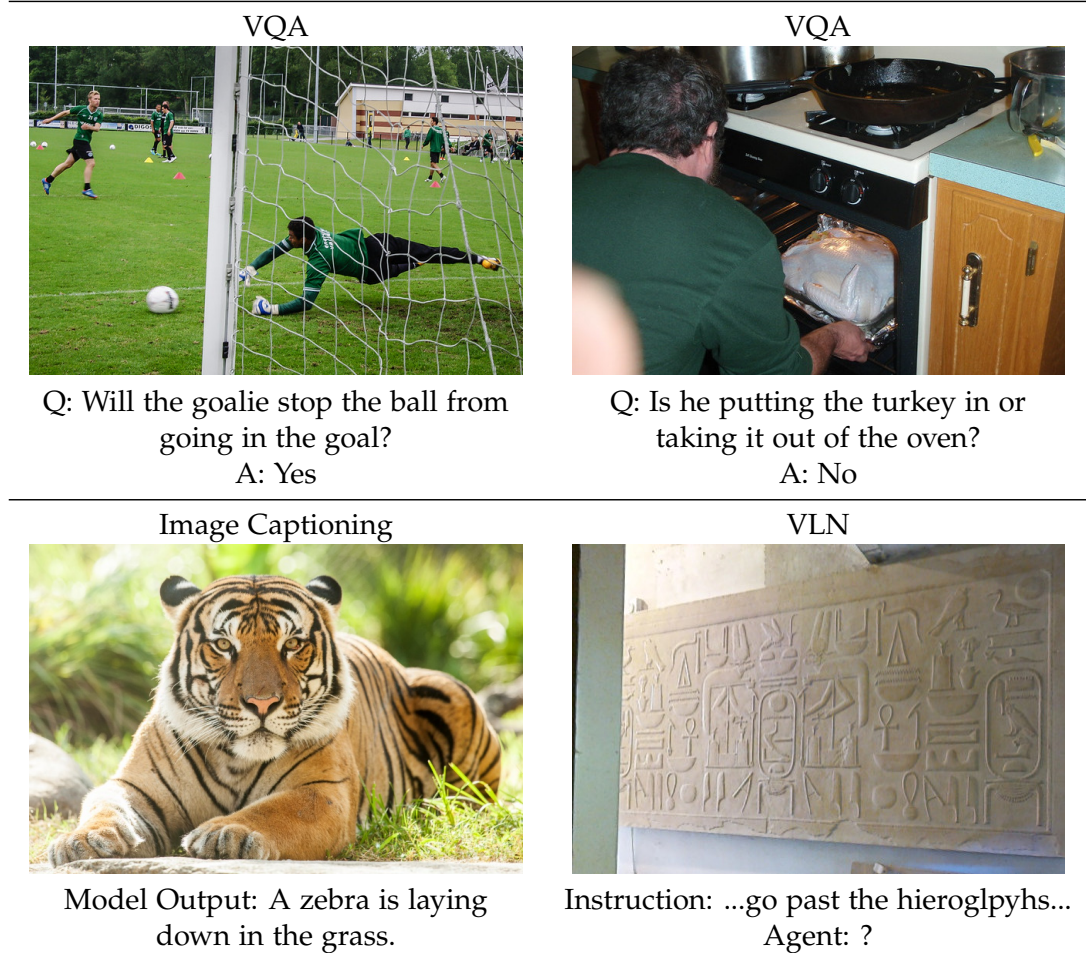


Figure 8.2: Difficulties caused by the long tail of visual and linguistic concepts. In VQA (top), there are not nearly enough training examples for current models to learn the concepts of stopping a goal or cooking a turkey. Similarly, datasets for image captioning and VLN (bottom) cannot possibly contain all the visual concepts that might be encountered at test time, such as tigers and hieroglyphs. New approaches are required to tackle this general problem, for example by incorporating internal physics simulations and improving transfer learning from external data sources.

phenomenon in Figure 8.2. For example, VQA models cannot correctly answer questions that require predicting the trajectory of a moving object, or reasoning about the change of state that occurs after a turkey spends time in the oven, since these concepts are not well captured in the training data. Similarly, in image captioning, models trained on the COCO dataset are insufficiently exposed to tigers during training, but are heavily exposed to zebra images, so they make predictable errors. Finally, in the VLN task, there are many novel visual concepts in the unseen validation set that are not mentioned in training, such as hieroglyphs.

For complex vision and language problems such as these, it seems very unlikely that larger curated datasets will be sufficient to capture all of the concepts, and ‘commonsense’ knowledge, that intelligent agents will require to reach human performance. Instead, a variety of complementary approaches may be required involving webly-supervised learning [Chen and Gupta, 2015] and transfer learning [Zamir et al., 2018a]. One exciting direction for making inferences about the physical world, such as whether the goalie will stop the ball, is using physics engines as ‘mental models’ to perform approximate, probabilistic internal simulations [Battaglia et al., 2013]. However, there remains much work to be done in order to automatically initialise these simulators at a useful level of abstraction from complex scenes like these.

