Grounding Language by Seeing, Hearing, and Interacting

Rowan Zellers

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

Doctor of Philosophy

University of Washington

2022

Reading Committee:
Yejin Choi, Chair
Ali Farhadi, Chair
Oren Etzioni

Program Authorized to Offer Degree:
Department of Computer Science & Engineering

University of Washington

Abstract

Grounding Language by Seeing, Hearing, and Interacting

Rowan Zellers

Co-Chairs of the Supervisory Committee:
Yejin Choi
Department of Computer Science & Engineering

Ali Farhadi
Department of Computer Science & Engineering

As humans, our understanding of language is grounded in a rich mental model about "how the world works." As children, we learn this mental model gradually. We take in raw perceptual input about the world through all of our senses, and learn to make sense of people and objects around us – enough to take action in the world. Our understanding of language and vision is *grounded* in the world.

Deep learning has made significant progress in recent years, for a variety of AI problems. Yet today's state-of-the-art models in natural language processing (NLP) and computer vision (CV) are ungrounded. They learn exclusively from text-only, or text-annotated data on the internet, making it harder for them to connect language and vision to the world beyond those modalities.

In this thesis, I will present a few lines of work to bridge this gap between machines and humans. I will first discuss how we might measure grounded understanding. I will introduce a suite of approaches for constructing benchmarks, using machines in the loop to filter out spurious biases. These include benchmarking grounding through exams about written text alone, through visual scenes, as well as through interacting with humans. Then, I will introduce PIGLeT: a model that learns physical commonsense understanding by interacting with the world through simulation, using this knowledge to ground language. PIGLeT learns linguistic form and meaning – together

– and outperforms text-to-text only models that are orders of magnitude larger. Finally, I will introduce MERLOT, which learns about situations in the world by watching millions of YouTube videos with transcribed speech. MERLOT is trained to jointly represent video, audio, and language, together and over time – learning multimodal and neural script knowledge representations.

TABLE OF CONTENTS

Page

List of Figures iv

List of Tables x

Chapter 1: Introduction 1

Chapter 2: Benchmarking Groundedness through Text Alone, Pt1: SWAG 3
2.1 Introduction 3
2.2 SWAG: Our new dataset 5
2.3 A solution to annotation artifacts 6
2.4 Experiments 11
2.5 Analysis 15
2.6 Related Work 17
2.7 Conclusion 19

Chapter 3: Benchmarking Groundedness through Text Alone, Pt2: HellaSWAG 22
3.1 Meta-Introduction 22
3.2 Introduction 22
3.3 Investigating SWAG 25
3.4 HELLASWAG 28
3.5 Results 31
3.6 Discussion 35
3.7 Conclusion 40

Chapter 4: Benchmarking Groundedness through Text and Images 41
4.1 Introduction 41
4.2 Task Overview 44

4.3 Data Collection . . . 46
4.4 Adversarial Matching . . . 47
4.5 Recognition to Cognition Networks . . . 49
4.6 Results . . . 51
4.7 Related Work . . . 56
4.8 Conclusion . . . 57

Chapter 5: Benchmarking Groundedness beyond Examinations . . . 59
5.1 Introduction . . . 59
5.2 Real World Language Use . . . 62
5.3 TURINGADVICE: a New Challenge for Natural Language Understanding . . . 65
5.4 Experimental Results on REDDITADVICE . . . 69
5.5 Analysis and Discussion . . . 72
5.6 Conclusion; Ethical Considerations . . . 75

Chapter 6: Language Grounding through Interaction in a 3D World . . . 77
6.1 Introduction . . . 77
6.2 PIGPEN: A Resource for Neuro-Symbolic Language Grounding . . . 80
6.3 Modeling PIGLET . . . 82
6.4 Experiments . . . 86
6.5 Analysis . . . 91
6.6 Related Work . . . 93
6.7 Conclusion . . . 94

Chapter 7: Language-and-Vision Neural Script Knowledge Models, (Merlot) . . . 95
7.1 Introduction . . . 95
7.2 Related Work . . . 97
7.3 MERLOT: Multimodal Event Representation Learning Over Time . . . 99
7.4 Experiments: Transferring MERLOT to Downstream Tasks . . . 104
7.5 Conclusion, Limitations, and Broader Impacts . . . 111

Chapter 8: Language-and-Vision-and-Sound Neural Script Knowledge Models, (Merlot Reserve) . . . 114
8.1 Introduction . . . 114

8.2 Related Work 116
8.3 Model: RESERVE 118
8.4 Experiments 123
8.5 Qualitative Analysis: Why does audio help? 133
8.6 Conclusion, Limitations, Broader Impact 134

Chapter 9: Conclusion and Future Work 135

Bibliography 138

LIST OF FIGURES

Figure Number Page

2.1 Overview of the data collection process. For a pair of sequential video captions, the second caption is split into noun and verb phrases. A language model generates many negative endings, of which a difficult subset are human-annotated. 6

2.2 Test accuracy by AF iteration, under the negatives given by $\mathcal{A}$. The accuracy drops from around 60% to close to random chance. For efficiency, the first 100 iterations only use the MLP. 9

2.3 Mechanical Turk instructions (abridged). 11

2.4 Top: Distribution of the 40 top verbs in the union of SNLI and SWAG. Our dataset shows a greater variety of dynamic verbs, such as "move", as well as temporal verbs such as "start" and "come." "Continue" is cut off for SNLI (it has frequency $6 \cdot 10^{-5}$). Bottom: CDF for verbs in SNLI and SWAG. 15

3.1 Models like BERT struggle to finish the sentences in HELLASWAG, even when they come from the same distribution as the training set. While the wrong endings are on-topic, with words that relate to the context, humans consistently judge their meanings to be either incorrect or implausible. For example, option **A** of the WikiHow passage suggests that a driver should stop at a red light for **no more than two seconds**. 23

3.2 Validation accuracy on SWAG for BERT-Large versus training set size. The baseline (25% accuracy) is random chance. BERT does well given as few as 16 training examples, but requires tens of thousands of examples to approach human performance. 25

3.4 Adversarial Filtering (AF) results with BERT-Large as the discriminator. **Left**: AF applied to ActivityNet generations produced by Zellers et al. [303]'s language model versus OpenAI GPT. While GPT converges at random, the LM used for SWAG converges at 75%. **Right**: AF applied to WikiHow generations from GPT, while varying the ending length from one to three sentences. They converge to random, ∼40%, and ∼50%, respectively. 27

3.5 For HELLASWAG, we ensure high human agreement through several rounds of annotation. By collecting how likely each ending is we can filter false negative endings – machine generations that sound realistic – and replace them with true negatives. On both subdatasets, BERT performance increases during validation, but the gap to human performance remains wide. 30

3.6 Lengths of ActivityNet and WikiHow; the latter with two-sentence generations. WikiHow is much longer, which corresponds to being easier for humans, while taking longer for AF to converge. 30

3.7 Examples on the in-domain validation set of HELLASWAG, grouped by category label. Our evaluation setup equally weights performance on categories seen during training as well as out-of-domain. 32

3.8 Transfer experiments from SWAG to HELLASWAG and vice versa, evaluated on the validation sets. Overall, a BERT-Large that is trained on SWAG hardly generalizes to HELLASWAG: it scores 34.6%. 34

3.9 Example questions answered by BERT-Large. Correct model predictions are **blue**, incorrect predictions are red. The right answers are **bolded**. 36

3.10 Performance on the WikiHow subset of alternative variations of HELLASWAG, where different Adversarial Filters are used (but without human validation). We consider the shallow stylistic adversaries used by Zellers et al. [303] (Stylistic Ensemble), as well as an LSTM with ELMo embeddings, GPT, BERT-Base, and BERT-Large. For each adversarial filtering model, we record the accuracy of that model before and after AF is used. We also evaluate each alternative dataset using BERT-Large. The results suggest that using a a stronger model at test time (over the model used for AF) improves performance, but is not enough to solve the task. 37

3.11 Estimated pretraining hours required to reach a desired accuracy on HELLASWAG. We estimate perfomance with respect to a RTX 2080 Ti - a modern, fast GPU, and fit a log-linear regression line. An extrapolation suggests that to reach human-level performance on HELLASWAG, without algorithmic or computational improvements, would require 10^9 GPU-hours of pretraining (over 100k GPU years). 38

4.1 VCR: Given an image, a list of regions, and a question, a model must answer the question and provide a *rationale* explaining why its answer is right. Our questions challenge computer vision systems to go beyond recognition-level understanding, towards a higher-order cognitive and commonsense understanding of the world depicted by the image. 41

4.2 Overview of the types of inference required by questions in VCR. Of note, 38% of the questions are explanatory 'why' or 'how' questions, 24% involve cognition-level activities, and 13% require temporal reasoning (i.e., what might come next). These categories are not mutually exclusive; an answer might require several hops of different types of inferences. 44

4.3 An overview of the construction of VCR. Using a state-of-the-art object detector [109, 86], we identify the objects in each image. The most interesting images are passed to crowd workers, along with scene-level context in the form of scene descriptions (MovieClips) and video captions (LSMDC, [224]). The crowd workers use a combination of natural language and detection tags to ask and answer challenging visual questions, also providing a rationale justifying their answer. . . 46

4.4 Overview of Adversarial Matching. Incorrect choices are obtained via maximum-weight bipartite matching between queries and responses; the weights are scores from state-of-the-art natural language inference models. Assigned responses are highly relevant to the query, while they differ in meaning versus the correct responses. 48

4.5 High-level overview of our model, **R2C**. We break the challenge of Visual Commonsense Reasoning into three components: grounding the query and response, contextualizing the response within the context of the query and the entire image, and performing additional reasoning steps on top of this rich representation. 49

4.6 Qualitative examples from **R2C**. Correct predictions are highlighted in blue. Incorrect predictions are in red with the correct choices **bolded**. For more predictions, see see `visualcommonsense.com/explore`. 55

5.1 TURINGADVICE. Humans are natural experts at *using* language to successfully address situations that arise, such as giving advice. We introduce a new framework, dataset, and leaderboard to generatively evaluate real-world language use. Today's most powerful models – which obtain near-human or superhuman performance on core NLP benchmarks for reading comprehension, natural language inference, and commonsense reasoning – struggle with all of these capabilities when generating advice, as highlighted in red. 60

5.2 Crowdsourcing workflow. Mechanical Turk Workers are given a situation, and two pieces of advice. First, they choose which is more helpful (here, B). Second, they rate the helpfulness of the worse advice (A); last, they answer a diagnostic question. 68

5.3 Helpfulness of models relative to top-scoring Reddit advice. We show results over 200 shared situations; we also show bootstrapped 95% confidence intervals. Advice from the best-scoring model, T5-11B, is preferred 14.5% over top-scoring Reddit advice. We also compare the second-top scoring piece of Reddit advice, which scores 41% – worse than the best advice (50% by definition), but better than any model. 70

5.4 Improvement (in absolute percentage $\%$) between pairs of models, along with statistical significance from a paired t-test. The improvement of T5-11B over smaller models like Grover-Mega is highly statistically significant (10% gap, $p<.01$), while being far worse than human performance. Our evaluation thus meaningfully grades varying levels of performance. 70

5.5 Qualitative example. Though machine-generated advice matches keywords from the situation, it is frequently not helpful or even self-contradictory. The issues are due to critical errors in natural language understanding, such as reading comprehension, entailment, and coreference. 73

5.6 Distribution of ratings for three models: TF-IDF retrieval, GPT3, and T5, along with ratings for the second-best rated Reddit advice. Though deep generators like GPT3 and T5 are often preferred over the retrieval baseline, they also often write advice that would never be helpful (33% GPT3, 13% T5), and that is racist, sexist, or otherwise dangerous (10% GPT3, 3% T5). 74

6.1 PIGLET. Through physical interaction in a 3D world, we learn a model for what actions do to objects. We use our physical model as an interface for a language model, jointly modeling elements of language *form* and *meaning*. Given an action expressed symbolically or in English, PIGLET can simulate what might happen next, expressing it symbolically or in English. 78

6.2 PIGPEN, a setting for few-shot language-world grounding. We collect data for 280k physical interactions in THOR, a 3D simulator with 20 actions and 125 object types, each with 42 attributes (e.g. `isBroken`). We annotate 2k interactions with English sentences describing the initial world state, the action, and the action result. 79

6.3 PIGLET architecture. We pretrain a model of physical world dynamics by learning to transform objects $\vec{\boldsymbol{o}}$ and actions $\boldsymbol{a}$ into new updated objects $\vec{\boldsymbol{o}}'$. Our underlying world dynamics model – the encoder, the decoder, and the action application module, can augment a language model with grounded commonsense knowledge. . . . 83

6.4 Qualitative examples. Our model PIGLET reliably predicts what might happen next (like the Mug becoming empty in Row 1), in a structured and explicit way. However, it often struggles at generating sentences for unseen objects like Mug that are excluded from the training set. T5 struggles to predict these changes, for example, it seems to suggest that emptying the Mug causes all containers in the scene to become empty. 91

6.5 PIGPEN-NLU performance of a **zero-shot** PIGLET, that was pretrained on Books and Wikipedia without reading any words of our 'unseen' objects like 'mug.' It outperforms a much bigger T5-11B overall, though is in turn beaten by PIGLET on unseen objects like 'Sink' and 'Microwave.' 92

7.1 Multimodal Event Representation Learning Over Time. We learn representations of multimodal script knowledge from 6 million YouTube videos. These representations can then be applied to a variety of downstream tasks that require commonsense or temporal visual reasoning. 95

7.2 **Left**: MERLOT learns to match contextualized captions with their corresponding video frames. **Right**: the same image encoding is provided, along with (masked) word embeddings, into a joint vision-language Transformer model; it then unmasks ground words (like 'saw' in this example) and puts scrambled video frames into the correct order. 100

7.3 Zero-shot story ordering (same setup as Table 7.2). MERLOT performs temporal commonsense reasoning accross frames. In the first row, it uses 'the old man' mentioned to identify the 'kids' as parent-aged; in the second, it identifies riding a merry-go-round as an activity that takes a while. 110

8.1 MERLOT RESERVE learns *multimodal neural script knowledge* representations of video – jointly reasoning over video frames, text, and audio. Our model is pretrained to predict which snippet of text (and audio) might be hidden by the MASK. This task enables it to perform well on a variety of vision-and-language tasks, in both zero-shot and finetuned settings. 114

8.2 RESERVE architecture. We provide sequence-level representations of video frames, and *either* words or audio, to a joint encoder. The joint encoder contextualizes over modalities and timesteps, to predict what is behind MASK for audio $\widehat{\mathbf{a}}_t$ and text $\widehat{\mathbf{w}}_t$. We supervise these predictions with independently encoded targets: $\mathbf{a}_t$ from the audio encoder, and $\mathbf{w}_t$ from a separate text encoder (not shown). 119

8.3 Contrastive span training. Given a video with all modalities temporally aligned, we MASK out a region of text and audio. The model must maximize its similarity *only to* an independent encoding of the text $\mathbf{w}_t$ and audio $\mathbf{a}_t$. 121

8.4 **Pretraining progress**: performance on contrastive-span pretraining, vs. finetuned VCR validation accuracy. Pretraining RESERVE-B for 9 more epochs boosts performance by 5%; L by 8%. 126

8.5 **Exploring MASKed audio self-supervision**. Shown are example videos from our validation set, with predictions from RESERVE-B. During pretraining, our model progressively learns to pick up on audio-specific clues. It seems to recognize physical dynamics of *cooking popcorn*, matching the first row to its MASKed audio. Likewise, it seems to use social reasoning to match the second row to its audio. Both of these clues are orthogonal to what the subtitles provide. 132

LIST OF TABLES

Table Number Page

2.1 Examples from SWAG; the correct answer is **bolded**. Adversarial Filtering ensures that stylistic models find all options equally appealing. 4

2.2 Annotators tend to label the found ending as likely and within the top 2 (column 2), in other cases the example is filtered out. Both label groups have high inter-annotator agreement, in terms of Krippendorff's α and pairwise percent agreement. 12

2.3 Performance of all models in accuracy (%). All models substantially underperform humans, although performance increases as more context is provided (left to right). We optionally train on found endings only, or found and human-validated generated endings (found+gen). 20

2.4 Justifications for ranking the gold answer over a wrong answer chosen by ESIM+ELMo. 21

2.5 Example questions answered by the best model, ESIM+Elmo, sorted by model probability. Correct model predictions are in **blue**, incorrect model predictions are red. The right answers are **bolded**. 21