Bibliography

- ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G. S.; DAVIS, A.; DEAN, J.; DEVIN, M.; ET AL., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, (2016). (cited on page 103)
- ADITYA, S.; YANG, Y.; BARAL, C.; FERMULLER, C.; AND ALOIMONOS, Y., 2015. From images to sentences through scene description graphs using commonsense reasoning and knowledge. *arXiv preprint arXiv:1511.03292*, (2015). (cited on pages 40 and 44)
- AGRAWAL, A.; BATRA, D.; AND PARIKH, D., 2016. Analyzing the behavior of visual question answering models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (cited on page 24)
- ALLAUZEN, C.; BYRNE, B.; DE GISPERT, A.; IGLESIAS, G.; AND RILEY, M., 2014. Push-down automata in statistical machine translation. *Computational Linguistics*, 40, 3 (2014), 687–723. (cited on page 81)
- AMMIRATO, P.; POIRSON, P.; PARK, E.; KOSECKA, J.; AND BERG, A. C., 2017. A dataset for developing and benchmarking active vision. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. (cited on page 99)
- ANDERSON, P.; FERNANDO, B.; JOHNSON, M.; AND GOULD, S., 2016. SPICE: Semantic Propositional Image Caption Evaluation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. (cited on pages 8, 65, and 88)
- ANDERSON, P.; FERNANDO, B.; JOHNSON, M.; AND GOULD, S., 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (cited on page 8)
- ANDERSON, P.; HE, X.; BUEHLER, C.; TENEY, D.; JOHNSON, M.; GOULD, S.; AND ZHANG, L., 2018a. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 8 and 110)

- ANDERSON, P.; WU, Q.; TENEY, D.; BRUCE, J.; JOHNSON, M.; SÜNDERHAUF, N.; REID, I.; GOULD, S.; AND VAN DEN HENGEL, A., 2018b. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 8)
- ANDREAS, J.; ROHRBACH, M.; DARRELL, T.; AND KLEIN, D., 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 11)
- ANTOL, S.; AGRAWAL, A.; LU, J.; MITCHELL, M.; BATRA, D.; ZITNICK, C. L.; AND PARIKH, D., 2015. VQA: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 2, 5, 11, 23, 65, 96, 97, 98, 103, and 106)
- ARMENI, I.; SAX, A.; ZAMIR, A. R.; AND SAVARESE, S., 2017. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *arXiv preprint arXiv:1702.01105*, (2017). (cited on page 103)
- ASHTON, B. J.; RIDLEY, A. R.; EDWARDS, E. K.; AND THORNTON, A., 2018. Cognitive performance is linked to group size and affects fitness in australian magpies. *Nature*, (2018). (cited on page 119)
- AUSTIN, J. L., 1962. *How to do things with words*. Oxford University Press. (cited on page 29)
- BAHDANAU, D.; CHO, K.; AND BENGIO, Y., 2015a. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*. (cited on page 10)
- BAHDANAU, D.; CHO, K.; AND BENGIO, Y., 2015b. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*. (cited on pages 52 and 109)
- BANARESCU, L.; BONIAL, C.; CAI, S.; GEORGESCU, M.; GRIFFITT, K.; HERMJAKOB, U.; KNIGHT, K.; KOEHN, P.; PALMER, M.; AND SCHNEIDER, N., 2012. Abstract meaning representation (AMR) 1.0 specification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (cited on pages 38 and 46)
- BARNARD, K.; DUYGULU, P.; FORSYTH, D.; FREITAS, N. D.; BLEI, D. M.; AND JORDAN, M. I., 2003. Matching words and pictures. *Journal of Machine Learning Research*, 3, Feb (2003), 1107–1135. (cited on page 10)

-
- BATTAGLIA, P. W.; HAMRICK, J. B.; AND TENENBAUM, J. B., 2013. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110, 45 (2013), 18327–18332. (cited on page 122)
- BEATTIE, C.; LEIBO, J. Z.; TEPLYASHIN, D.; WARD, T.; WAINWRIGHT, M.; KÜTTLER, H.; LEFRANCQ, A.; GREEN, S.; VALDÉS, V.; SADIK, A.; ET AL., 2016. Deepmind lab. *arXiv preprint arXiv:1612.03801*, (2016). (cited on pages 4, 96, and 98)
- BENGIO, S.; VINYALS, O.; JAITLY, N.; AND SHAZEER, N., 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 111)
- BERNARDI, R.; CAKICI, R.; ELLIOTT, D.; ERDEM, A.; ERDEM, E.; IKIZLER-CINBIS, N.; KELLER, F.; MUSCAT, A.; AND PLANK, B., 2016. Automatic description generation from images: A survey of models, datasets, and evaluation measures. *Journal of Artificial Intelligence Research*, 55 (2016), 409–442. (cited on page 27)
- BIGHAM, J. P.; JAYANT, C.; JI, H.; LITTLE, G.; MILLER, A.; MILLER, R. C.; MILLER, R.; TATAROWICZ, A.; WHITE, B.; WHITE, S.; ET AL., 2010. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, 333–342. ACM. (cited on page 11)
- BODEN, M. A., 2008. *Mind as Machine: A History of Cognitive Science*. Oxford University Press. (cited on page 9)
- BROCKMAN, G.; CHEUNG, V.; PETERSSON, L.; SCHNEIDER, J.; SCHULMAN, J.; TANG, J.; AND ZAREMBA, W., 2016. OpenAI gym. *arXiv preprint arXiv:1606.01540*, (2016). (cited on page 103)
- BRODEUR, S.; PEREZ, E.; ANAND, A.; GOLEMO, F.; CELOTTI, L.; STRUB, F.; ROUAT, J.; LAROCHELLE, H.; AND COURVILLE, A., 2017. HoME: A household multimodal environment. *arXiv preprint arXiv:1711.11017*, (2017). (cited on page 98)
- BUSCHMAN, T. J. AND MILLER, E. K., 2007. Top-down versus bottom-up control of attention in the prefrontal and posterior parietal cortices. *Science*, 315, 5820 (2007), 1860–1862. (cited on page 53)
- CAI, S. AND KNIGHT, K., 2013. Smatch: an evaluation metric for semantic feature structures. In *ACL (2)*, 748–752. (cited on page 38)
- CHANG, A.; DAI, A.; FUNKHOUSER, T.; HALBER, M.; NIESSNER, M.; SAVVA, M.; SONG, S.; ZENG, A.; AND ZHANG, Y., 2017. Matterport3d: Learning from RGB-D data in