3.1 Performance of models, evaluated with accuracy (%).We report results on the full validation and test sets (Overall), as well as results on informative subsets of the data: evaluated on in-domain, versus zero-shot situations, along with performance on the underlying data sources (ActivityNet versus WikiHow). All models substantially underperform humans: the gap is over 45% on in-domain categories, and 50% on zero-shot categories. 33

4.1 Experimental results on VCR. VQA models struggle on both question-answering ($Q \rightarrow A$) as well as answer justification ($Q \rightarrow AR$), possibly due to the complex language and diversity of examples in the dataset. While language-only models perform well, our model **R2C** obtains a significant performance boost. Still, all models underperform human accuracy at this task. For more up-to-date results, see the leaderboard at https://visualcommonsense.com/leaderboard. 52

4.2 Ablations for **R2C**, over the validation set. 'No query' tests the importance of integrating the query during contextualization; removing this reduces $Q \rightarrow AR$ performance by 20%. In 'no reasoning', the LSTM in the reasoning stage is removed; this hurts performance by roughly 1%. Removing the visual features during grounding, or using GloVe embeddings rather than BERT, lowers performance significantly, by 10% and 25% respectively. 54

6.1 **Overall results**. **Left**: we show the model accuracies at predicting all attributes of an object correctly. We compare PIGLET with 'text-to-text' approaches that represent the object states as a string, along with BERT-style approaches with additional machinery to encode inputs or decode outputs. PIGLET outperforms a T5 model 100x its size (11B params) and shows gains over the BERT-style models that also model action dynamics through a language transformer. **Right**: we show several attribute-level accuracies, along with the number of categories per attribute; PIGLET outperforms baselines by over 4 points for some attributes such as size and distance. 87

6.2 Ablation study on PIGPEN-NLU's validation set. Our model improves 6% by modeling global dynamics of all objects in the scene, versus applying actions to single objects in isolation. We improve another 3% by adding an auxiliary generation loss. 89

6.3 Text generation results on PIGPEN-NLG, showing models of roughly equivalent size (up to 117M parameters). Our PIGLET outperforms the LM baseline (using the same architecture but omitting the physical reasoning component) by 4 BLEU points, 2 BERTScore F_1 points, and 0.35 points in a human evaluation of language faithfulness to the actual scene. 90

7.1 Results on VCR [305]. We compare against SOTA models of the same '`base`' size as ours (12-layer vision-and-language Transformers). MERLOT performs best on all metrics. 105

7.2 Results unscrambling SIND visual stories[126, 4]. Captions are provided in the correct order; models must arrange the images temporally. MERLOT performs best on all metrics by reasoning over the entire story, instead of independently matching images with captions. 106

7.3 Comparison with state-of-the-art methods on video reasoning tasks. MERLOT outperforms state of the art methods in **12** downstream tasks that involve short and long videos. 107

7.4 Ablation study on the validation set of VCR question answering ($Q \rightarrow A$) and TVQA+, in accuraty (%). We put a 🍷 next to the configurations we chose for MERLOT. 108

8.1 **Ablation study** of our contrastive span objective. It outperforms prior work in a Vision+Text setting, with a 1% boost when audio is added. Our full setup, adding written text, improves another 1%. 🍷 denotes part of our full model. 125

8.2 RESERVE gets **state-of-the-art leaderboard performance on VCR**. We compare it with the largest submitted single models, including image-caption models that utilize heavy manual supervision (e.g. object detections and captions). 127

8.3 RESERVE gets state-of-the-art results on TVQA by **over 7%**, versus prior work (that cannot make use of audio). 128

8.4 RESERVE gets state-of-the-art results on Kinetics-600 by **1.5%** versus standard approaches (that cannot make use of audio). 129

8.5 Zero shot results. On STAR, RESERVE obtains state-of-the-art results, outperforming finetuned video models. It performs well on EPIC-Kitchens (verb and noun forecasting), along with LSMDC, despite their long-tail distributions. On MSR-VTT QA, it outperforms past work on weakly-supervised video QA. Further, it outperforms CLIP (that cannot handle dynamic situations), and benefits from audio when given. 130

ACKNOWLEDGMENTS

I'm done! There are so many people who have helped me along this journey.

First, thank you to my advisors Yejin Choi and Ali Farhadi. Over the years both of you truly shaped me into the researcher and the person I am today, at least the positive aspects. Both of you were there for me during all the ups and downs of the rollercoaster that was my PhD. I'm really lucky to have been your student.

Thanks to my committee members. Thanks to Oren Etzioni, for being so supportive throughout the 4 years I've been at AI2 (off and on). Zaid Harchaoui, thanks for your helpful and wise insights on my research.

Thanks to Luke Zettlemoyer for writing me a letter of recommendation, and for providing me advice over the years – not only about our field but also about the research process in general.

Thanks to all of the other mentors I've had over the years in grad school. In no particular order, this list includes Franzi Roesner, Kate Saenko, Hannaneh Hajishirzi, Noah Smith, Yonatan Bisk, Mark Yatskar, Roy Schwartz, Omer Levy, Roozbeh Mottaghi, Ani Kembhavi, Matt Peters, Ludwig Schmidt, and Ranjay Krishna. Thanks for providing me so much guidance during all the times I've been lost, confused, or stuck.

Thanks to my undergraduate research mentors. Jackie Dresch and Rob Drewell – if it weren't for you, I probably wouldn't have found my way into research in the first place. LP Morency – thanks for supporting me over not just one but two summers, for teaching me the ropes about how to do research, and for inspiring me to look at the intersection of language-and-vision in the first place.

Thanks to my labmates, friends, and colleagues. That list includes Eunsol Choi, Hannah Rashkin, Antoine Bosselut, Maarten Sap, Max Forbes, Ari Holtzman, Ximing Lu, Jeff Da, Jan

Buys, Rachel Rudinger, Yannis Konstas, Sean Welleck, Saadia Gabriel, Peter West, Karen Qin, James Park, Xiujun Li, Liwei Jiang, Alisa Liu, Gary Liu, Jillian Fisher, Melanie Sclar, Jaehun Jung; Kiana Ehsani, Daniel Gordon, Max Horton, Gabe Ilharco, Aditya Kusupati, Keunhong Park, Joe Redmon, Junha Roh, Minjoon Seo, Matt Wallingford, Aaron Walsman, Nancy Wang, Mitch Wortsman, Kuo-Hao Zeng, and Keivan Alizadeh. I've learned so much from you all over the years.

Thanks to my amazing AI2 collaborators who have made it such a vibrant place over the years. Some names that come to mind, but of course not a complete list by any means, would be Jack Hessel, Chandra Bhagavatula, Ronan Le Bras, Jena Hwang, Luca Weihs, Jiasen Lu, Daniel Khashabi, Raj Ammanabrolu, Swabha Swayamdipta, Nick Lourie, and Keisuke Sakaguchi. It's been amazing working with you all at AI2.

Thanks to all my friends in Seattle, and to the broader community who do some subset of indoor bouldering, snowboarding, acro-yoga, and hiking. It's fun doing those activities with you of course, but it's also been amazing to spend time with you all in general.

And last but certainly not least, thanks to Paula for being the best partner I could ask for throughout all of this!

Chapter 1

INTRODUCTION

Human intelligence is inherently grounded. As humans, when we are (for instance) perceiving the visual world, or reading language, we are at the same time interpreting these raw observations through a mental model of the world around us. This *commonsense* mental model lets us conceptualize everything from objects and atomic actions, to more complex (and abstract) situations. A goal of Artificial Intelligence is to likewise build agents that likewise perform human-level reasoning. To be able to do this – especially across modalities – requires grounding.

Especially during the last decade before this dissertation, the fields of natural language processing (NLP) and computer vision have made incredible progress on standard benchmarks. The high-level reason has been the development of deep-learning based approaches, that make heavy use of both compute, as well as easy-to-download data on the internet.

At the same time, there are also fundamental questions about how to test the groundedness of AI systems. Grounding is inherently difficult to define, or even to operationalize. Yet when it is operationalized through benchmarks, it can be possible to produce systems that narrowly pass the benchmarks while showing little generalization beyond what those benchmarks directly test.

In this thesis, we present work towards building AI systems with grounded understanding. First, we present a few proposals to *operationalize* grounding. In Chapters 2 and 3, we benchmark AI models' abilities to connect natural language event descriptions to a model about what might happen next in the world. In Chapter 4, we examine *visual commonsense reasoning* through a benchmark named VCR: anticipating what might be going on in the world beyond an image. Creating both of these benchmarks involved additional technical advancements in terms of 1) how to collect requisite data at scale, and 2) how to turn that raw data into exams that are not easily gameable by machines. Exams are easily reproducible, yet there are limits when we try to evaluate

grounding in terms of 'correctness': in Chapter 5 we present a benchmark for evaluating machines by how they *use language*.

Next, we study how to learn this kind of grounded understanding. In Chapter 6 we present PIGLeT, a model for connecting language to a model of the physical world. In Chapter 7, we present MERLOT, a model that learns multimodal event representations, fusing vision and language. We incorporate sound on top of this in Chapter 8. These modeling approaches make progress on some of the exams sketched out in the earlier part of this thesis, with MERLOT Reserve obtaining state-of-the-art results on VCR. While they are just a step towards grounding, we argue that they contain principles that should be used in future AI systems.

We conclude the dissertation by discussing a few future directions towards making AI systems more grounded.

Chapter 2

BENCHMARKING GROUNDEDNESS THROUGH TEXT ALONE, PT1: SWAG

This chapter contains material that was originally published in [303].

2.1 Introduction

When we read a story, we bring to it a large body of implicit knowledge about the physical world. For instance, given the context "on stage, a woman takes a seat at the piano," shown in Table 2.1, we can easily infer what the situation might *look* like: a woman is giving a piano performance, with a crowd watching her. We can furthermore infer her likely *next* action: she will most likely set her fingers on the piano keys and start playing.

This type of natural language inference requires commonsense reasoning, substantially broadening the scope of prior work that focused primarily on linguistic entailment [44]. Whereas the dominant entailment paradigm asks if two natural language sentences (the 'premise' and the 'hypothesis') describe the same set of possible worlds [54, 32], here we focus on whether a (multiple-choice) ending describes a possible (*future*) world that can be anticipated from the situation described in the premise, even when it is not strictly entailed. Making such inference necessitates a rich understanding about everyday physical situations, including object affordances [83] and frame semantics [17].

A first step toward grounded commonsense inference with today's deep learning machinery is to create a large-scale dataset. However, recent work has shown that human-written datasets are susceptible to *annotation artifacts*: unintended stylistic patterns that give out clues for the gold labels [103, 210]. As a result, models trained on such datasets with human biases run the risk of over-estimating the actual performance on the underlying task, and are vulnerable to adversarial or

On stage, a woman takes a seat at the piano. She
a) sits on a bench as her sister plays with the doll.
b) smiles with someone as the music plays.
c) is in the crowd, watching the dancers.
d) nervously sets her fingers on the keys.
A girl is going across a set of monkey bars. She
a) jumps up across the monkey bars.
b) struggles onto the monkey bars to grab her head.
c) gets to the end and stands on a wooden plank.
d) jumps up and does a back flip.
The woman is now blow drying the dog. The dog
a) is placed in the kennel next to a woman's feet.
b) washes her face with the shampoo.
c) walks into frame and walks towards the dog.
d) tried to cut her face, so she is trying to do something very close to her face.

Table 2.1: Examples from SWAG; the correct answer is **bolded**. Adversarial Filtering ensures that stylistic models find all options equally appealing.

out-of-domain examples [269, 87].

In this section of the thesis, we introduce Adversarial Filtering (AF), a new method to automatically detect and reduce stylistic artifacts. We use this method to construct SWAG: an adversarial dataset with 113k multiple-choice questions. We start with pairs of temporally adjacent video captions, each with a context and a follow-up event that we *know* is physically possible. We then use a state-of-the-art language model fine-tuned on this data to massively oversample a diverse set of possible negative sentence endings (or *counterfactuals*). Next, we filter these candidate endings aggressively and adversarially using a committee of trained models to obtain a population of de-

biased endings with similar stylistic features to the real ones. Finally, these filtered counterfactuals are validated by crowd workers to further ensure data quality.

Extensive empirical results demonstrate unique contributions of our dataset, complementing existing datasets for natural langauge inference (NLI) [32, 278] and commonsense reasoning [221, 193, 316]. **First**, our dataset poses a new challenge of grounded commonsense inference that is easy for humans (88%) while hard for current state-of-the-art NLI models ($<60\%$). **Second**, our proposed adversarial filtering methodology allows for cost-effective construction of a large-scale dataset while substantially reducing known annotation artifacts. The generality of adversarial filtering allows it to be applied to build future datasets, ensuring that they serve as reliable benchmarks.

2.2 SWAG*: Our new dataset*

We introduce a new dataset for studying physically grounded commonsense inference, called SWAG.[1] Our task is to predict which event is most likely to occur next in a video. More formally, a model is given a context $\boldsymbol{c} = (\boldsymbol{s}, \boldsymbol{n})$: a complete sentence $\boldsymbol{s}$ and a noun phrase $\boldsymbol{n}$ that begins a second sentence, as well as a list of possible verb phrase sentence endings $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_4\}$. See Figure 2.1 for an example triple $(\boldsymbol{s}, \boldsymbol{n}, \boldsymbol{v}_i)$. The model must then select the most appropriate verb phrase $\boldsymbol{v}_i \in \boldsymbol{V}$.

Overview Our corpus consists of 113k multiple choice questions (73k training, 20k validation, 20k test) and is derived from pairs of consecutive video captions from ActivityNet Captions [148, 111] and the Large Scale Movie Description Challenge (LSMDC; 224). The two datasets are slightly different in nature and allow us to achieve broader coverage: ActivityNet contains 20k YouTube clips containing one of 203 activity types (such as doing gymnastics or playing guitar); LSMDC consists of 128k movie captions (audio descriptions and scripts). For each pair of captions, we use a constituency parser [246] to split the second sentence into noun and verb phrases (Figure 2.1).[2] Each question has a human-verified gold ending and 3 distractors.

---

[1] Short for Situations with Adversarial Generations.

[2] We filter out sentences with rare tokens ($\leq$3 occurrences), that are short ($l \leq 5$), or that lack a verb phrase.

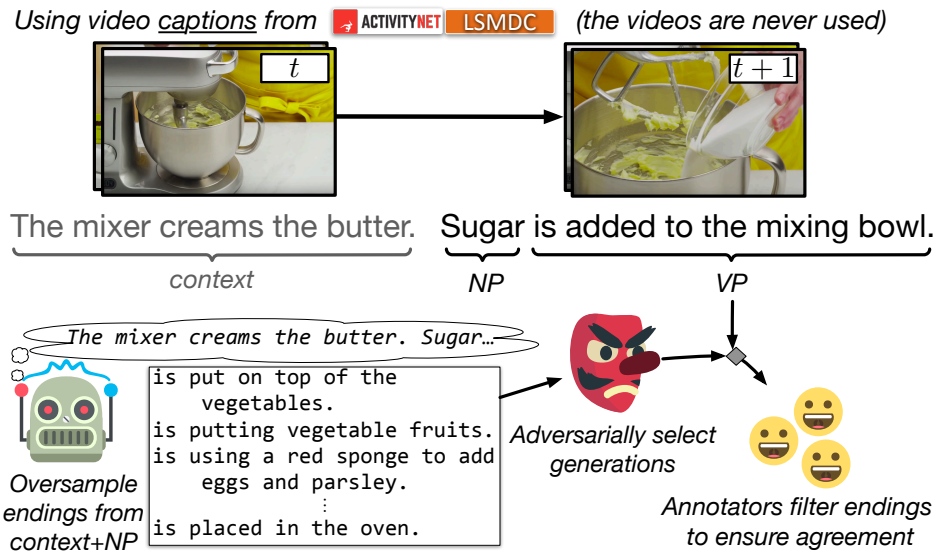


Figure 2.1: Overview of the data collection process. For a pair of sequential video captions, the second caption is split into noun and verb phrases. A language model generates many negative endings, of which a difficult subset are human-annotated.

2.3 A solution to annotation artifacts

In this section, we outline the construction of SWAG. We seek dataset diversity while minimizing *annotation artifacts*, conditional stylistic patterns such as length and word-preference biases. For many NLI datasets, these biases have been shown to allow shallow models (e.g. bag-of-words) obtain artificially high performance.

To avoid introducing easily “gamed" patterns, we present Adversarial Filtering (AF), a generally-applicable treatment involving the iterative refinement of a set of assignments to increase the entropy under a chosen model family. We then discuss how we generate counterfactual endings, and finally, the models used for filtering.

Algorithm 1 Adversarial filtering (AF) of negative samples. During our experiments, we set $N^{easy} = 2$ for refining a population of $N^- = 1023$ negative examples to $k = 9$, and used a 80%/20% train/test split.

while convergence not reached **do**

- Split the dataset $\mathcal{D}$ randomly up into training and testing portions $\mathcal{D}^{tr}$ and $\mathcal{D}^{te}$.
- Optimize a model f_θ on $\mathcal{D}^{tr}$.

for index i in $\mathcal{D}^{te}$ **do**

- Identify easy indices:

$\mathcal{A}_i^{easy} = \{j \in \mathcal{A}_i : f_\theta(x_i^+) > f_\theta(x_{i,j}^-)\}$

- Replace N^{easy} easy indices $j \in \mathcal{A}_i^{easy}$ with adversarial indices $k \notin \mathcal{A}_i$ satisfying $f_\theta(x_{i,k}^-) > f_\theta(x_{i,j}^-)$.

end for

end while

2.3.1 Formal definition

In this section, we formalize what it means for a dataset to be *adversarial*. Intuitively, we say that an adversarial dataset for a model f is one on which f will not generalize, even if evaluated on test data from the same distribution. More formally, let our input space be $\mathcal{X}$ and the label space be $\mathcal{Y}$. Our trainable classifier f, taking parameters θ is defined as $f_\theta : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}$. Let our dataset of size N be defined as $\mathcal{D} = \{(x_i, y_i)\}_{1 \le i \le N}$, and let the loss function over the dataset be $L(f_\theta, \mathcal{D})$. We say that a dataset is *adversarial* with respect to f if we expect high empirical error I over all leave-one-out train/test splits [259]:

$$I(\mathcal{D}, f) = \frac{1}{N} \sum_{i=1}^{N} L(f_{\theta_i^\star}, \{(x_i, y_i)\}), \tag{2.1}$$

$$\text{where } \theta_i^\star = \underset{\theta}{\operatorname{argmin}}\, L(f_\theta, \mathcal{D} \setminus \{(x_i, y_i)\}), \tag{2.2}$$

with regularization terms omitted for simplicity.

2.3.2 *Adversarial filtering (AF) algorithm*

In this section, we outline an approach for generating an adversarial dataset $\mathcal{D}$, effectively maximizing empirical error I with respect to a family of trainable classifiers f. Without loss of generality, we consider the situation where we have N *contexts*, each associated with a single positive example $(x_i^+, 1) \in \mathcal{X} \times \mathcal{Y}$, and a large population of context-specific negative examples $(x_{i,j}^-, 0) \in \mathcal{X} \times \mathcal{Y}$, where $1 \leq j \leq N^-$ for each i. For instance, the negative examples could be incorrect relations in knowledge-base completion [244], or all words in a dictionary for a single-word cloze task [327].

Our goal will be to filter the population of negative examples for each instance i to a size of $k \ll N^-$. This will be captured by returning a set of *assignments* $\mathcal{A}$, where for each instance the assignment will be a k-subset $\mathcal{A}_i = [1 \dots N^-]^k$. The filtered dataset will then be:

$$\mathcal{D}^{AF} = \{(x_i, 1), \{(x_{i,j}^-, 0)\}_{j \in \mathcal{A}_i}\}_{1 \leq i \leq N} \tag{2.3}$$

Unfortunately, optimizing $I(\mathcal{D}^{AF}, f)$ is difficult as $\mathcal{A}$ is global and non-differentiable. To address this, we present Algorithm 1. On each iteration, we split the data into dummy 'train' and 'test' splits. We train a model f on the training portion and obtain parameters θ, then use the remaining test portion to reassign the indices of $\mathcal{A}$. For each context, we replace some number of 'easy' negatives in $\mathcal{A}$ that f_θ classifies correctly with 'adversarial' negatives outside of $\mathcal{A}$ that f_θ misclassifies.

This process can be thought of as increasing the overall entropy of the dataset: given a strong model f_θ that is compatible with a random subset of the data, we aim to ensure it cannot generalize to the held-out set. We repeat this for several iterations to reduce the generalization ability of the model family f over arbitrary train/test splits.

2.3.3 *Generating candidate endings*

To generate counterfactuals for SWAG, we use an LSTM [119] language model (LM), conditioned on contexts from video captions. We first pretrain on BookCorpus [325], then finetune on the video caption datasets. The architecture uses standard best practices and was validated on held-out perplexity of the video caption datasets; details are in the appendix. We use the LM to sample

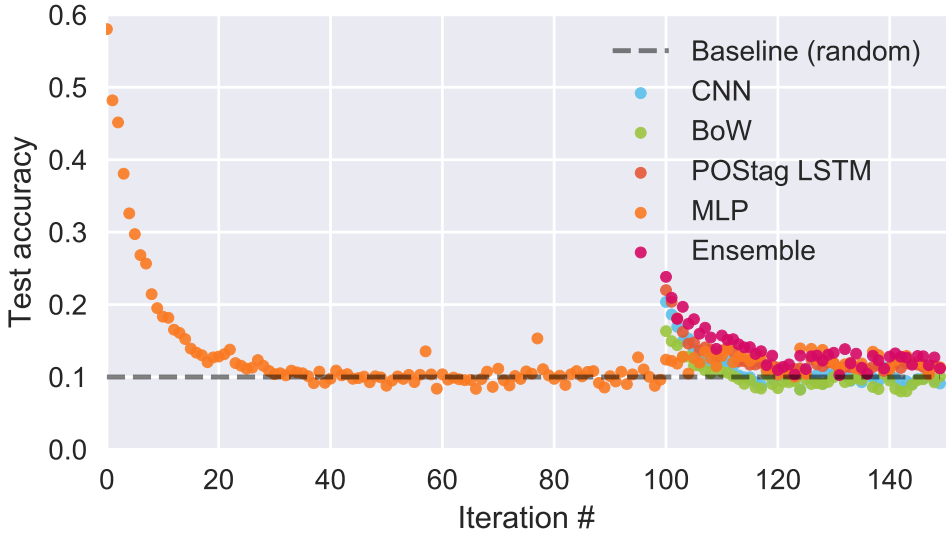


Figure 2.2: Test accuracy by AF iteration, under the negatives given by $\mathcal{A}$. The accuracy drops from around 60% to close to random chance. For efficiency, the first 100 iterations only use the MLP.

$N^- = 1023$ unique endings for a partial caption.[3]

Importantly, we *greedily* sample the endings, since beam search decoding biases the generated endings to be of lower perplexity (and thus easily distinguishable from found endings). We find this process gives good counterfactuals: the generated endings tend to use *topical* words, but often make little sense physically, making them perfect for our task. Further, the generated endings are marked as "gibberish" by humans only 9.1% of the time (Sec 2.3.5); in that case the ending is filtered out.

2.3.4 Stylistic models for adversarial filtering

In creating SWAG, we designed the model family f to pick up on low-level *stylistic features* that we posit should not be predictive of whether an event happens next in a video. These stylistic

[3]To ensure that the LM generates unique endings, we split the data into five validation folds and train five separate LMs, one for each set of training folds. This means that each LM never sees the found endings during training.

features are an obvious case of annotation artifacts [34, 231].[4] Our final classifier is an ensemble of four stylistic models:

1. A multilayer perceptron (MLP) given LM perplexity features and context/ending lengths.

2. A bag-of-words model that averages the word embeddings of the second sentence as features.

3. A one-layer CNN, with filter sizes ranging from 2-5, over the second sentence.

4. A bidirectional LSTM over the 100 most common words in the second sentence; uncommon words are replaced by their POS tags.

We ensemble the models by concatenating their final representations and passing it through an MLP. On every adversarial iteration, the ensemble is trained jointly to minimize cross-entropy.

The accuracies of these models (at each iteration, evaluated on a 20% split of the test dataset before indices of $\mathcal{A}$ get remapped) are shown in Figure 2.2. Performance decreases from 60% to close to random chance; moreover, confusing the perplexity-based MLP is not sufficient to lower performance of the ensemble. Only once the other stylistic models are added does the ensemble accuracy drop substantially, suggesting that our approach is effective at reducing stylistic artifacts.

2.3.5 Human verification

The final data-collection step is to have humans verify the data. Workers on Amazon Mechanical Turk were given the caption context, as well as six candidate endings: one found ending and five adversarially-sampled endings. The task was twofold: Turkers ranked the endings independently as likely, unlikely, or gibberish, and selected the best and second best endings (Fig 2.3).

We obtained the correct answers to each context in two ways. If a Turker ranks the found ending as either best or second best (73.7% of the time), we add the found ending as a gold example, with negatives from the generations not labelled best or gibberish. Further, if a Turker ranks a generated ending as best, and the found ending as second best, then we have reason to believe that the generation is good. This lets us add an additional training example, consisting of the generated

[4] A broad definition of annotation artifacts might include aspects besides lexical/stylistic features: for instance, certain events are less likely semantically regardless of the context (e.g. riding a horse using a hose). For this work, we erred more conservatively and only filtered based on style.

Imagine that you are watching a video clip. The clip has a caption, but it is missing the final phrase. Please choose the best 2 caption endings, and classify each as:

- **likely**, if it completes the caption in a reasonable way;
- **unlikely**, if it sounds ridiculous or impossible;
- **gibberish** if it has such serious errors that it doesn't feel like a valid English sentence.

Example: Someone is shown sitting on a fence and talking to the camera while pointing out horses. Someone

- stands in front of a podium. (**likely**, **second best**)
- rides a horse using a hose. (**unlikely**)
- is shown riding a horse. (**likely**, **best**)
- , the horse in a plaza field. (**gibberish**)

Figure 2.3: Mechanical Turk instructions (abridged).

best ending as the gold, and remaining generations as negatives.[5] Examples with ≤ 3 non-gibberish endings were filtered out.[6]

We found after 1000 examples that the annotators tended to have high agreement, also generally choosing found endings over generations (see Table 2.2). Thus, we collected the remaining 112k examples with one annotator each, periodically verifying that annotators preferred the found endings.

2.4 Experiments

In this section, we evaluate the performance of various NLI models on SWAG. Recall that models for our dataset take the following form: given a sentence and a noun phrase as context $\boldsymbol{c} = (\boldsymbol{s}, \boldsymbol{n})$,

[5]These two examples share contexts. To prevent biasing the test and validation sets, we didn't perform this procedure on answers from the evaluation sets' context.

[6]To be data-efficient, we reannotated filtered-out examples by replacing gibberish endings, as well as generations that outranked the found ending, with candidates from $\mathcal{A}$.

	Label distribution by ending type		Inter-annotator agreement	
Labels	Found end	Gen. end	α	ppa
Best	53.5%	9.3%		
Second Best	20.2%	15.9%	0.43	72%
Neither	26.3%	74.8%		
Likely	80.3%	33.3%		
Unlikely	19.0%	57.5%	0.39	64%
Gibberish	0.7%	9.1%		

Table 2.2: Annotators tend to label the found ending as likely and within the top 2 (column 2), in other cases the example is filtered out. Both label groups have high inter-annotator agreement, in terms of Krippendorff's α and pairwise percent agreement.

as well as a list of possible verb phrase endings $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_4\}$, a model f_θ must select a verb $\hat{i}$ that hopefully matches i_{gold}:

$$\hat{i} = \underset{i}{\operatorname{argmax}}\, f_\theta(\boldsymbol{s}, \boldsymbol{n}, \boldsymbol{v}_i) \tag{2.4}$$

To study the amount of bias in our dataset, we also consider models that take as input just the ending verb phrase $\boldsymbol{v}_i$, or the entire second sentence $(\boldsymbol{n}, \boldsymbol{v}_i)$. For our learned models, we train f by minimizing multi-class cross-entropy. We consider three different types of word representations: 300d GloVe vectors from Common Crawl [203], 300d Numberbatch vectors retrofitted using ConceptNet relations [245], and 1024d ELMo contextual representations that show improvement on a variety of NLP tasks, including standard NLI [205]. We follow the final dataset split (see Section 2.2) using two training approaches: training on the found data, and the found and highly-ranked generated data. See the appendix for more details.

2.4.1 Unary models

The following models predict labels from *a single span* of text as input; this could be the ending only, the second sentence only, or the full passage.

a. **fastText** [136]: This library models a single span of text as a bag of n-grams, and tries to predict the probability of an ending being correct or incorrect independently.[7]

b. **Pretrained sentence encoders** We consider two types of pretrained RNN sentence encoders, SkipThoughts [143] and InferSent [50]. SkipThoughts was trained by predicting adjacent sentences in book data, whereas InferSent was trained on supervised NLI data. For each second sentence (or just the ending), we feed the encoding into an MLP.

c. **LSTM sentence encoder** Given an arbitrary span of text, we run a two-layer BiLSTM over it. The final hidden states are then max-pooled to obtain a fixed-size representation, which is then used to predict the potential for that ending.

2.4.2 Binary models

The following models predict labels from *two spans* of text. We consider two possibilties for these models: using just the second sentence, where the two text spans are $\boldsymbol{n}, \boldsymbol{v}_i$, or using the context and the second sentence, in which case the spans are $\boldsymbol{s}, (\boldsymbol{n}, \boldsymbol{v}_i)$. The latter case includes many models developed for the NLI task.

d. **Dual Bag-of-Words** For this baseline, we treat each sentence as a bag-of-embeddings $(\mathbf{c}, \mathbf{v}_i)$. We model the probability of picking an ending i using a bilinear model: $\text{softmax}_i(\mathbf{c}\mathbf{W}\mathbf{v}_i^T)$.[8]

e. **Dual pretrained sentence encoders** Here, we obtain representations from SkipThoughts or InferSent for each span, and compute their pairwise compatibility using either 1) a bilinear model or 2) an MLP from their concatenated representations.

---

[7]The fastText model is trained using binary cross-entropy; at test time we extract the prediction by selecting the ending with the highest positive likelihood under the model.

[8]We also tried using an MLP, but got worse results.

f. **SNLI inference** Here, we consider two models that do well on SNLI [32]: Decomposable Attention [200] and ESIM [42]. We use pretrained versions of these models (with ELMo embeddings) on SNLI to obtain 3-way entailment, neutral, and contradiction probabilities for each example. We then train a log-linear model using these 3-way probabilities as features.

g. **SNLI models (retrained)** Here, we train ESIM and Decomposable Attention on our dataset: we simply change the output layer size to 1 (the potential of an ending $\boldsymbol{v}_i$) with a softmax over i.

2.4.3 Other models

We also considered the following models:

h. **Length**: Although length was used by the adversarial classifier, we want to verify that human validation didn't reintroduce a length bias. For this baseline, we always choose the shortest ending.

i. **ConceptNet** As our task requires world knowledge, we tried a rule-based system on top of the ConceptNet knowledge base [245]. For an ending sentence, we use the spaCy dependency parser to extract the head verb and its dependent object. The ending score is given by the number of ConceptNet causal relations[9] between synonyms of the verb and synonyms of the object.

j. **Human performance** To benchmark human performance, five Mechanical Turk workers were asked to answer 100 dataset questions, as did an 'expert' annotator (the first author of this paper). Predictions were combined using a majority vote.

2.4.4 Results

We present our results in Table 2.3. The best model that only uses the ending is the LSTM sequence model with ELMo embeddings, which obtains 43.6%. This model, as with most models studied, greatly improves with more context: by 3.1% when given the initial noun phrase, and by an additional 4% when also given the first sentence.

Further improvement is gained from models that compute pairwise representations of the in-

[9]We used the relations 'Causes', 'CapableOf', 'ReceivesAction', 'UsedFor', and 'HasSubevent'. Though their coverage is low (30.4% of questions have an answer with $\geq$1 causal relation), the more frequent relations in ConceptNet, such as 'IsA', at best only indirectly relate to our task.