- indoor environments. *International Conference on 3D Vision (3DV)*, (2017). (cited on pages 4, 8, 96, 98, 99, 100, 106, 109, 114, and 118)
- CHAPLOT, D. S.; SATHYENDRA, K. M.; PASUMARTHI, R. K.; RAJAGOPAL, D.; AND SALAKHUTDINOV, R., 2018. Gated-attention architectures for task-oriented language grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*. (cited on pages 4, 95, 97, and 99)
- CHEN, D. L. AND MOONEY, R. J., 2011. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*. (cited on pages 4, 95, 96, and 97)
- CHEN, M.; ZHENG, A. X.; AND WEINBERGER, K. Q., 2013. Fast image tagging. In *Proceedings of the International Conference on Machine Learning (ICML)*. (cited on page 77)
- CHEN, X. AND GUPTA, A., 2015. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (cited on page 122)
- CHEN, X.; HAO FANG, T.-Y. L.; VEDANTAM, R.; GUPTA, S.; DOLLAR, P.; AND ZITNICK, C. L., 2015. Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv preprint arXiv:1504.00325*, (2015). (cited on pages 2, 5, 21, 27, 29, 39, 41, 51, 77, 91, 93, 95, 96, 97, and 103)
- CHO, K.; VAN MERRIENBOER, B.; GULCEHRE, C.; BOUGARES, F.; SCHWENK, H.; AND BENGIO, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (cited on page 62)
- CORBETTA, M. AND SHULMAN, G. L., 2002. Control of goal-directed and stimulus-driven attention in the brain. *Nature Reviews Neuroscience*, 3, 3 (2002), 201–215. (cited on page 53)
- CUI, Y.; YANG, G.; VEIT, A.; HUANG, X.; AND BELONGIE, S., 2018. Learning to evaluate image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 47)
- DAI, A.; CHANG, A. X.; SAVVA, M.; HALBER, M.; FUNKHOUSER, T.; AND NIESSNER, M., 2017a. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 99)

-
- DAI, B.; FIDLER, S.; URTASUN, R.; AND LIN, D., 2017b. Towards diverse and natural image descriptions via a conditional GAN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (cited on page 11)
- DAS, A.; DATTA, S.; GKIOXARI, G.; LEE, S.; PARIKH, D.; AND BATRA, D., 2018. Embodied Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 98 and 103)
- DAS, A.; KOTTUR, S.; GUPTA, K.; SINGH, A.; YADAV, D.; MOURA, J. M. F.; PARIKH, D.; AND BATRA, D., 2017. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 12, 97, and 98)
- DAUPHIN, Y. N.; FAN, A.; AULI, M.; AND GRANGIER, D., 2016. Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083*, (2016). (cited on page 61)
- DE MARNEFFE, M.-C.; DOZAT, T.; SILVEIRA, N.; HAVERINEN, K.; GINTER, F.; NIVRE, J.; AND MANNING, C. D., 2014. Universal stanford dependencies: A cross-linguistic typology. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, vol. 14, 4585–4592. (cited on page 31)
- DEEMTER, K. V.; THEUNE, M.; AND KRAHMER, E., 2005. Real versus template-based natural language generation: A false opposition? *Computational Linguistics*, 31, 1 (2005), 15–24. (cited on page 9)
- DENKOWSKI, M. AND LAVIE, A., 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL) Workshop on Statistical Machine Translation*, vol. 6. (cited on pages 27, 28, 34, 65, and 88)
- DEVLIN, J.; CHENG, H.; FANG, H.; GUPTA, S.; DENG, L.; HE, X.; ZWEIG, G.; AND MITCHELL, M., 2015a. Language models for image captioning: The quirks and what works. *arXiv preprint arXiv:1505.01809*, (2015). (cited on pages 80 and 85)
- DEVLIN, J.; GUPTA, S.; GIRSHICK, R. B.; MITCHELL, M.; AND ZITNICK, C. L., 2015b. Exploring nearest neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467*, (2015). (cited on page 10)
- DONAHUE, J.; A. HENDRICKS, L.; GUADARRAMA, S.; ROHRBACH, M.; VENUGOPALAN, S.; SAENKO, K.; AND DARRELL, T., 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 10, 12, 79, and 85)

- EGLY, R.; DRIVER, J.; AND RAFAL, R. D., 1994. Shifting visual attention between objects and locations: evidence from normal and parietal lesion subjects. *Journal of Experimental Psychology: General*, 123, 2 (1994), 161. (cited on pages 3 and 54)
- ELLEBRACHT, L.; RAMISA, A.; SWAROOP, P.; CORDERO, J.; MORENO-NOGUER, F.; AND QUATTONI, A., 2015. Semantic tuples for evaluation of image sentence generation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 4th Workshop on Vision and Language*, (2015). (cited on page 38)
- ELLIOT, D. AND DE VRIES, A. P., 2015. Describing images using inferred visual dependency representations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. (cited on page 79)
- ELLIOTT, D. AND KELLER, F., 2014. Comparing automatic evaluation measures for image description. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. (cited on pages 3, 27, and 28)
- EVERINGHAM, M.; VAN GOOL, L.; WILLIAMS, C. K. I.; WINN, J.; AND ZISSERMAN, A., 2010. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88, 2 (2010), 303–338. (cited on page 21)
- FANG, H.; GUPTA, S.; IANDOLA, F. N.; SRIVASTAVA, R.; DENG, L.; DOLLAR, P.; GAO, J.; HE, X.; MITCHELL, M.; PLATT, J. C.; ZITNICK, C. L.; AND ZWEIG, G., 2015. From captions to visual concepts and back. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 2, 10, 43, and 79)
- FARHADI, A.; HEJRATI, M.; SADEGHI, M. A.; YOUNG, P.; RASHTCHIAN, C.; HOCKENMAIER, J.; AND FORSYTH, D., 2010. Every picture tells a story: Generating sentences from images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 15–29. (cited on pages 9 and 10)
- FELLBAUM, C., 1998. *WordNet: An Electronic Lexical Database*. Bradford Books. (cited on page 91)
- FERNANDO, B.; ANDERSON, P.; HUTTER, M.; AND GOULD, S., 2016. Discriminative hierarchical rank pooling for activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 8)
- FLANIGAN, J.; THOMSON, S.; CARBONELL, J.; DYER, C.; AND SMITH, N. A., 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. (cited on page 38)