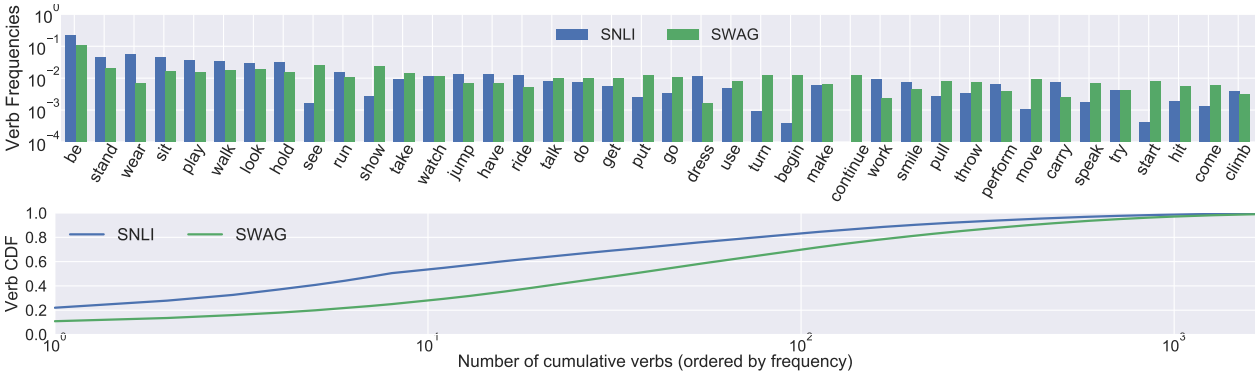


Figure 2.4: Top: Distribution of the 40 top verbs in the union of SNLI and SWAG. Our dataset shows a greater variety of dynamic verbs, such as "move", as well as temporal verbs such as "start" and "come." "Continue" is cut off for SNLI (it has frequency $6 \cdot 10^{-5}$). Bottom: CDF for verbs in SNLI and SWAG.

puts. While the simplest such model, DualBoW, obtains only 35.1% accuracy, combining InferSent sentence representations gives 40.5% accuracy (InferSent-Bilinear). The best results come from pairwise NLI models: when fully trained on SWAG, ESIM+ELMo obtains 59.2% accuracy.

When comparing machine results to human results, we see there exists a lot of headroom. Though there likely is some noise in the task, our results suggest that humans (even untrained) converge to a consensus. Our in-house "expert" annotator is outperformed by an ensemble of 5 Turk workers (with 88% accuracy); thus, the effective upper bound on our dataset is likely even higher.

2.5 Analysis

2.5.1 SWAG *versus existing NLI datasets*

The past few years have yielded great advances in NLI and representation learning, due to the availability of large datasets like SNLI and MultiNLI [32, 278]. With the release of SWAG, we hope to continue this trend, particularly as our dataset largely has the same input/output format as other NLI datasets. We observe three key differences between our dataset and others in this space:

First, as noted in Section 2.1, SWAG requires a unique type of temporal reasoning. A state-of-the-art NLI model such as ESIM, when bottlenecked through the SNLI notion of entailment (SNLI-ESIM), only obtains 36.1% accuracy.[10] This implies that these datasets necessitate different (and complementary) forms of reasoning.

Second, our use of videos results in wide coverage of dynamic and temporal situations Compared with SNLI, with contexts from Flickr30K [209] image captions, SWAG has more active verbs like 'pull' and 'hit,' and fewer static verbs like 'sit' and 'wear' (Figure 2.4).[11]

Third, our dataset suffers from few lexical biases. Whereas fastText, a bag of n-gram model, obtains 67.0% accuracy on SNLI versus a 34.3% baseline [103], fastText obtains only 29.0% accuracy on SWAG.[12]

2.5.2 Error analysis

We sought to quantify how human judgments differ from the best studied model, ESIM+ELMo. We randomly sampled 100 validation questions that ESIM+ELMo answered incorrectly, for each extracting both the gold ending and the model's preferred ending. We asked 5 Amazon Mechanical Turk workers to pick the better ending (of which they preferred the gold endings 94% of the time) and to select one (or more) multiple choice reasons explaining why the chosen answer was better.

The options, and the frequencies, are outlined in Table 2.4. The most common reason for the turkers preferring the correct answer is situational (52.3% of the time), followed by weirdness (17.5%) and plausibility (14.4%). This suggests that ESIM+ELMo already does a good job at filtering out weird and implausible answers, with the main bottleneck being grounded physical understanding. The ambiguous percentage is also relatively low (12.0%), implying significant headroom.

---

[10] The weights of SNLI-ESIM pick up primarily on entailment probability (0.59), as with neutral (0.46), while contradiction is negatively correlated (-.42).

[11] Video data has other language differences; notably, character names in LSMDC were replaced by 'someone'

[12] The most predictive individual words on SWAG are infrequent in number: 'dotted' with P($+$|dotted) $= 77\%$ with 10.3 counts, and P($-$|similar) $= 81\%$ with 16.3 counts. (Counts from negative endings were discounted 3x, as there are 3 times as many negative endings as positive endings).

2.5.3 Qualitative examples

Last, we show several qualitative examples in Table 2.5. Though models can do decently well by identifying complex alignment patterns between the two sentences (e.g. being "up a tree" implies that "tree" is the end phrase), the incorrect model predictions suggest this strategy is insufficient. For instance, answering "An old man rides a small bumper car" requires knowledge about *bumper cars* and how they differ from regular cars: bumper cars are tiny, don't drive on roads, and don't work in parking lots, eliminating the alternatives. However, this knowledge is difficult to extract from existing corpora: for instance, the ConceptNet entry for Bumper Car has only a single relation: bumper cars are a type of vehicle. Other questions require intuitive physical reasoning: e.g, for "he pours the raw egg batter into the pan," about what happens next in making an omelet.

2.5.4 Where to go next?

Our results suggest that SWAG is a challenging testbed for NLI models. However, the adversarial models used to filter the dataset are purely stylistic and focus on the second sentence; thus, subtle artifacts still likely remain in our dataset. These patterns are ostensibly picked up by the NLI models (particularly when using ELMo features), but the large gap between machine and human performance suggests that more is required to solve the dataset. As models are developed for commonsense inference, and more broadly as the field of NLP advances, we note that AF can be used again to create a more adversarial version of SWAG using better language models and AF models.

2.6 Related Work

Entailment NLI There has been a long history of NLI benchmarks focusing on linguistic entailment [51, 54, 185, 32, 155, 278]. Recent NLI datasets in particular have supported learning broadly-applicable sentence representations [50]; moreover, models trained on these datasets were used as components for performing better video captioning [201], summarization [202], and generation [120], confirming the importance of NLI research. The NLI task requires a variety of

commonsense knowledge [176], which our work complements. However, previous datasets for NLI have been challenged by unwanted annotation artifacts, [103, 210] or scale issues. Our work addresses these challenges by constructing a new NLI benchmark focused on grounded commonsense reasoning, and by introducing an adversarial filtering mechanism that substantially reduces known and easily detectable annotation artifacts.

Commonsense NLI Several datasets have been introduced to study NLI beyond linguistic entailment: for inferring likely causes and endings given a sentence (COPA; 221), for choosing the most sensible ending to a short story (RocStories; 193, 236), and for predicting likelihood of a hypothesis by regressing to an ordinal label (JOCI; [316]). These datasets are relatively small: 1k examples for COPA and 10k cloze examples for RocStories.[13] JOCI increases the scale by generating the hypotheses using a knowledge graph or a neural model. In contrast to JOCI where the task was formulated as a regression task on the degree of plausibility of the hypothesis, we frame commonsense inference as a multiple choice question to reduce the potential ambiguity in the labels and to allow for direct comparison between machines and humans. In addition, SWAG's use of adversarial filtering increases diversity of situations and counterfactual generation quality.

Last, another related task formulation is sentence completion or cloze, where the task is to predict a single word that is removed from a given context [327, 198].[14] Our work in contrast requires longer textual descriptions to reason about.

Vision datasets Several resources have been introduced to study temporal inference in vision. The Visual Madlibs dataset has 20k image captions about hypothetical next/previous events [294]; similar to our work, the test portion is multiple-choice, with counterfactual answers retrieved from similar images and verified by humans. The question of 'what will happen next?' has also been studied in photo albums [126], videos of team sports, [72] and egocentric dog videos [69]. Last, annotation artifacts are also a recurring problem for vision datasets such as Visual Genome [304] and Visual QA [129]; recent work was done to create a more challenging VQA dataset by annotating

[13]For RocStories, this was by design to encourage learning from the larger corpus of 98k sensible stories.

[14]Prior work on sentence completion filtered negatives with heuristics based on LM perplexities. We initially tried something similar, but found the result to still be gameable.

complementary image pairs [95].

Reducing gender/racial bias Prior work has sought to reduce demographic biases in word embeddings [314] as well as in image recognition models [319]. Our work has focused on producing a dataset with minimal annotation artifacts, which in turn helps to avoid some gender and racial biases that stem from elicitation [226]. However, it is not perfect in this regard, particularly due to biases in movies [230, 228]. Our methodology could potentially be extended to construct datasets free of (possibly intersectional) gender or racial bias.

Physical knowledge Prior work has studied learning grounded knowledge about objects and verbs: from knowledge bases [170], syntax parses [74], word embeddings [180], and images and dictionary definitions [301]. An alternate thread of work has been to learn scripts: high-level representations of event chains [229, 37]. SWAG evaluates both of these strands.

2.7 Conclusion

We propose a new challenge of physically situated commonsense inference that broadens the scope of natural language inference (NLI) with commonsense reasoning. To support research toward commonsense NLI, we create a large-scale dataset SWAGwith 113k multiple-choice questions. Our dataset is constructed using Adversarial Filtering (AF), a new paradigm for robust and cost-effective dataset construction that allows datasets to be constructed at scale while automatically reducing annotation artifacts that can be easily detected by a committee of strong baseline models. Our adversarial filtering paradigm is general, allowing potential applications to other datasets that require human composition of question answer pairs.

			Ending only				2nd sentence only				Context+2nd sentence			
			found only		found+gen		found only		found+gen		found only		found+gen	
		Model	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test
	misc	Random	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
	misc	Length	26.7	27.0	26.7	27.0								
	misc	ConceptNet					26.0	26.0	26.0	26.0				
Unary models	Sentence encoders	fastText	27.5	26.9	29.9	29.0	29.2	27.8	29.8	29.0	29.4	28.0	30.3	29.8
Unary models	Sentence encoders	SkipThoughts	32.4	32.1	32.2	31.8	33.0	32.4	32.8	32.3				
Unary models	Sentence encoders	InferSent	30.6	30.2	32.0	31.9	33.2	32.0	34.0	32.6				
Unary models	LSTM sequence model	LSTM+GloVe	31.9	31.8	32.9	32.4	32.7	32.4	34.3	33.5	43.1	43.6	45.6	45.7
Unary models	LSTM sequence model	LSTM+Numberbatch	32.4	32.6	32.3	31.9	31.9	31.9	34.1	32.8	39.9	40.2	41.2	40.5
Unary models	LSTM sequence model	LSTM+ELMo	**43.6**	**42.9**	**43.3**	**42.3**	**47.4**	**46.7**	**46.3**	**46.0**	51.4	50.6	51.3	50.4
Binary models	DualBoW	DualBoW+GloVe					31.3	31.3	31.9	31.2	34.5	34.7	32.9	33.1
Binary models	DualBoW	DualBoW+Numberbatch					31.9	31.4	31.6	31.3	35.1	35.1	34.2	34.1
Binary models	Dual sentence encoders	SkipThoughts-MLP					34.6	33.9	36.2	35.5	33.4	32.3	37.4	36.4
Binary models	Dual sentence encoders	SkipThoughts-Bilinear					36.0	35.7	34.7	34.5	36.5	35.6	35.3	34.9
Binary models	Dual sentence encoders	InferSent-MLP					32.9	32.1	32.8	32.7	35.9	36.2	39.5	39.4
Binary models	Dual sentence encoders	InferSent-Bilinear					32.0	31.3	31.6	31.3	40.5	40.3	39.0	38.4
Binary models	SNLI inference	SNLI-ESIM									36.4	36.1	36.2	36.0
Binary models	SNLI inference	SNLI-DecompAttn									35.8	35.8	35.8	35.7
Binary models	SNLI models (re-trained)	DecompAttn+GloVe					29.8	30.3	31.1	31.7	47.4	47.6	48.5	48.6
Binary models	SNLI models (re-trained)	DecompAttn+Numberbatch					32.4	31.7	32.5	31.9	47.4	48.0	48.0	48.3
Binary models	SNLI models (re-trained)	DecompAttn+ELMo					43.4	43.4	40.6	40.3	47.7	47.3	46.0	45.4
Binary models	SNLI models (re-trained)	ESIM+GloVe					34.8	35.1	36.3	36.7	51.9	52.7	52.5	52.5
Binary models	SNLI models (re-trained)	ESIM+Numberbatch					33.1	32.6	33.0	32.4	46.5	46.4	44.0	44.6
Binary models	SNLI models (re-trained)	ESIM+ELMo					46.0	45.7	45.9	44.8	**59.1**	**59.2**	**58.7**	**58.5**
	Human	1 turker									82.8			
	Human	3 turkers									85.1			
	Human	5 turkers									**88.0**			
	Human	Expert									85.0			

Table 2.3: Performance of all models in accuracy (%). All models substantially underperform humans, although performance increases as more context is provided (left to right). We optionally train on found endings only, or found and human-validated generated endings (found+gen).

Reason	Explanation	Freq.
Situational	The good ending is better *in context*.	53.7%
Plausibility	The bad ending is implausible *regardless of context*.	14.4%
Novelty	The bad ending seems redundant; it is entailed by the context.	1.8%
Weirdness	The bad ending is semantically or grammatically malformed, e.g. 'the man is getting out of the horse.'	18.1%
Ambiguous	Both endings seem equally likely.	12.0%

Table 2.4: Justifications for ranking the gold answer over a wrong answer chosen by ESIM+ELMo.

A waiter brings a fork. The waiter **a) starts to step away. (74.76%)** b) adds spaghetti to the table. (21.57%) c) brings a bunch of pie to the food (2.67%) d) drinks from the mug in the bowl. (0.98%)	He is up a tree. Someone **a) stands underneath the tree. (97.44%)** b) is at a pool table holding a cup. (1.14%) c) grabs a flower from a paper. (0.96%) d) is eating some cereal. (0.45%)
An old man rides a small bumper car. Several people a) get in the parking lot. (76.58%) b) wait in the car. (15.28%) **c) get stuck with other bumper cars. (6.75%)** d) are running down the road. (1.39%)	He pours the raw egg batter into the pan. He a) drops the tiny pan onto a plate. (93.48%) **b) lifts the pan and moves it around to shuffle the eggs. (4.94%)** c) stirs the dough into a kite. (1.53%) d) swirls the stir under the adhesive. (0.05%)

Table 2.5: Example questions answered by the best model, ESIM+Elmo, sorted by model probability. Correct model predictions are in **blue**, incorrect model predictions are red. The right answers are **bolded**.

Chapter 3

BENCHMARKING GROUNDEDNESS THROUGH TEXT ALONE, PT2: HELLASWAG

This chapter contains material that was originally published in [307].

3.1 Meta-Introduction

The previous paper in this thesis introduced a new task of *commonsense natural language inference*: given an event description such as "A woman sits at a piano," a machine must select the most likely followup: "She sets her fingers on the keys." But soon after that paper was published, with the introduction of BERT [61], near human-level performance was reached. Does this mean that machines can perform human level commonsense inference?

In the next part of the thesis, we show that commonsense inference still proves difficult for even state-of-the-art models, by presenting HELLASWAG, a new challenge dataset.

3.2 Introduction

Imagine a woman chasing a dog around outside, trying to give it a bath. What might happen next? Humans can read a narrative like this, shown in Figure 2.1, and connect it to a rich model of the world: the dog is currently dry and not soapy, and it actively doesn't want to be bathed. Thus, one plausible next event is option **C**—that she'll get the dog wet and it will run away again.