-
- FUKUI, A.; PARK, D. H.; YANG, D.; ROHRBACH, A.; DARRELL, T.; AND ROHRBACH, M., 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (cited on pages 11, 51, 53, and 70)
- GEMAN, D.; GEMAN, S.; HALLONQUIST, N.; AND YOUNES, L., 2015. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 112, 12 (2015), 3618–3623. (cited on page 11)
- GHAZVININEJAD, M.; SHI, X.; CHOI, Y.; AND KNIGHT, K., 2016. Generating topical poetry. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (cited on page 81)
- GIMÉNEZ, J. AND MÀRQUEZ, L., 2007. Linguistic features for automatic evaluation of heterogenous MT systems. In *ACL Second Workshop on Statistical Machine Translation*. (cited on pages 29 and 38)
- GOODFELLOW, I.; BENGIO, Y.; AND COURVILLE, A., 2016. *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>. (cited on pages 12 and 13)
- GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; AND BENGIO, Y., 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 47)
- GORDON, D.; KEMBHAVI, A.; RASTEGARI, M.; REDMON, J.; FOX, D.; AND FARHADI, A., 2018. IQA: Visual question answering in interactive environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 98 and 103)
- GOYAL, Y.; KHOT, T.; SUMMERS-STAY, D.; BATRA, D.; AND PARIKH, D., 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 2, 3, 5, 11, 24, 51, 65, 70, 77, 95, 96, and 97)
- GRAVES, A.; MOHAMED, A.-R.; AND HINTON, G., 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6645–6649. (cited on page 14)
- GUADARRAMA, S.; RIANO, L.; GOLLAND, D.; GO, D.; JIA, Y.; KLEIN, D.; ABBEEL, P.; DARRELL, T.; ET AL., 2013. Grounding spatial relations for human-robot interaction. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. (cited on pages 4, 95, and 97)

- GUPTA, S.; DAVIDSON, J.; LEVINE, S.; SUKTHANKAR, R.; AND MALIK, J., 2017. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 99 and 114)
- GURARI, D.; LI, Q.; STANGL, A. J.; GUO, A.; LIN, C.; GRAUMAN, K.; LUO, J.; AND BIGHAM, J. P., 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 2)
- GUSZCZA, J.; EVANS-GREENWOOD, P.; AND LEWIS, H., 2017. Cognitive collaboration: Why humans and computers think better together. *Deloitte Review*, 20 (Jan. 2017). (cited on page 119)
- HALE, J., 2001. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. (cited on page 37)
- HE, K.; ZHANG, X.; REN, S.; AND SUN, J., 2016a. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 56, 65, 85, 91, and 110)
- HE, K.; ZHANG, X.; REN, S.; AND SUN, J., 2016b. Identity mappings in deep residual networks. *arXiv preprint arXiv:1603.05027*, (2016). (cited on page 65)
- HENDRICKS, L. A.; VENUGOPALAN, S.; ROHRBACH, M.; MOONEY, R.; SAENKO, K.; AND DARRELL, T., 2016. Deep Compositional Captioning: Describing Novel Object Categories without Paired Training Data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 77, 78, 79, 86, 87, 89, and 90)
- HOCHREITER, S. AND SCHMIDHUBER, J., 1997. Long Short-Term Memory. *Neural Computation*, (1997). (cited on pages 14, 58, 85, and 109)
- HODOSH, M.; YOUNG, P.; AND HOCKENMAIER, J., 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47 (2013), 853–899. (cited on pages 3, 10, 11, 27, 28, 40, and 44)
- HUANG, A. S.; TELLEX, S.; BACHRACH, A.; KOLLAR, T.; ROY, D.; AND ROY, N., 2010. Natural language command of an autonomous micro-air vehicle. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. (cited on page 97)

-
- HUH, M.; AGRAWAL, P.; AND EFROS, A. A., 2016. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, (2016). (cited on pages 13 and 20)
- JABRI, A.; JOULIN, A.; AND VAN DER MAATEN, L., 2016. Revisiting visual question answering baselines. *arXiv preprint arXiv:1606.08390*, (2016). (cited on pages 24, 55, and 61)
- JADERBERG, M.; SIMONYAN, K.; ZISSERMAN, A.; AND KAVUKCUOGLU, K., 2015. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 55)
- JIA, Y.; SHELHAMER, E.; DONAHUE, J.; KARAYEV, S.; LONG, J.; GIRSHICK, R.; GUADARRAMA, S.; AND DARRELL, T., 2014. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, (2014). (cited on pages 84 and 103)
- JIN, J.; FU, K.; CUI, R.; SHA, F.; AND ZHANG, C., 2015. Aligning where to see and what to tell: image caption with region-based attention and scene factorization. *arXiv preprint arXiv:1506.06272*, (2015). (cited on page 55)
- JOHNSON, J.; KRISHNA, R.; STARK, M.; LI, L.-J.; SHAMMA, D. A.; BERNSTEIN, M. S.; AND FEI-FEI, L., 2015. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 30, 32, and 33)
- JOHNSON, M.; ANDERSON, P.; DRAS, M.; AND STEEDMAN, M., 2018. Predicting accuracy on large datasets from smaller pilot data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. (cited on page 8)
- KARPATHY, A. AND FEI-FEI, L., 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 23, 40, 64, 79, and 85)
- KARPATHY, A.; JOULIN, A.; AND LI, F., 2014. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 10)
- KAZEMI, V. AND ELQURSH, A., 2017. Show, ask, attend, and answer: A strong baseline for visual question answering. *arXiv preprint arXiv:1704.03162*, (2017). (cited on pages 11, 55, and 61)
- KAZEMZADEH, S.; ORDONEZ, V.; MATTEN, M.; AND BERG, T. L., 2014. Referit game: Referring to objects in photographs of natural scenes. In *Proceedings of the Conference*

- on Empirical Methods in Natural Language Processing (EMNLP)*. (cited on pages 12 and 98)
- KEMPKA, M.; WYDMUCH, M.; RUNC, G.; TOCZEK, J.; AND JAŚKOWSKI, W., 2016. ViZ-Doom: A Doom-based AI research platform for visual reinforcement learning. In *IEEE Conference on Computational Intelligence and Games*. (cited on pages 4, 96, 98, and 99)
- KILICKAYA, M.; ERDEM, A.; IKIZLER-CINBIS, N.; AND ERDEM, E., 2017. Re-evaluating automatic metrics for image captioning. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. (cited on page 37)
- KINGMA, D. AND BA, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, (2014). (cited on page 111)
- KINGSBURY, P. AND PALMER, M., 2002. From treebank to propbank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*. (cited on page 38)
- KIPERWASSER, E. AND GOLDBERG, Y., 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *arXiv preprint arXiv:1603.04351*, (2016). (cited on page 16)
- KLEIN, D. AND MANNING, C. D., 2003. Accurate unlexicalized parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. (cited on pages 31 and 37)
- KOEHN, P., 2010. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edn. ISBN 0521874157, 9780521874151. (cited on pages 17 and 80)
- KOLLAR, T.; TELLEX, S.; ROY, D.; AND ROY, N., 2010. Toward understanding natural language directions. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, 259–266. IEEE. (cited on page 97)
- KOLVE, E.; MOTTAGHI, R.; GORDON, D.; ZHU, Y.; GUPTA, A.; AND FARHADI, A., 2017. AI2-THOR: An interactive 3d environment for visual AI. *arXiv preprint arXiv:1712.05474*, (2017). (cited on page 98)
- KRAUSE, J.; SAPP, B.; HOWARD, A.; ZHOU, H.; TOSHEV, A.; DUERIG, T.; PHILBIN, J.; AND FEI-FEI, L., 2016. The unreasonable effectiveness of noisy data for fine-grained

-
- recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*. (cited on page 77)
- KRISHNA, R.; ZHU, Y.; GROTH, O.; JOHNSON, J.; HATA, K.; KRAVITZ, J.; CHEN, S.; KALANTIDIS, Y.; LI, L.-J.; SHAMMA, D. A.; BERNSTEIN, M.; AND FEI-FEI, L., 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, (2016). (cited on pages 24, 26, 33, 56, 62, 63, and 65)
- KRIZHEVSKY, A.; SUTSKEVER, I.; AND HINTON, G. E., 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*. (cited on page 13)
- KULKARNI, G.; PREMRAJ, V.; ORDONEZ, V.; DHAR, S.; LI, S.; CHOI, Y.; BERG, A. C.; AND BERG, T. L., 2013. Babytalk: Understanding and generating simple image descriptions. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35, 12 (2013), 2891–2903. (cited on pages 3, 9, 27, and 28)
- KULKARNI, T. D.; SAEEDI, A.; GAUTAM, S.; AND GERSHMAN, S. J., 2016. Deep successor reinforcement learning. *arXiv preprint arXiv:1606.02396*, (2016). (cited on page 99)
- LAMB, A. M.; GOYAL, A. G. A. P.; ZHANG, Y.; ZHANG, S.; COURVILLE, A. C.; AND BENGIO, Y., 2016. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 111)
- LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W.; AND JACKEL, L. D., 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1, 4 (1989), 541–551. (cited on pages 12 and 13)
- LEVY, R., 2008. Expectation-based syntactic comprehension. *Cognition*, 106, 3 (2008), 1126–1177. (cited on page 37)
- LIN, C., 2004. Rouge: a package for automatic evaluation of summaries. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) Workshop: Text Summarization Branches Out*. (cited on pages 27, 28, and 65)
- LIN, D.; FIDLER, S.; KONG, C.; AND URTASUN, R., 2014a. Visual semantic search: Retrieving videos via complex textual queries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 30)
- LIN, T.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLAR, P.; AND ZITNICK, C. L., 2014b. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision (ECCV)*. (cited on pages 11, 21, 23, 25, 37, 62, 64, and 87)

- LIU, S.; ZHU, Z.; YE, N.; GUADARRAMA, S.; AND MURPHY, K., 2017a. Improved image captioning via policy gradient optimization of SPIDeR. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 2, 10, and 37)
- LIU, S.; ZHU, Z.; YE, N.; GUADARRAMA, S.; AND MURPHY, K., 2017b. Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (cited on page 68)
- LIU, W.; ANGUELOV, D.; ERHAN, D.; SZEGEDY, C.; REED, S.; FU, C.-Y.; AND BERG, A. C., 2016. SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*. (cited on page 56)
- LO, C.-K.; TUMULURU, A. K.; AND WU, D., 2012. Fully automatic semantic MT evaluation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) Seventh Workshop on Statistical Machine Translation*. (cited on page 38)
- LU, J.; LIN, X.; BATRA, D.; AND PARIKH, D., 2015. Deeper LSTM and normalized CNN visual question answering model. https://github.com/VT-vision-lab/VQA_LSTM_CNN. (cited on page 70)
- LU, J.; XIONG, C.; PARIKH, D.; AND SOCHER, R., 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 11, 51, 53, 58, and 68)
- LU, J.; YANG, J.; BATRA, D.; AND PARIKH, D., 2016. Hierarchical question-image co-attention for visual question answering. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on pages 11, 51, 53, 55, and 110)
- LU, J.; YANG, J.; BATRA, D.; AND PARIKH, D., 2018. Neural baby talk. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 9 and 79)
- LUONG, M.-T.; PHAM, H.; AND MANNING, C. D., 2014. Effective approaches to attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (cited on pages 52, 53, and 110)
- MACHACEK, M. AND BOJAR, O., 2014. Results of the WMT14 metrics shared task. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) Ninth Workshop on Statistical Machine Translation*. (cited on page 39)