When the SWAG dataset was first announced [303], this new task of *commonsense natural language inference* seemed trivial for humans (88%) and yet challenging for then-state-of-the-art models (<60%), including ELMo [205]. However, BERT [61] soon reached over 86%, almost human-level performance. One news article on this development was headlined "*finally, a machine*

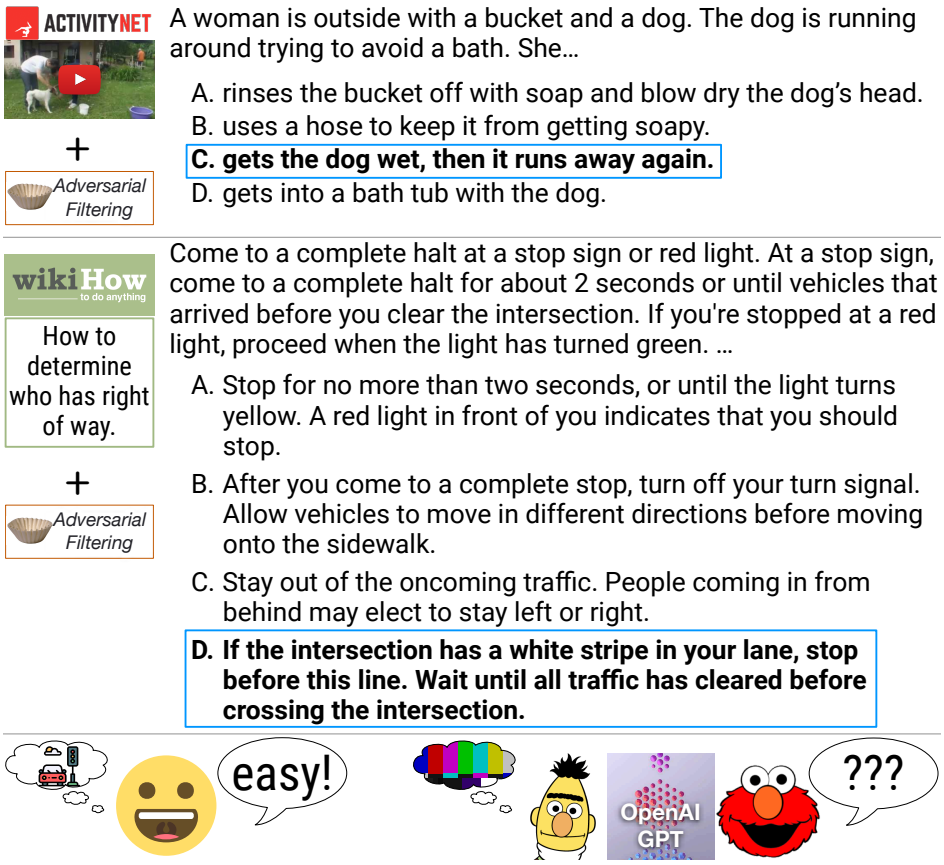


Figure 3.1: Models like BERT struggle to finish the sentences in HELLASWAG, even when they come from the same distribution as the training set. While the wrong endings are on-topic, with words that relate to the context, humans consistently judge their meanings to be either incorrect or implausible. For example, option **A** of the WikiHow passage suggests that a driver should stop at a red light for **no more than two seconds**.

that can finish your sentence."[1]

In this thesis section, we investigate the following question: How well do deep pretrained models, like BERT, perform at commonsense natural language inference (NLI)? Our surprising conclusion is that the underlying *task* remains unsolved. Indeed, we find that deep models such as

[1] A New York Times article at https://nyti.ms/2DycutY.

BERT do not demonstrate robust commonsense reasonining ability by themselves. Instead, they operate more like *rapid surface learners* for a particular dataset. Their strong performance on SWAG is dependent on the finetuning process, wherein they largely learn to pick up on dataset-specific distributional biases. When the distribution of language shifts slightly, performance drops drastically – even if the domain remains identical.

We study this question by introducing HELLASWAG,[2] a new benchmark for commonsense NLI. We use Adversarial Filtering (AF), a data-collection paradigm in which a series of discriminators is used to select a challenging set of generated wrong answers. AF is surprisingly effective towards this goal: the resulting dataset of 70k problems is easy for humans (95.6% accuracy), yet challenging for machines ($<50\%$). This result holds even when models are given a significant number of training examples, and even when the test data comes from the exact same distribution as the training data. Machine performance slips an additional 5% when evaluated on examples that cover novel concepts from the same domain.

To make this dataset robust to deep pretrained models, we use a trifecta of state-of-the-art generators [212], state-of-the-art discriminators (BERT), and high quality source text. We expand on the SWAG's original video-captioning domain by using WikiHow articles, greatly increasing the context diversity and generation length. Our investigation reveals a Goldilocks zone – roughly three sentences of context, and two generated sentences – wherein generations are largely nonsensical, even though state-of-the-art discriminators cannot reliably tell the difference between these generations and the ground truth.

More broadly, this section of the thesis presents a case-study towards a future of verified progress in NLP, via iterative rounds of building and breaking datasets. If our ultimate goal is to provide reliable benchmarks for challenging tasks, such as commonsense NLI, these benchmarks cannot be static. Instead, they must evolve together with the evolving state-of-the-art. Continued evolution in turn requires principled dataset creation algorithms. Whenever a new iteration of a dataset is created, these algorithms must leverage existing modeling advancements to filter out

[2]Short for Harder Endings, Longer Contexts, and Low-Shot Activities. Dataset and code at https://rowanzellers.com/hellaswag.

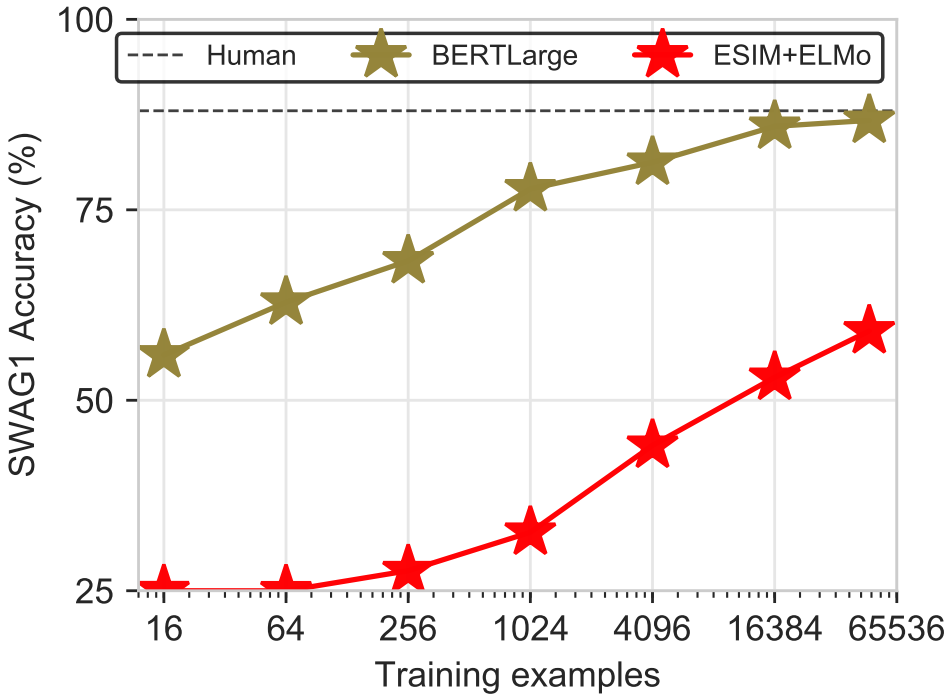


Figure 3.2: Validation accuracy on SWAG for BERT-Large versus training set size. The baseline (25% accuracy) is random chance. BERT does well given as few as 16 training examples, but requires tens of thousands of examples to approach human performance.

spurious biases. Only once this cycle becomes impossible can we say that the underlying *task* – as opposed an individual dataset – is solved.

3.3 Investigating SWAG

In this section, we investigate why SWAG was solved. We focus on BERT, since it is the best known approach at the time of writing.[3] Core to our analysis is investigating how a model trained on Wikipedia and books can be so effectively finetuned for SWAG, a dataset from video captions.

[3]See the appendix for a discussion of the BERT architecture and hyperparameter settings we used in our experiments.

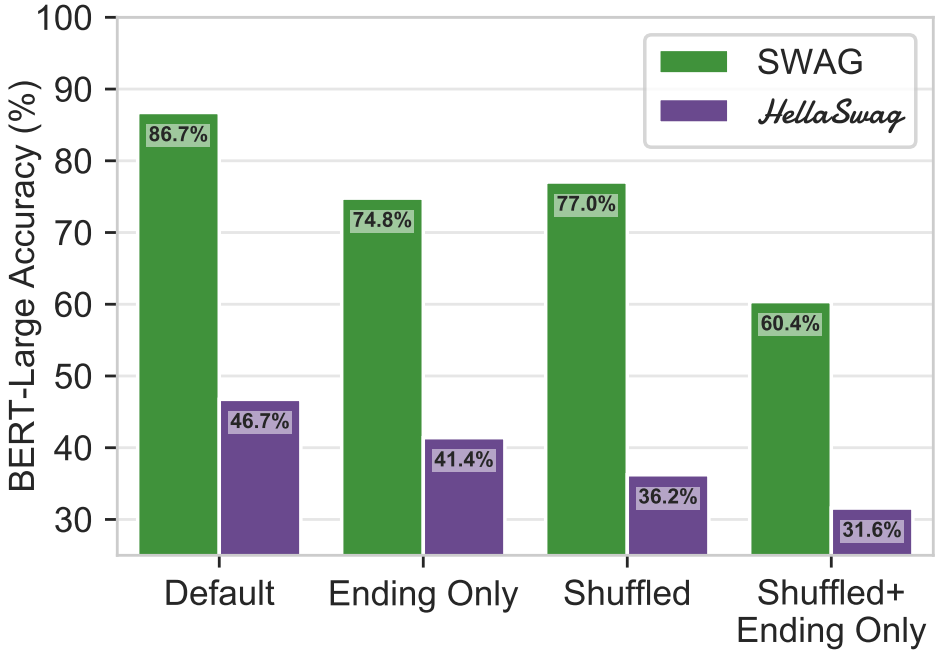


Figure 3.3: BERT validation accuracy when trained and evaluated under several versions of SWAG, with the new dataset HELLASWAG as comparison. We compare:

`Ending Only`	No context is provided; just the endings.
`Shuffled`	Endings that are indidivually tokenized, shuffled, and then detokenized.
`Shuffled+ Ending Only`	No context is provided *and* each ending is shuffled.

3.3.1 How much innate knowledge does BERT have about SWAG?

We investigate this question by measuring BERT's performance on SWAG while varying the size of the training dataset; results are shown in Figure 3.2. While the best known ELMo NLI model (ESIM+ELMo; 42) requires the entire training set to reach 59%, BERT outperforms this given only 64 examples. However, BERT still needs upwards of 16k examples to approach human performance, around which it plateaus.

3.3.2 What is learned during finetuning?

Figure 3.3 compares BERT's performance when trained and evaluated on variants of SWAG.

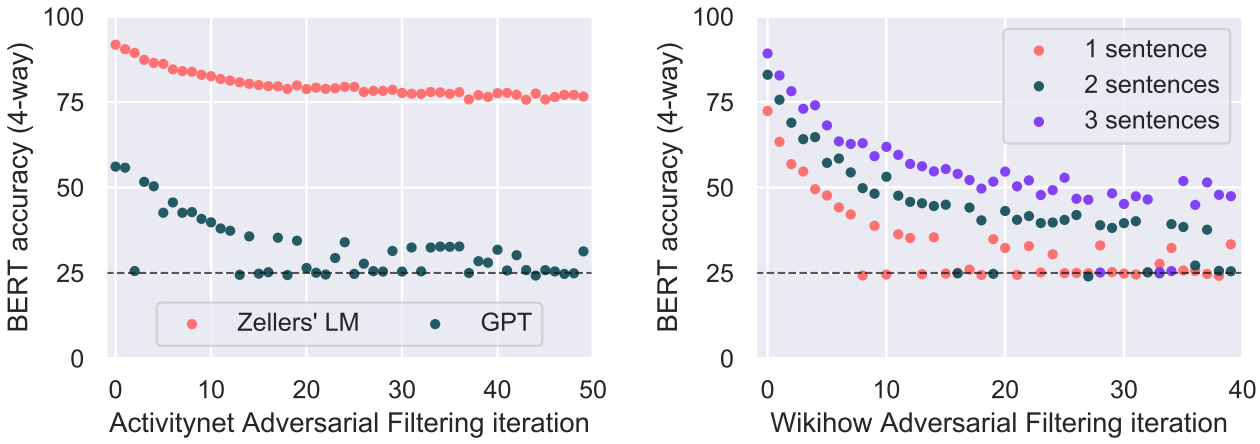


Figure 3.4: Adversarial Filtering (AF) results with BERT-Large as the discriminator. **Left**: AF applied to ActivityNet generations produced by Zellers et al. [303]'s language model versus OpenAI GPT. While GPT converges at random, the LM used for SWAG converges at 75%. **Right**: AF applied to WikiHow generations from GPT, while varying the ending length from one to three sentences. They converge to random, ∼40%, and ∼50%, respectively.

Context: BERT's performance only slips 11.9 points (86.7%→74.8%) when context is omitted (`Ending Only`), suggesting a bias exists in the endings themselves.[4] If a followup event seems unreasonable *absent of context*, then there must be something markedly different between the space of human-written and machine-generated endings.

Structure: To distinguish word usage from structural patterns, we consider a new scenario, `Shuffled`. Here the shared context is provided, but the words in each ending choice are randomly permuted. Surprisingly, this reduces BERT performance by less than 10%. Even though BERT was never exposed to randomly shuffled text during pretraining, it easily adapts to this setting, which suggests that BERT is largely performing lexical reasoning over each (context, answer) pair.

Finally, when the context is removed and the words in each ending are shuffled, performance drops to 60.4%. While low, this is still higher than ELMo's performance (<60% from 303). As neither context nor structure is needed to discriminate between human and machine-written end-

[4]These biases are similar to those in NLI datasets, as found by Gururangan et al. [103], Poliak et al. [210].

ings in a majority of cases, it is likely that systems primarily learn to detect distributional stylistic patterns during finetuning.

3.3.3 Where do the stylistic biases come from?

SWAG was constructed via Adversarial Filtering (AF). Endings were generated via a language model, and then selected to fool a discriminator. To understand why it was solved requires understanding the interplay of AF with respect to SWAG's generators and discriminators.

Zellers et al. [303] used a two-layer LSTM for generation, with shallow stylistic adversarial filters.[5] This setup was robust against ELMo models, but has the shallow LM in particular produced distributional artifacts that BERT picks up on?

To investigate this, we perform AF using BERT-Large as the discriminator[6] in two settings, comparing generations from Zellers et al. [303] with those from a finetuned GPT [212].

Strikingly, the results, Figure 3.4 (left), show that the generations used in SWAG are so different from the human-written endings that *AF never drops the accuracy to chance*; instead, it converges to roughly 75%. On the other hand, GPT's generations are good enough that BERT accuracy drops below 30% over many random subsplits of the data, revealing the importance of the generator.

3.4 HELLASWAG

The success of BERT implies that high-quality generators and discriminators are crucial to AF's success. However, it does *not* imply that the underlying task of commonsense NLI – as opposed to a single dataset – is solved. To evaluate this claim requires us to try making a new evolution of the SWAG dataset, one in which artifacts are removed. In this section, we do just that by introducing HELLASWAG.

---

[5]The discriminator was an ensemble that featured a bag of words model, a shallow CNN, a multilayer perceptron operating on language model perplexities.

[6]On each iteration, BERT-Large is re-initialized from its pretrained checkpoint, finetuned, and then evaluated in a four-way setting on the dummy test set of held-out data. See Supp **??** for a details of our BERT-Large AF setup.

3.4.1 ActivityNet Captions

We start by including video captions from the ActivityNet Captions dataset [148]. The original SWAG dataset contains these, along with captions from LSMDC [224], but for HELLASWAG we solely used ActivityNet. In addition to temporal descriptions, ActivityNet also provides activity labels for each caption (e.g. `jumping rope`). We will use these activity labels as additional structure to test generalization ability.

3.4.2 WikiHow: A New Testbed

We next consider a new and challenging testbed for commonsense reasoning: completing how-to articles from WikiHow, an online how-to manual. We scrape 80k context and follow-up paragraphs from WikiHow, covering such diverse topics as "how to make an origami owl" to "how to survive a bank robbery." Each context has at most three sentences, as do the follow-ups.

AF's effectiveness in this new setting is shown in Figure 3.4 (right). We consider three settings, corresponding to endings that are either one, two, or three sentences long. In all cases, BERT performance begins high (70-90%), but there are enough generations for Adversarial Filtering to lower the final accuracy considerably. While the one-sentence case converges to slightly higher than random – 35% when it converges – the two and three sentence cases are higher, at 40% and 50% respectively. Given more context, it becomes easier to classify an ending as machine- or human-written. We compromise and use two-sentence generations. Particularly in the two-sentence case, we find ourselves in a Goldilocks zone wherein generations are challenging for deep models, yet as we shall soon see, easy for humans.

3.4.3 Obtaining high human agreement

How well can humans distinguish human-written endings from machine generations refined with Adversarial Filtering? In Figure 3.5, we compare human performance with that of BERT on a random 80%/20% split. We see a contrast between the ActivityNet and WikiHow performance. While ActivityNet starts off harder for BERT (25.5%), it also proves difficult for humans (60%).