-
- MACMAHON, M.; STANKIEWICZ, B.; AND KUIPERS, B., 2006. Walk the talk: Connecting language, knowledge, and action in route instructions. In *Proceedings of the AAAI Conference on Artificial Intelligence*. (cited on page 97)
- MALINOWSKI, M. AND FRITZ, M., 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 11)
- MANNING, C. D.; SURDEANU, M.; BAUER, J.; FINKEL, J.; BETHARD, S. J.; AND MC-CLOSKEY, D., 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, 55–60. <http://www.aclweb.org/anthology/P/P14/P14-5010>. (cited on page 35)
- MAO, J.; JONATHAN, H.; TOSHEV, A.; CAMBURU, O.; YUILLE, A.; AND MURPHY, K., 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 12 and 98)
- MAO, J.; XU, W.; YANG, Y.; WANG, J.; HUANG, Z.; AND YUILLE, A., 2015. Deep captioning with multimodal recurrent neural networks (m-RNN). In *Proceedings of the International Conference on Learning Representations (ICLR)*. (cited on pages 10, 79, and 85)
- MATTERPORT, 2017. Press release 23 october 2017. (cited on pages 115 and 120)
- MEI, H.; BANSAL, M.; AND WALTER, M. R., 2016. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *Proceedings of the AAAI Conference on Artificial Intelligence*. (cited on pages 4, 95, and 97)
- MILLER, A. H.; FENG, W.; FISCH, A.; LU, J.; BATRA, D.; BORDES, A.; PARIKH, D.; AND WESTON, J., 2017. ParlAI: A dialog research software platform. *arXiv preprint arXiv:1705.06476*, (2017). (cited on page 103)
- MILLER, G. A., 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38, 11 (Nov. 1995), 39–41. (cited on pages 25, 34, and 84)
- MIROWSKI, P.; PASCANU, R.; VIOLA, F.; SOYER, H.; BALLARD, A.; BANINO, A.; DENIL, M.; GOROSHIN, R.; SIFRE, L.; KAVUKCUOGLU, K.; ET AL., 2017. Learning to navigate in complex environments. In *Proceedings of the International Conference on Learning Representations (ICLR)*. (cited on page 109)
- MISRA, D. K.; LANGFORD, J.; AND ARTZI, Y., 2017. Mapping instructions and visual observations to actions with reinforcement learning. In *Proceedings of the Conference*

- on Empirical Methods in Natural Language Processing (EMNLP)*. (cited on pages 4, 95, 99, and 109)
- NAIR, V. AND HINTON, G. E., 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*. (cited on page 12)
- NATHAN SILBERMAN, P. K., DEREK HOIEM AND FERGUS, R., 2012. Indoor segmentation and support inference from RGBD images. In *Proceedings of the European Conference on Computer Vision (ECCV)*. (cited on page 99)
- NAVALPAKKAM, V. AND ITTI, L., 2006. An integrated model of top-down and bottom-up attention for optimizing detection speed. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 53)
- NORMAN, D. A., 2002. *The Design of Everyday Things*. Basic Books, Inc., New York, NY, USA. ISBN 9780465067107. (cited on page 106)
- OLIVA, A. AND TORRALBA, A., 2006. Building the gist of a scene: the role of global image features in recognition. *Progress in brain research*, 155 (2006), 23–36. (cited on page 9)
- PALMER, M.; GILDEA, D.; AND KINGSBURY, P., 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31, 1 (2005), 71–106. (cited on page 38)
- PAPINENI, K.; ROUKOS, S.; WARD, T.; AND ZHU, W., 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. (cited on pages 2, 27, 28, and 65)
- PEDERSOLI, M.; LUCAS, T.; SCHMID, C.; AND VERBEEK, J., 2017. Areas of attention for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (cited on page 55)
- PENNINGTON, J.; SOCHER, R.; AND MANNING, C. D., 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (cited on pages 64 and 86)
- PLUMMER, B. A.; WANG, L.; CERVANTES, C. M.; CAICEDO, J. C.; HOCKENMAIER, J.; AND LAZEBNIK, S., 2015. Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 33)

-
- PRADHAN, S. S.; WARD, W.; HACIOGLU, K.; MARTIN, J. H.; AND JURAFSKY, D., 2004. Shallow semantic parsing using support vector machines. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): Human Language Technologies*, 233–240. (cited on page 38)
- RASHTCHIAN, C.; YOUNG, P.; HODOSH, M.; AND HOCKENMAIER, J., 2010. Collecting image annotations using amazon’s mechanical turk. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*. (cited on pages 11 and 40)
- REDMON, J.; DIVVALA, S. K.; GIRSHICK, R. B.; AND FARHADI, A., 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 56)
- REITER, E. AND DALE, R., 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3, 1 (Mar. 1997), 57–87. (cited on page 9)
- REN, S.; HE, K.; GIRSHICK, R.; AND SUN, J., 2015a. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on pages 6, 54, 55, 63, and 72)
- REN, S.; HE, K.; GIRSHICK, R.; AND SUN, J., 2015b. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on pages 71 and 77)
- RENNIE, S. J.; MARCHERET, E.; MROUEH, Y.; ROSS, J.; AND GOEL, V., 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 10, 51, 53, 58, 60, 65, 66, 67, 68, and 111)
- ROHRBACH, A.; ROHRBACH, M.; HU, R.; DARRELL, T.; AND SCHIELE, B., 2016. Grounding of textual phrases in images by reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*. (cited on page 12)
- ROSS, S.; GORDON, G.; AND BAGNELL, D., 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. (cited on page 111)
- RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; AND FEI-FEI, L., 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, (2015). (cited on pages 13, 19, 55, 56, 65, 85, 87, 91, 110, and 119)