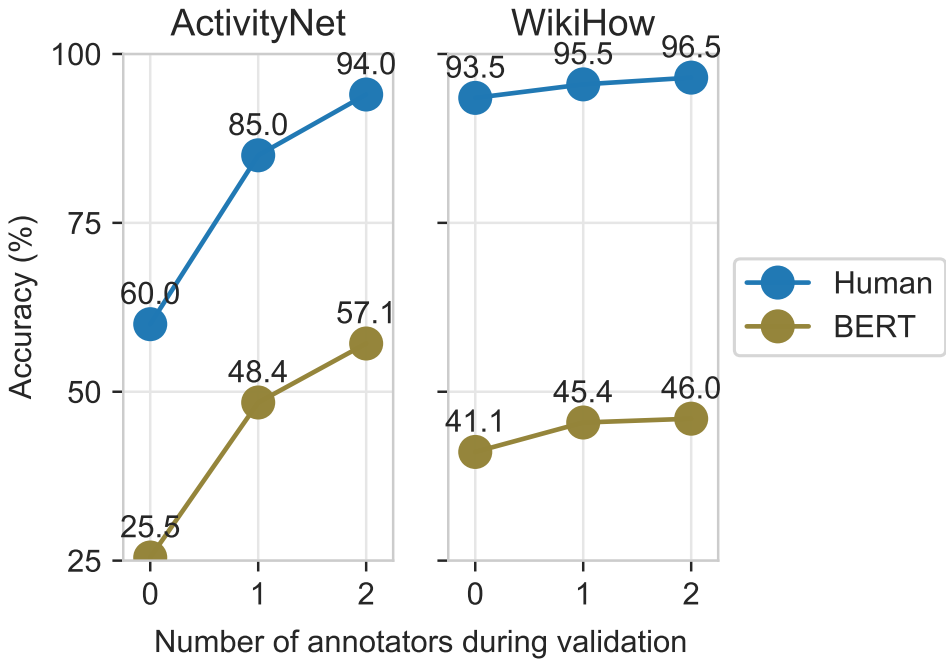


Figure 3.5: For HELLASWAG, we ensure high human agreement through several rounds of annotation. By collecting how likely each ending is we can filter false negative endings – machine generations that sound realistic – and replace them with true negatives. On both subdatasets, BERT performance increases during validation, but the gap to human performance remains wide.

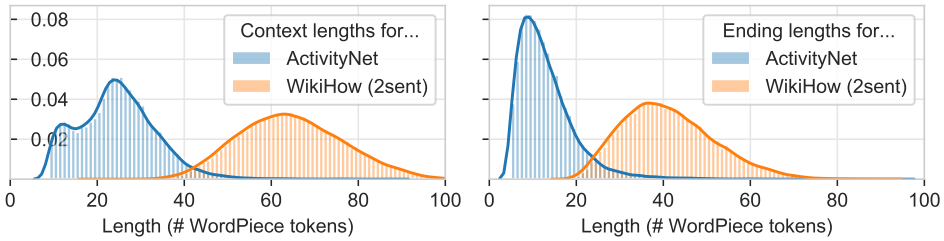


Figure 3.6: Lengths of ActivityNet and WikiHow; the latter with two-sentence generations. WikiHow is much longer, which corresponds to being easier for humans, while taking longer for AF to converge.

In contrast, WikiHow starts easier for BERT (41.1%) and humans find the domain almost trivial (93.5%). We hypothesis this discrepancy is due to the lengths of both datasets (Figure 3.6). WikiHow's 2-sentence generations average 41 tokens, versus 13 for ActivityNet. This gives WikiHow generations three times as many opportunities to make a detectable mistake.

To ensure high agreement on ActivityNet, we perform several rounds of human filtering, increasing human performance to 94%. During human validation, crowd workers are given a context and six ending choices, of which one is the true ending, and the other five are from AF. On each iteration, we replace machine-written endings that the worker rated as realistic with new samples. In the end, we keep the 25k best ActivityNet contexts (i.e. those with highest agreement among workers) and the 45k best WikiHow contexts.

3.4.4 Zero-shot categories for evaluation

To evaluate a model's ability to generalize to new situations, we use category labels from WikiHow and ActivityNet to make 'zero-shot' evaluation sets. For each set (validation or test), we craft two subsets: one containing 5k 'in-domain' examples that come from categories as seen during training (Figure 3.7), and another with 5k 'zero-shot' examples from randomly chosen held-out categories. In total, there are 70k dataset examples.

3.5 Results

We evaluate the difficulty of HELLASWAG using a variety of strong baselines, with and without massive pretraining. The models share the same format: given a context and an ending, return a *logit* for that ending. Accordingly, we train our models using a four-way cross-entropy loss, where the objective is to predict the correct ending. In addition to BERT-Large, our comparisons include:

a. **OpenAI GPT** [212]: A finetuned 12-layer transformer that was pre-trained on the BookCorpus [325].

b. **Bert-Base**: A smaller version of the BERT model whose architecture size matches GPT.

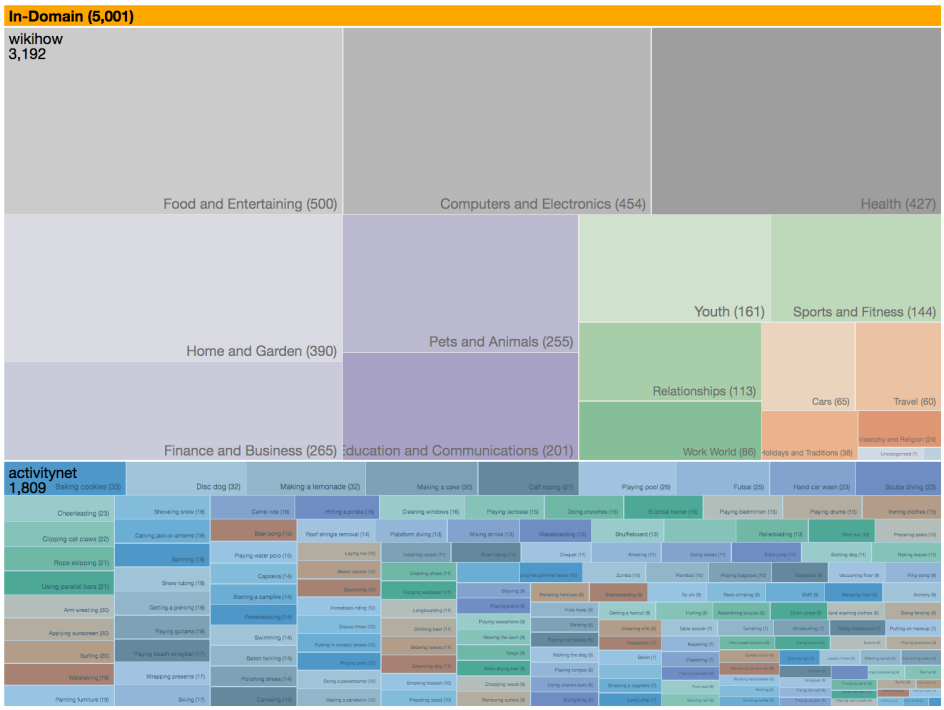


Figure 3.7: Examples on the in-domain validation set of HELLASWAG, grouped by category label. Our evaluation setup equally weights performance on categories seen during training as well as out-of-domain.

c. **ESIM+ELMo** [42, 205]: This is the best-performing ELMo model for NLI, modified slightly so the final output layer is now a four-way softmax over endings.

d. **LSTM sentence encoder**: This is a randomly initialized two-layer bi-LSTM; the second layer's hidden states are max-pooled and fed into an MLP to predict the logit. We consider three variations: GloVe embeddings, ELMo embeddings, or (frozen) BERT-Base embeddings.[7]

e. **FastText**: [136] An off-the-shelf library for bag-of-words text classification.[8]

We compare all models to human performance by asking five independent crowd workers to solve the same four-way multiple choice problems; their predictions are combined via majority

[7]For ELMo and BERT-Base, the model learns scalar weights to combine each internal layer of the encoder.

[8]This model is trained with binary cross entropy loss.

	Overall		In-Domain		Zero-Shot		ActivityNet		WikiHow	
Model	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test
Split Size→	10K	10K	5K	5K	5K	5K	3.2K	3.5K	6.8K	6.5K
Chance	25.0									
fastText	30.9	31.6	33.8	32.9	28.0	30.2	27.7	28.4	32.4	33.3
LSTM+GloVe	31.9	31.7	34.3	32.9	29.5	30.4	34.3	33.8	30.7	30.5
LSTM+ELMo	31.7	31.4	33.2	32.8	30.4	30.0	33.8	33.3	30.8	30.4
LSTM+BERT-Base	35.9	36.2	38.7	38.2	33.2	34.1	40.5	40.5	33.7	33.8
ESIM+ELMo	33.6	33.3	35.7	34.2	31.5	32.3	37.7	36.6	31.6	31.5
OpenAI GPT	41.9	41.7	45.3	44.0	38.6	39.3	46.4	43.8	39.8	40.5
BERT-Base	39.5	40.5	42.9	42.8	36.1	38.3	48.9	45.7	34.9	37.7
BERT-Large	**46.7**	**47.3**	**50.2**	**49.7**	**43.3**	**45.0**	**54.7**	**51.7**	**42.9**	**45.0**
Human	95.7	95.6	95.6	95.6	95.8	95.7	94.0	94.0	96.5	96.5

Table 3.1: Performance of models, evaluated with accuracy (%).We report results on the full validation and test sets (Overall), as well as results on informative subsets of the data: evaluated on in-domain, versus zero-shot situations, along with performance on the underlying data sources (ActivityNet versus WikiHow). All models substantially underperform humans: the gap is over 45% on in-domain categories, and 50% on zero-shot categories.

vote.

Our results, shown in Table 3.1, hint at the difficulty of the dataset: human performance is over 95%, while overall model performance is below 50% for every model. Surprisingly, despite BERT-Large having been used as the adversarial filter, it still performs the strongest at 47.3% overall. By making the dataset adversarial for BERT, it seems to also have become adversarial **for every other model.** For instance, while ESIM+ELMo obtained 59% accuracy on SWAG, it obtains only 33.3% accuracy on HELLASWAG.

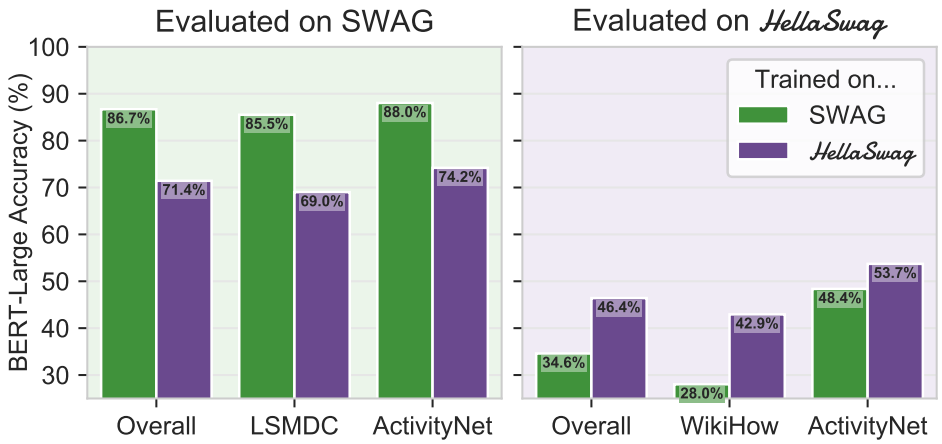


Figure 3.8: Transfer experiments from SWAG to HELLASWAG and vice versa, evaluated on the validation sets. Overall, a BERT-Large that is trained on SWAG hardly generalizes to HELLASWAG: it scores 34.6%.

In addition to pretraining being critical, so too is end-to-end finetuning. Freezing BERT-Base and adding an LSTM on top lowers its overall performance 4.3%. This may help explain why models such as ESIM+ELMo struggled on SWAG, as ELMo isn't updated during finetuning.

While BERT is the best model, it still struggles on HELLASWAG, and especially so on zero-shot categories. Performance drops roughly 5% on the test fold, which suggests that the finetuning is not enough for BERT to learn to generalize to novel activities or how-to categories.

Last, we see that WikiHow is a much harder domain that ActivityNet for machines: 45% Bert-Large performance, versus 96.5% for humans. Curiously, it is on this source dataset that we see the smallest gap between OpenAI GPT and BERT. In fact, OpenAI GPT outperforms BERT on WikiHow, but the reverse is true for ActivityNet. One possibility is that the left-to-right structure of GPT is the right inductive bias for WikiHow - perhaps reasoning bidirectionally over long contexts is too much for a 12-layer transformer to learn.

3.5.1 SWAG *to* HELLASWAG *transfer*

Given the shared goals and partial domains of SWAG and HELLASWAG, it is natural to ask to what extent models can transfer between the two datasets. In Figure 3.8 we show the results from transfer experiments: models are trained on one dataset and evaluated on the other.[9]

The best models are trained on the same dataset that they are evaluated on: training on SWAG and evaluating on HELLASWAG lowers performance by 12%; vice versa lowers performance by 15%. The missing domain for HELLASWAG models is movie descriptions (LSMDC), still, HELLASWAG models obtain 69% accuracy. On the other hand, SWAG models do not generalize at all to their missing domain, WikiHow (28%), suggesting that learning general commonsense reasoning was hardly necessary to solve SWAG.

3.5.2 Qualitative examples

We show several qualitative examples in Figure 3.9, along with BERT-Large's predictions. BERT does well on some ActivityNet contexts, such as in the first row, where it correctly predicts the ending for a `shaving` caption. Whereas *shaving* is in-domain, the second example about `sharpening knives` is zero-shot. In this context, BERT's answer suggests that one would use a knife to sharpen a stone, rather than vice versa. The last example comes from WikiHow, which appears to be incredibly challenging for BERT. BERT picks answer **d**, which has more words that match the context of *technology* (planes, traffic, laptop), but is incoherent.[10]

3.6 Discussion

Our results suggest that HELLASWAG is a challenging testbed for state-of-the-art NLI models, even those built on extensive pretraining. The question still remains, though, of *where will the field*

[9]Note that the ActivityNet splits are different for each dataset. To avoid skewing the results, we report only on the validation video captions that are not in the training sets of either dataset. The overall accuracy is then a weighted average, where ActivityNet examples are weighted proportionately more. This gives a slight advantage to training on SWAG, as it sees all the ActivityNet categories when training.

[10]Among other issues, why would someone suddenly be aware that they are 'flying at high speed on a plane...?'

Category: Shaving (ActivityNet; In-domain)

A bearded man is seen speaking to the camera and making several faces. the man

a) then switches off and shows himself via the washer and dryer rolling down a towel and scrubbing the floor. (0.0%)

b) then rubs and wipes down an individual's face and leads into another man playing another person's flute. (0.0%)

c) is then seen eating food on a ladder while still speaking. (0.0%)

d) then holds up a razor and begins shaving his face. (100.0%)

Category: Sharpening knives (ActivityNet; Zero-Shot)

Two men are in a room and the man with a blue shirt takes out a bench stone and with a little lubricant on the stone takes an knife and explains how to sharpen it. then he

a) uses a sharpener to smooth out the stone using the knife. (100.0%)

b) shows how to cut the bottom with the knife and place a tube on the inner and corner. (0.0%)

c) bends down and grabs the knife and remove the appliance. (0.0%)

d) stops sharpening the knife and takes out some pieces of paper to show how sharp the knife is as he cuts slivers of paper with the knife. (0.0%)

Category: Youth (WikiHow; In-Domain)

HOW TO MAKE UP A GOOD EXCUSE FOR YOUR HOMEWORK NOT BEING FINISHED

Blame technology. One of the easiest and most believable excuses is simply blaming technology. You can say your computer crashed, your printer broke, your internet was down, or any number of problems.

a) Your excuses will hardly seem believable. [substeps] This doesn't mean you are lying, just only that you don't have all the details of how your computer ran at the time of the accident. (0.0%)

b) The simplest one to have in a classroom is to blame you entire classroom, not just lab. If you can think of yourself as the victim, why not blame it on technology. (9.4%)

c) Most people, your teacher included, have experienced setbacks due to technological problems. [substeps] This is a great excuse if you had a paper you needed to type and print. (29.1%)

d) It may also be more believable if you are fully aware that you may be flying at high speed on a plane and need someone to give you traffic report. Your problem might be your laptop failing to charge after a long flight. (61.5%)

Figure 3.9: Example questions answered by BERT-Large. Correct model predictions are **blue**, incorrect predictions are red. The right answers are **bolded**.

go next?

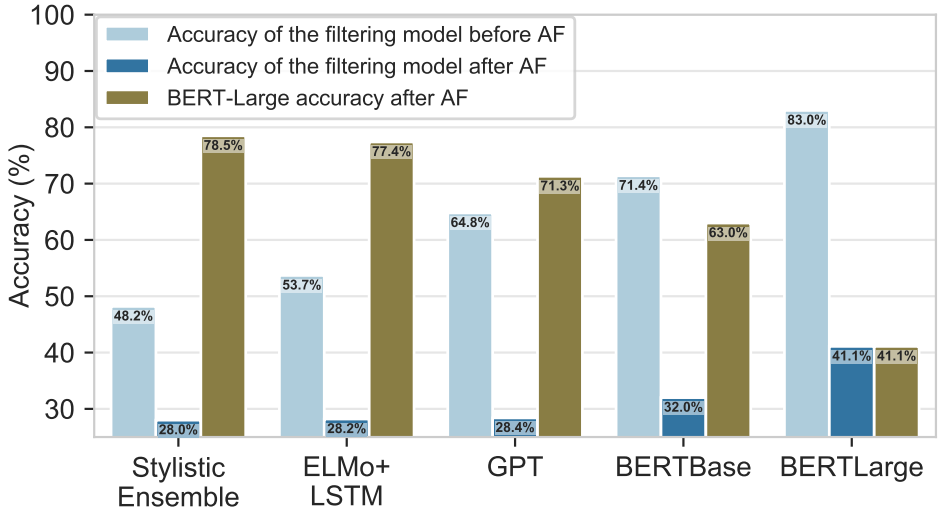


Figure 3.10: Performance on the WikiHow subset of alternative variations of HELLASWAG, where different Adversarial Filters are used (but without human validation). We consider the shallow stylistic adversaries used by Zellers et al. [303] (Stylistic Ensemble), as well as an LSTM with ELMo embeddings, GPT, BERT-Base, and BERT-Large. For each adversarial filtering model, we record the accuracy of that model before and after AF is used. We also evaluate each alternative dataset using BERT-Large. The results suggest that using a a stronger model at test time (over the model used for AF) improves performance, but is not enough to solve the task.

3.6.1 How easy might HELLASWAG *be for future discriminators?*

In this section of the thesis, we showed the existence of a Goldilocks zone of text complexity – in which generations are nonsensical, but existing state-of-the-art NLP models cannot tell the difference. How hard will the dataset be for future, even more powerful, models?

Answering this question is challenging because *these models don't exist (or are unavailable) at the time of writing*. However, one remedy is to perform an ablation study on the Adversarial Filtering model used, comparing weaker filters with stronger discriminators. We present our results in Figure 3.10, and find that while weak discriminators (like the stylistic ensemble used to make SWAG) only marginally reduce the accuracy of BERT-Large, increasing the gap between the filter

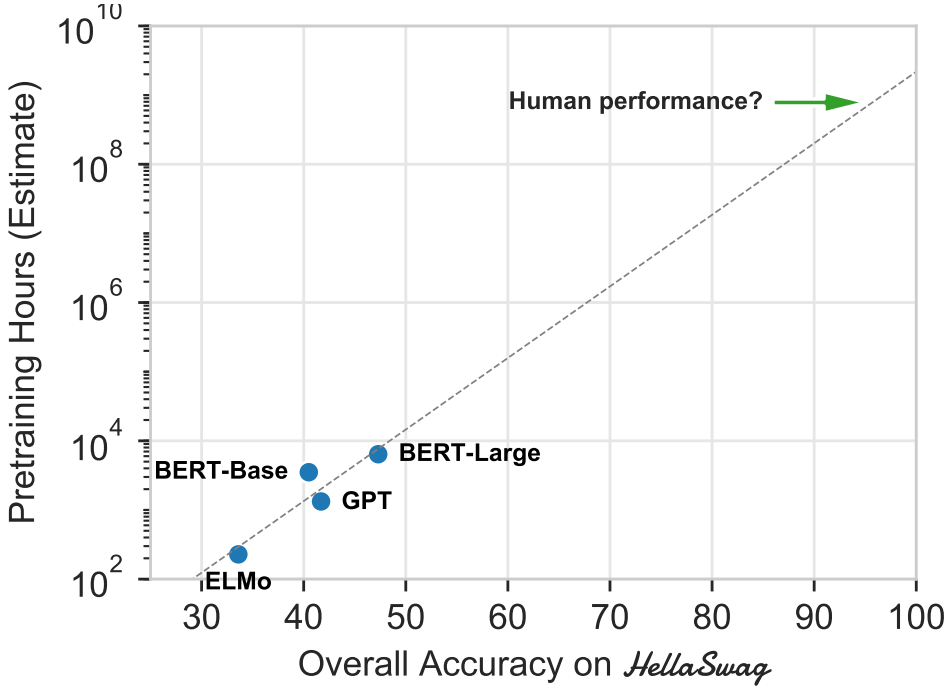


Figure 3.11: Estimated pretraining hours required to reach a desired accuracy on HELLASWAG. We estimate perfomance with respect to a RTX 2080 Ti - a modern, fast GPU, and fit a log-linear regression line. An extrapolation suggests that to reach human-level performance on HELLASWAG, without algorithmic or computational improvements, would require 10^9 GPU-hours of pretraining (over 100k GPU years).

and the final discriminator is not enough to solve the task. For instance, using a discriminator with 3x the parameters as the adversarial filter (BERT-Large vs. BERT-Base) results in 63% machine accuracy.

3.6.2 How well does pretraining scale?

Overall, the current paradigm of pretraining large models on lots of data has made immense progress on NLP benchmarks. Though we expect this trend to continue, it also behooves us to consider its limits. If more compute is indeed the answer for human-level commonsense inference,

what would the compute requirements of this hypothetical massive model look like?

We investigate this in Figure 3.11 by comparing the accuracies of known models on HELLASWAG with their computational needs. This estimation is a rough estimate: we convert reported TPU runtimes to our benchmark RTX 2080 Ti GPU using the Roofline model [279], which focuses primarily on the bottleneck of loading tensors into GPU memory. Extrapolating from an exponential fit suggests that reaching human-level performance on our dataset would require 10^9 GPU hours, or 100k years – unless algorithmic improvements are made.

What might these algorithmic improvements look like? These could include architectural advances, better pretraining objectives, and beyond. However, these improvements share the bottleneck of the data source. To answer some HELLASWAG questions correctly without reasoning deeply – like knowing that it is a bad idea to stop at a red light for 'at most two seconds' – might require an exponential number of samples, due to problems of reporting bias [93]. Alternatively, future models might answer correctly only by picking up on spurious patterns, in which case a new development of the benchmark – using these models as adversaries – would place us in the same position as we are right now.

Put another way, for humans to answer HELLASWAG questions requires *abstracting away* from language and modeling *world states* instead. We postulate that this is what separates solving the *task* of commonsense NLI, as opposed to a particular dataset. Indeed, we find that existing deep methods often get fooled by lexical false friends. For example, in the WikiHow example from Figure 3.9, BERT chooses an ending that matches the *technology* words in the context, rather than matching the deeper topic: using technology as an excuse for not doing homework.

3.6.3 Towards a future of evolving benchmarks

What happens when HELLASWAG gets solved? We believe the answer is simple: crowdsource another dataset, with the same exact format, and see where models fail. Indeed, in our work we found this to be straightforward from an *algorithmic* perspective: by throwing in the *best known generator* (GPT) and the *best known discriminator* (BERT-Large), we made a dataset that is adversarial - not just to BERT, but to all models we have access to.

While this was easy algorithmically, care must be taken from a data curation standpoint. Indeed, we find success exists within a Goldilocks zone: the data source must be complex enough that state-of-the-art generators often make mistakes, while simple enough such that discriminators often fail to catch them. This ties the future of SWAG-style benchmarks to progress on language generation: until generation is solved, commonsense NLI will remain unsolved. Even recent promising results on scaling up language models [213] find problems in terms of consistency, with the best curated examples requiring 25 random seeds.

3.7 Conclusion

In this section of the thesis, we presented HELLASWAG, a new dataset for physically situated commonsense reasoning. By constructing the dataset through adversarial filtering, combined with state-of-the-art models for language generation and discrimination, we produced a dataset that is adversarial to the most robust models available – even when models are evaluated on items from the training distribution. In turn, we provided insight into the inner workings of pretrained models, and suggest a path for NLP progress going forward: towards benchmarks that adversarially co-evolve with evolving state-of-the-art models.

Chapter 4

BENCHMARKING GROUNDEDNESS THROUGH TEXT AND IMAGES

This chapter contains material that was originally published in [305].

4.1 Introduction

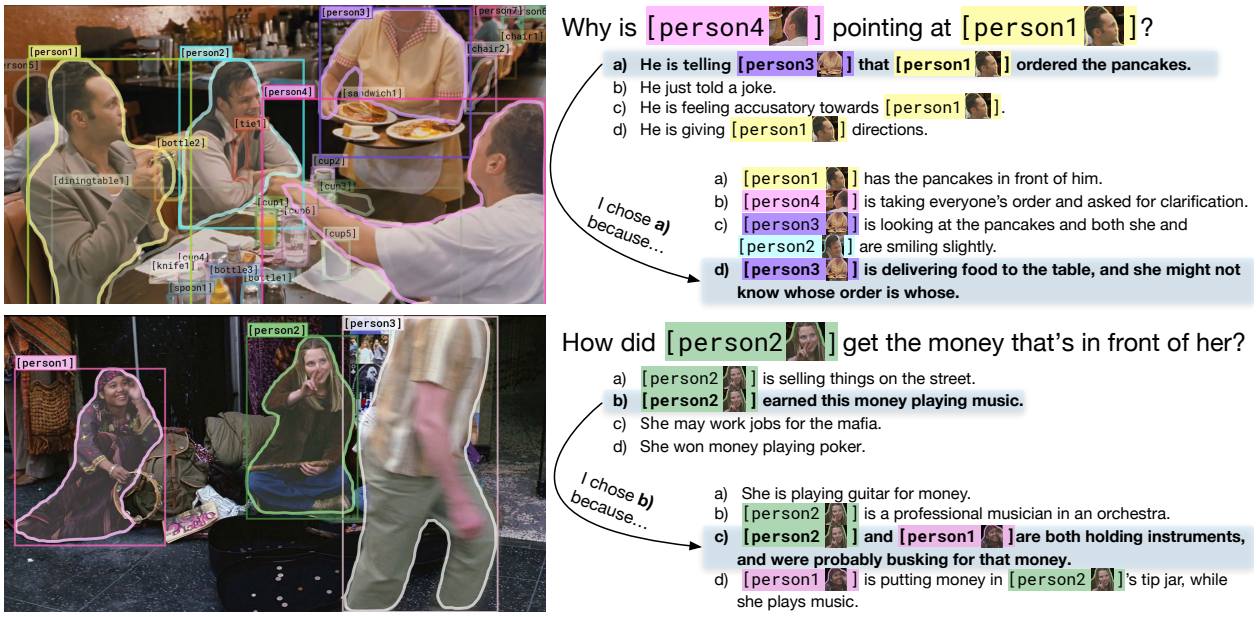


Figure 4.1: VCR: Given an image, a list of regions, and a question, a model must answer the question and provide a *rationale* explaining why its answer is right. Our questions challenge computer vision systems to go beyond recognition-level understanding, towards a higher-order cognitive and commonsense understanding of the world depicted by the image.

With one glance at an image, we can immediately infer what is happening in the scene beyond what is visually obvious. For example, in the top image of Figure 2.1, not only do we see several

objects (people, plates, and cups), we can also reason about the entire situation: three people are dining together, they have already ordered their food before the photo has been taken, PERSON3 is serving and not eating with them, and what PERSON1 ordered are the pancakes and bacon (as opposed to the cheesecake), because PERSON4 is pointing to PERSON1 while looking at the server, PERSON3.

Visual understanding requires seamless integration between *recognition* and *cognition*: beyond recognition-level perception (e.g., detecting objects and their attributes), one must perform cognition-level reasoning (e.g., inferring the likely intents, goals, and social dynamics of people) [57]. State-of-the-art vision systems can reliably perform *recognition-level* image understanding, but struggle with complex inferences, like those in Figure 2.1. We argue that as the field has made significant progress on recognition-level building blocks, such as object detection, pose estimation, and segmentation, now is the right time to tackle cognition-level reasoning at scale.

As a critical step toward complete visual understanding, we present the task of VISUAL COMMONSENSE REASONING. Given an image, a machine must answer a question that requires a thorough understanding of the visual world evoked by the image. Moreover, the machine must provide a rationale justifying why that answer is true, referring to the details of the scene, as well as background knowledge about how the world works. These questions, answers, and rationales are expressed using a mixture of rich natural language as well as explicit references to image regions. To support clean-cut evaluation, all our tasks are framed as multiple choice QA.

Our new dataset for this task, VCR, is the first of its kind and is large-scale — 290k pairs of questions, answers, and rationales, over 110k unique movie scenes. A crucial challenge in constructing a dataset of this complexity at this scale is how to avoid annotation artifacts. A recurring challenge in most recent QA datasets has been that human-written answers contain unexpected but distinct biases that models can easily exploit. Often these biases are so prominent so that models can select the right answers without even looking at the questions [103, 210, 231].

Thus, we present **Adversarial Matching**, a novel QA assignment algorithm that allows for robust multiple-choice dataset creation at scale. The key idea is to recycle each correct answer

for a question exactly three times — as a negative answer for three other questions. Each answer thus has the same probability (25%) of being correct: this resolves the issue of answer-only biases, and disincentivizes machines from always selecting the most generic answer. We formulate the answer recycling problem as a constrained optimization based on the relevance and entailment scores between each candidate negative answer and the gold answer, as measured by state-of-the-art natural language inference models [42, 205, 61]. A neat feature of our recycling algorithm is a knob that can control the tradeoff between human and machine difficulty: we want the problems to be hard for machines while easy for humans.

Narrowing the gap between recognition- and cognition-level image understanding requires grounding the meaning of the natural language passage in the visual data, understanding the answer in the context of the question, and reasoning over the shared and grounded understanding of the question, the answer, the rationale and the image. In this section of the thesis we introduce a new model, **Recognition to Cognition Networks** (**R2C**). Our model performs three inference steps. First, it *grounds* the meaning of a natural language passage with respect to the image regions (objects) that are directly referred to. It then *contextualizes* the meaning of an answer with respect to the question that was asked, as well as the global objects not mentioned. Finally, it *reasons* over this shared representation to arrive at an answer.

Experiments on VCR show that **R2C** greatly outperforms state-of-the-art visual question-answering systems: obtaining 65% accuracy at question answering, 67% at answer justification, and 44% at staged answering and justification. Still, the task and dataset is far from solved: humans score roughly 90% on each. We provide detailed insights and an ablation study to point to avenues for future research.

In sum, our major contributions are fourfold: (1) we formalize a new task, Visual Commonsense Reasoning, and (2) present a large-scale multiple-choice QA dataset, VCR, (3) that is automatically assigned using Adversarial Matching, a new algorithm for robust multiple-choice dataset creation. (4) We also propose a new model, **R2C**, that aims to mimic the layered inferences from recognition to cognition; this also establishes baseline performance on our new challenge.

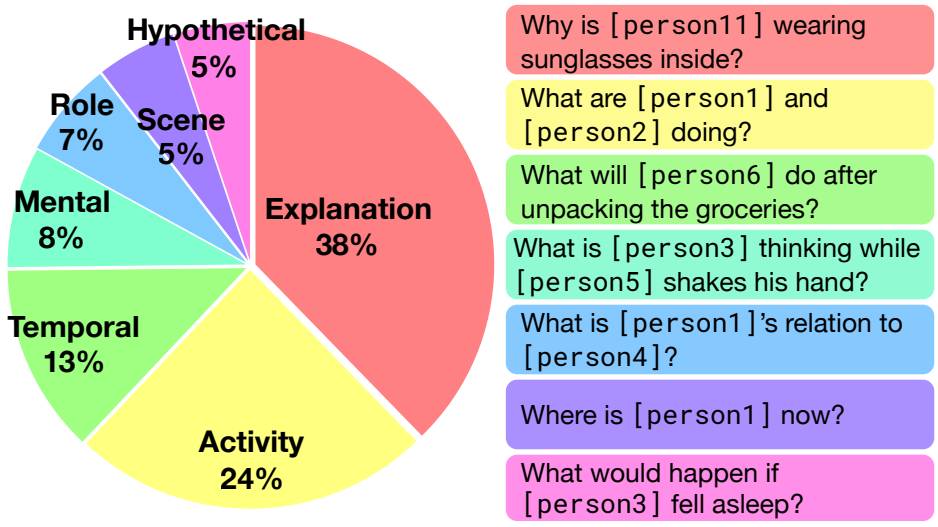


Figure 4.2: Overview of the types of inference required by questions in VCR. Of note, 38% of the questions are explanatory 'why' or 'how' questions, 24% involve cognition-level activities, and 13% require temporal reasoning (i.e., what might come next). These categories are not mutually exclusive; an answer might require several hops of different types of inferences.