- RUSSELL, S. J. AND NORVIG, P., 2003. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edn. ISBN 0137903952. (cited on page 119)
- SAVVA, M.; CHANG, A. X.; DOSOVITSKIY, A.; FUNKHOUSER, T.; AND KOLTUN, V., 2017. MINOS: Multimodal indoor simulator for navigation in complex environments. *arXiv preprint arXiv:1712.03931*, (2017). (cited on pages 98 and 103)
- SCHOLL, B. J., 2001. Objects and attention: The state of the art. *Cognition*, 80, 1 (2001), 1–46. (cited on pages 3, 54, and 71)
- SCHUSTER, S.; KRISHNA, R.; CHANG, A.; FEI-FEI, L.; AND MANNING, C. D., 2015. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 4th Workshop on Vision and Language*. (cited on pages 30, 31, 32, and 33)
- SIMONYAN, K. AND ZISSERMAN, A., 2015. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*. (cited on page 85)
- SIPSER, M., 2012. *Introduction to the Theory of Computation*. Cengage Learning, 3rd edn. (cited on page 81)
- SONG, S.; LICHTENBERG, S. P.; AND XIAO, J., 2015. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 99)
- SRIVASTAVA, R. K.; GREFF, K.; AND SCHMIDHUBER, J., 2015. Highway networks. *arXiv preprint arXiv:1505.00387v1*, (2015). (cited on page 61)
- STANOJEVIĆ, M.; KAMRAN, A.; KOEHN, P.; AND BOJAR, O., 2015. Results of the WMT15 metrics shared task. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) Tenth Workshop on Statistical Machine Translation*. (cited on page 39)
- SUTSKEVER, I.; MARTENS, J.; AND HINTON, G., 2011. Generating text with recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*. (cited on pages 13 and 16)
- SUTSKEVER, I.; VINYALS, O.; AND LE, Q. V., 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on pages 10, 16, 80, and 110)

-
- TAI, L. AND LIU, M., 2016. Towards cognitive exploration through deep reinforcement learning for mobile robots. *arXiv preprint arXiv:1610.01733*, (2016). (cited on page 99)
- TAPASWI, M.; ZHU, Y.; STIEFELHAGEN, R.; TORRALBA, A.; URTASUN, R.; AND FIDLER, S., 2016. MovieQA: Understanding stories in movies through question-answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 98)
- TELLEX, S.; KOLLAR, T.; DICKERSON, S.; WALTER, M. R.; BANERJEE, A. G.; TELLER, S. J.; AND ROY, N., 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. (cited on pages 4, 95, 96, and 97)
- TENEY, D.; ANDERSON, P.; HE, X.; AND VAN DEN HENGEL, A., 2018. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 8, 61, and 64)
- TENEY, D. AND VAN DEN HENGEL, A., 2016. Zero-shot visual question answering. *arXiv preprint arXiv:1611.05546*, (2016). (cited on page 61)
- TESSLER, C.; GIVONY, S.; ZAHAVY, T.; MANKOWITZ, D. J.; AND MANNOR, S., 2017. A deep hierarchical approach to lifelong learning in minecraft. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1553–1561. (cited on page 99)
- THOMEE, B.; SHAMMA, D. A.; FRIEDLAND, G.; ELIZALDE, B.; NI, K.; POLAND, D.; BORTH, D.; AND LI, L.-J., 2016. YFCC100M: the new data in multimedia research. *Communications of the ACM*, 59, 2 (2016), 64–73. (cited on page 25)
- TORRALBA, A. AND EFROS, A. A., 2011. Unbiased look at dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 6, 37, and 103)
- TRAN, K.; HE, X.; ZHANG, L.; SUN, J.; CARAPCEA, C.; THRASHER, C.; BUEHLER, C.; AND SIENKIEWICZ, C., 2016. Rich Image Captioning in the Wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. (cited on pages 3, 6, and 77)
- TREISMAN, A., 1982. Perceptual grouping and attention in visual search for features and for objects. *Journal of Experimental Psychology: Human Perception and Performance*, 8, 2 (1982), 194. (cited on page 69)

- TREISMAN, A. M. AND GELADE, G., 1980. A feature-integration theory of attention. *Cognitive Psychology*, 12 (1980), 97–136. (cited on page 69)
- UIJLINGS, J. R.; VAN DE SANDE, K. E.; GEVERS, T.; AND SMEULDERS, A. W., 2013. Selective search for object recognition. *International Journal of Computer Vision (IJCV)*, (2013). (cited on page 55)
- VAN MILTENBURG, E.; MORANTE, R.; AND ELLIOTT, D., 2016. Pragmatic factors in image description: the case of negations. *arXiv preprint arXiv:1606.06164*, (2016). (cited on page 37)
- VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; AND POLOSUKHIN, I., 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 53)
- VEDANTAM, R.; ZITNICK, C. L.; AND PARIKH, D., 2015. CIDEr: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 2, 28, 40, 44, 60, 65, and 89)
- VENUGOPALAN, S.; HENDRICKS, L. A.; ROHRBACH, M.; MOONEY, R. J.; DARRELL, T.; AND SAENKO, K., 2017. Captioning Images with Diverse Objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 79, 87, and 89)
- VENUGOPALAN, S.; XU, H.; DONAHUE, J.; ROHRBACH, M.; MOONEY, R.; AND SAENKO, K., 2015. Translating videos to natural language using deep recurrent neural networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): Human Language Technologies*. (cited on page 12)
- VINYALS, O.; TOSHEV, A.; BENGIO, S.; AND ERHAN, D., 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 1, 10, 16, 80, and 85)
- VOGEL, A. AND JURAFSKY, D., 2010. Learning to follow navigational directions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. (cited on page 97)
- WANG, C.; XUE, N.; AND PRADHAN, S., 2015. A transition-based algorithm for AMR parsing. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): Human Language Technologies*. (cited on page 38)

-
- WANG, Y.-S.; LIU, C.; ZENG, X.; AND YUILLE, A., 2018. Scene graph parsing as dependency parsing. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. (cited on page 46)
- WERLING, K.; ANGELI, G.; AND MANNING, C., 2015. Robust subgraph generation improves abstract meaning representation parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. (cited on page 38)
- WIERSTRA, D.; FOERSTER, A.; PETERS, J.; AND SCHMIDHUBER, J., 2007. Solving deep memory POMDPs with recurrent policy gradients. In *International Conference on Artificial Neural Networks*. (cited on page 110)
- WILLIAMS, R. J., 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8, 3-4 (May 1992), 229–256. (cited on page 60)
- WINOGRAD, T., 1971. Procedures as a representation for data in a computer program for understanding natural language. Technical report, Massachusetts Institute of Technology. (cited on page 97)
- WINSTON, P. H., 1992. *Artificial Intelligence*. Addison-Wesley, 3 edn. ISBN 978-0-201-53377-4. (cited on page 118)
- WU, Q.; SHEN, C.; LIU, L.; DICK, A.; AND VAN DEN HENGEL, A., 2016a. What value do explicit high level concepts have in vision to language problems? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 10 and 79)
- WU, Q.; TENEY, D.; WANG, P.; SHEN, C.; DICK, A.; AND VAN DEN HENGEL, A., 2017. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, (2017). (cited on page 11)
- WU, Q.; WANG, P.; SHEN, C.; DICK, A.; AND VAN DEN HENGEL, A., 2016b. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 11)
- WU, Y.; WU, Y.; GKIOXARI, G.; AND TIAN, Y., 2018. Building generalizable agents with a realistic and rich 3d environment. *arXiv preprint arXiv:1801.02209*, (2018). (cited on page 98)
- XU, H. AND SAENKO, K., 2016. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *Proceedings of the European Conference on Computer Vision (ECCV)*. (cited on pages 51 and 53)

- XU, K.; BA, J.; KIROS, R.; CHO, K.; COURVILLE, A. C.; SALAKHUTDINOV, R.; ZEMEL, R. S.; AND BENGIO, Y., 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning (ICML)*. (cited on pages 1, 3, 6, 10, 11, 51, 52, 53, and 58)
- YAN, C.; MISRA, D.; BENNETT, A.; WALSMAN, A.; BISK, Y.; AND ARTZI, Y., 2018. CHALET: Cornell house agent learning environment. *arXiv preprint arXiv:1801.07357*, (2018). (cited on page 98)
- YANG, Z.; HE, X.; GAO, J.; DENG, L.; AND SMOLA, A. J., 2016a. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 11, 51, 53, 55, and 110)
- YANG, Z.; YUAN, Y.; WU, Y.; SALAKHUTDINOV, R.; AND COHEN, W. W., 2016b. Re-view networks for caption generation. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on pages 51, 53, and 68)
- YAO, T.; PAN, Y.; LI, Y.; AND MEI, T., 2017a. Incorporating copying mechanism in image captioning for learning novel objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 79)
- YAO, T.; PAN, Y.; LI, Y.; QIU, Z.; AND MEI, T., 2017b. Boosting image captioning with attributes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 10 and 68)
- YOU, Q.; JIN, H.; WANG, Z.; FANG, C.; AND LUO, J., 2016. Image captioning with semantic attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 11)
- YOUNG, P.; LAI, A.; HODOSH, M.; AND HOCKENMAIER, J., 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2 (2014), 67–78. (cited on pages 11, 37, 40, and 91)
- ZAMIR, A. R.; SAX, A.; SHEN, W.; GUIBAS, L. J.; MALIK, J.; AND SAVARESE, S., 2018a. Taskonomy: Disentangling task transfer learning. *arXiv preprint arXiv:1804.08328*, (2018). (cited on page 122)
- ZAMIR, A. R.; XIA, F.; HE, J.; SAX, S.; MALIK, J.; AND SAVARESE, S., 2018b. Gibson Env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 98)

-
- ZEILER, M. D., 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, (2012). (cited on page 64)
- ZHANG, P.; GOYAL, Y.; SUMMERS-STAY, D.; BATRA, D.; AND PARIKH, D., 2016a. Yin and yang: Balancing and answering binary visual questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 24)
- ZHANG, Y.; GONG, B.; AND SHAH, M., 2016b. Fast zero-shot image tagging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 77)
- ZHU, Y.; GROTH, O.; BERNSTEIN, M.; AND FEI-FEI, L., 2016. Visual7W: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 3, 6, 16, 51, 52, and 53)
- ZHU, Y.; MOTTAGHI, R.; KOLVE, E.; LIM, J. J.; GUPTA, A.; FEI-FEI, L.; AND FARHADI, A., 2017. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. (cited on pages 4, 96, 99, 103, and 109)
- ZHU, Y.; ZHANG, C.; RÉ, C.; AND FEI-FEI, L., 2015. Building a Large-scale Multimodal Knowledge Base System for Answering Visual Queries. In *arXiv preprint arXiv:1507.05670*. (cited on page 11)
- ZITNICK, L. AND DOLLÁR, P., 2014. Edge boxes: Locating object proposals from edges. In *Proceedings of the European Conference on Computer Vision (ECCV)*. (cited on page 55)