4.2 Task Overview

We present VCR, a new task that challenges vision systems to holistically and cognitively understand the content of an image. For instance, in Figure 4.1, we need to understand the activities (PERSON3 is delivering food), the roles of people (PERSON1 is a customer who previously ordered food), the mental states of people (PERSON1 wants to eat), and the likely events before and after the scene (PERSON3 will serve the pancakes next). Our task covers these categories and more: a distribution of the inferences required is in Figure 4.2.

Visual understanding requires not only answering questions correctly, but doing so *for the right reasons*. We thus require a model to give a *rationale* that explains why its answer is true. Our questions, answers, and rationales are written in a mixture of rich natural language as well as detection tags, like 'PERSON2': this helps to provide an unambiguous link between the textual description of an object ('the man on the left in the white shirt') and the corresponding

image region.

To make evaluation straightforward, we frame our ultimate task – of staged answering and justification – in a multiple-choice setting. Given a question along with four answer choices, a model must first select the right answer. If its answer was correct, then it is provided four rationale choices (that could purportedly justify its correct answer), and it must select the correct rationale. We call this $Q \rightarrow AR$ as for the model prediction to be correct requires *both the chosen answer and then the chosen rationale* to be correct.

Our task can be decomposed into two multiple-choice sub-tasks, that correspond to answering ($Q \rightarrow A$) and justification ($QA \rightarrow R$) respectively:

Definition VCR *subtask.* A single example of a VCR subtask consists of an image $\boldsymbol{I}$, and:

- A sequence $\boldsymbol{o}$ of object detections. Each object detection o_i consists of a *bounding box* $\mathbf{b}$, a segmentation mask $\mathbf{m}^1$, and a class label $\ell_i \in \mathcal{L}$.
- A *query* $\boldsymbol{q}$, posed using a mix of natural language and pointing. Each word $\boldsymbol{q}_i$ in the query is either a word in a vocabulary $\mathcal{V}$, or is a tag referring to an object in $\boldsymbol{o}$.
- A set of N *responses*, where each response $\boldsymbol{r}^{(i)}$ is written in the same manner as the query: with natural language and pointing. Exactly one response is correct.

The model chooses a single (best) response.

In question-answering ($Q \rightarrow A$), the query is the question and the responses are answer choices. In answer justification ($QA \rightarrow R$), the query is the concatenated question and correct answer, while the responses are rationale choices.

In this paper, we evaluate models in terms of accuracy and use $N=4$ responses. Baseline accuracy on each subtask is then 25% ($1/N$). In the holistic setting ($Q \rightarrow AR$), baseline accuracy is 6.25% ($1/N^2$) as there are two subtasks.

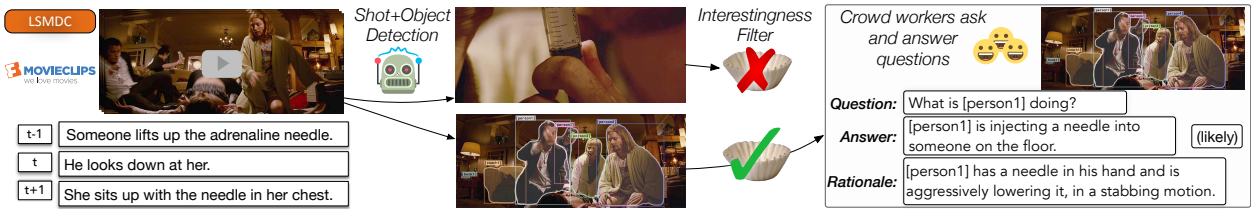


Figure 4.3: An overview of the construction of VCR. Using a state-of-the-art object detector [109, 86], we identify the objects in each image. The most interesting images are passed to crowd workers, along with scene-level context in the form of scene descriptions (MovieClips) and video captions (LSMDC, [224]). The crowd workers use a combination of natural language and detection tags to ask and answer challenging visual questions, also providing a rationale justifying their answer.

4.3 Data Collection

In this section, we describe how we collect the questions, *correct answers* and *correct rationales* for VCR. Our key insight – towards collecting commonsense visual reasoning problems at scale – is to carefully select interesting situations. We thus extract still images from movie clips. The images from these clips describe complex situations that humans can decipher without additional context: for instance, in Figure 4.1, we know that PERSON3 will serve PERSON1 pancakes, whereas a machine might not understand this unless it sees the entire clip.

Interesting and Diverse Situations To ensure diversity, we make no limiting assumptions about the predefined set of actions. Rather than searching for predefined labels, which can introduce search engine bias [256, 60, 75], we collect images from movie scenes. The underlying scenes come from the Large Scale Movie Description Challenge [224] and YouTube movie clips.[2] To avoid simple images, we train and apply an 'interestingness filter' (e.g. a closeup of a syringe in

[1]The task is agnostic to the representation of the mask, but it could be thought of as a list of polygons $\boldsymbol{p}$, with each polygon consisting of a sequence of 2d vertices inside the box $\boldsymbol{p}_j = \{x_t, y_t\}_t$.

[2]Namely, Fandango MovieClips: `youtube.com/user/movieclips`.

Figure 4.3).

We center our task around challenging questions requiring cognition-level reasoning. To make these cognition-level questions simple to ask, and to avoid the clunkiness of referring expressions, VCR's language integrates object tags (PERSON2 ) and explicitly excludes referring expressions ('the woman on the right.') These object tags are detected from Mask-RCNN [109, 86], and the images are filtered so as to have at least three high-confidence tags.

Crowdsourcing Quality Annotations Workers on Amazon Mechanical Turk were given an image with detections, along with additional context in the form of video captions.[3] They then ask one to three questions about the image; for each question, they provide a reasonable answer and a rationale. To ensure top-tier work, we used a system of quality checks and paid our workers well.[4]

The result is an underlying dataset with high agreement and diversity of reasoning. Our dataset contains a myriad of interesting commonsense phenomena (Figure 4.2) and a great diversity in terms of unique examples.

4.4 Adversarial Matching

We cast VCR as a four-way multiple choice task, to avoid the evaluation difficulties of language generation or captioning tasks where current metrics often prefer incorrect machine-written text over correct human-written text [171]. However, it is not obvious how to obtain high-quality incorrect choices, or counterfactuals, at scale. While past work has asked humans to write several counterfactual choices for each correct answer [253, 162], this process is expensive. Moreover, it has the potential of introducing annotation artifacts: subtle patterns that are by themselves highly predictive of the 'correct' or 'incorrect' label [231, 103, 210].

In this work, we propose Adversarial Matching: a new method that allows for any 'language generation' dataset to be turned into a multiple choice test, while requiring minimal human involvement. An overview is shown in Figure 4.4. Our key insight is that the problem of obtaining

[3]This additional clip-level context helps workers ask and answer about what will happen next.

[4]More details in the appendix, Sec **??**.

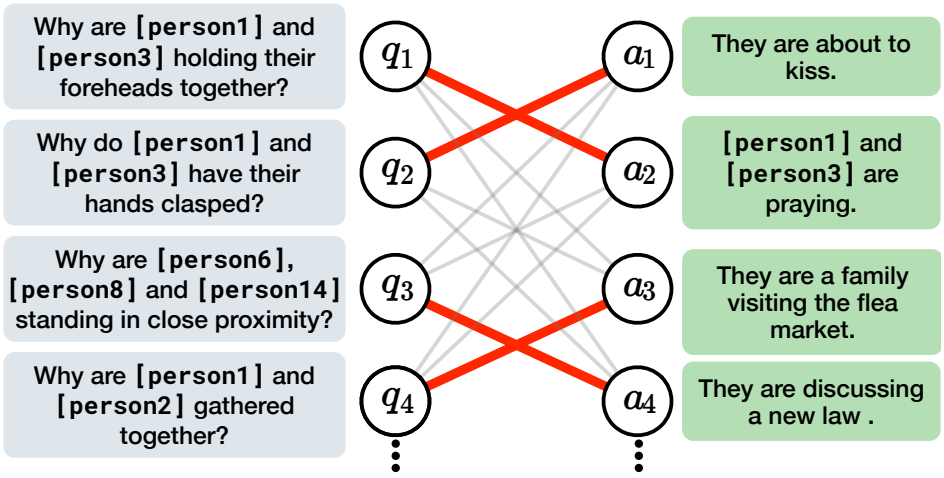


Figure 4.4: Overview of Adversarial Matching. Incorrect choices are obtained via maximum-weight bipartite matching between queries and responses; the weights are scores from state-of-the-art natural language inference models. Assigned responses are highly relevant to the query, while they differ in meaning versus the correct responses.

good counterfactuals can be broken up into two subtasks: the counterfactuals must be as **relevant** as possible to the context (so that they appeal to machines), while they cannot be overly **similar** to the correct response (so that they don't become correct answers incidentally). We balance between these two objectives to create a dataset that is challenging for machines, yet easy for humans.

Formally, our procedure requires two models: one to compute the relevance between a query and a response, P_{rel}, and another to compute the similarity between two response choices, P_{sim}. Here, we employ state-of-the-art models for Natural Language Inference: BERT [61] and ESIM+ELMo [42, 205], respectively.[5] Then, given dataset examples $(\boldsymbol{q}_i, \boldsymbol{r}_i)_{1 \leq i \leq N}$, we obtain a counterfactual for each $\boldsymbol{q}_i$ by performing maximum-weight bipartite matching [196, 134] on a weight matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$, given by

$$\mathbf{W}_{i,j} = \log(P_{rel}(\boldsymbol{q}_i, \boldsymbol{r}_j)) + \lambda \log(1 - P_{sim}(\boldsymbol{r}_i, \boldsymbol{r}_j)). \tag{4.1}$$

[5]We finetune P_{rel} (BERT), on the annotated data (taking steps to avoid data leakage), whereas P_{sim} (ESIM+ELMo) is trained on entailment and paraphrase data - details in appendix Sec **??**.

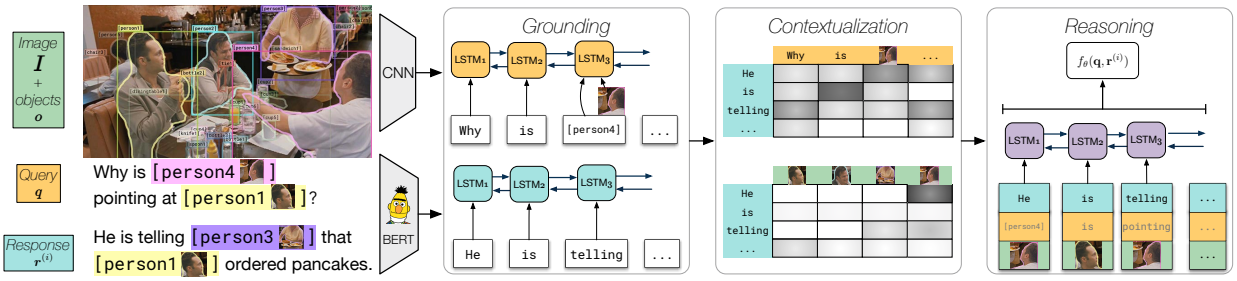


Figure 4.5: High-level overview of our model, **R2C**. We break the challenge of Visual Commonsense Reasoning into three components: grounding the query and response, contextualizing the response within the context of the query and the entire image, and performing additional reasoning steps on top of this rich representation.

Here, $\lambda>0$ controls the tradeoff between similarity and relevance.[6] To obtain multiple counterfactuals, we perform several bipartite matchings. To ensure that the negatives are diverse, during each iteration we replace the similarity term with the maximum similarity between a candidate response $\boldsymbol{r}_j$ and all responses currently assigned to $\boldsymbol{q}_i$.

Ensuring dataset integrity To guarantee that there is no question/answer overlap between the training and test sets, we split our full dataset (by movie) into 11 folds. We match the answers and rationales invidually for each fold. Two folds are pulled aside for validation and testing.

4.5 Recognition to Cognition Networks

We introduce Recognition to Cognition Networks (**R2C**), a new model for visual commonsense reasoning. To perform well on this task requires a deep understanding of language, vision, and the world. For example, in Figure 4.5, answering '*Why is* PERSON4 *pointing at* PERSON1*?*' requires multiple inference steps. First, we **ground** the meaning of the query and each response, which involves referring to the image for the two people. Second, we **contextualize** the meaning of the query, response, and image together. This step includes resolving the referent 'he,' and

[6] We tuned this hyperparameter by asking crowd workers to answer multiple-choice questions at several thresholds, and chose the value for which human performance is above 90% - details in appendix Sec **??**.

why one might be pointing in a diner. Third, we **reason** about the interplay of relevant image regions, the query, and the response. In this example, the model must determine the social dynamics between PERSON1  and PERSON4  . We formulate our model as three high-level stages: grounding, contextualization, and reasoning, and use standard neural building blocks to implement each component.

In more detail, recall that a model is given an image, a set of objects $\boldsymbol{o}$, a query $\boldsymbol{q}$, and a set of responses $\boldsymbol{r}^{(i)}$ (of which exactly one is correct). The query $\boldsymbol{q}$ and response choices $\boldsymbol{r}^{(i)}$ are all expressed in terms of a mixture of natural language and pointing to image regions: notation-wise, we will represent the object tagged by a word w as o_w. If w isn't a detection tag, o_w refers to the entire image boundary. Our model will then consider each response $\boldsymbol{r}$ separately, using the following three components:

Grounding The grounding module will learn a joint image-language representation for each token in a sequence. Because both the query and the response contain a mixture of tags and natural language words, we apply the same grounding module for each (allowing it to share parameters). At the core of our grounding module is a bidirectional LSTM [119] which at each position is passed as input a word representation for w_i, as well as visual features for o_{w_i}. We use a CNN to learn object-level features: the visual representation for each region o is Roi-Aligned from its bounding region [217, 109]. To additionally encode information about the object's class label ℓ_o, we project an embedding of ℓ_o (along with the object's visual features) into a shared hidden representation. Let the output of the LSTM over all positions be $\mathbf{r}$, for the response and $\mathbf{q}$ for the query.

Contextualization Given a grounded representation of the query and response, we use attention mechanisms to contextualize these sentences with respect to each other and the image context. For each position i in the response, we will define the attended query representation as $\hat{\mathbf{q}}_i$ using the following equation:

$$\alpha_{i,j} = \operatorname*{softmax}_j(\mathbf{r}_i \mathbf{W} \mathbf{q}_j) \qquad \hat{\mathbf{q}}_i = \sum_j \alpha_{i,j} \mathbf{q}_j. \tag{4.2}$$

To contextualize an answer with the image, including implicitly relevant objects that have not been picked up from the grounding stage, we perform another bilinear attention between the response $\mathbf{r}$

and each object o's image features. Let the result of the object attention be $\hat{\mathbf{o}}_i$.

Reasoning Last, we allow the model to *reason* over the response, attended query and objects. We accomplish this using a bidirectional LSTM that is given as context $\hat{\mathbf{q}}_i$, $\mathbf{r}_i$, and $\hat{\mathbf{o}}_i$ for each position i. For better gradient flow through the network, we concatenate the output of the reasoning LSTM along with the question and answer representations for each timestep: the resulting sequence is max-pooled and passed through a multilayer perceptron, which predicts a logit for the query-response compatibility.

Neural architecture and training details For our image features, we use ResNet50 [108]. To obtain strong representations for language, we used BERT representations [61]. BERT is applied over the entire question and answer choice, and we extract a feature vector from the second-to-last layer for each word. We train **R2C** by minimizing the multi-class cross entropy between the prediction for each response $\boldsymbol{r}^{(i)}$, and the gold label.

4.6 Results

In this section, we evaluate the performance of various models on VCR. Recall that our main evaluation mode is the staged setting ($Q \rightarrow AR$). Here, a model must choose the right answer for a question (given four answer choices), and then choose the right rationale for that question and answer (given four rationale choices). If it gets either the answer or the rationale wrong, the entire prediction will be wrong. This holistic task decomposes into two sub-tasks wherein we can train individual models: question answering ($Q \rightarrow A$) as well as answer justification ($QA \rightarrow R$). Thus, in addition to reporting combined $Q \rightarrow AR$ performance, we will also report $Q \rightarrow A$ and $QA \rightarrow R$.

Task setup A model is presented with a query $\boldsymbol{q}$, and four response choices $\boldsymbol{r}^{(i)}$. Like our model, we train the baselines using multi-class cross entropy between the set of responses and the label. Each model is trained separately for question answering and answer justification.[7]

[7]We follow the standard train, val and test splits.

		$Q \rightarrow A$		$QA \rightarrow R$		$Q \rightarrow AR$	
	Model	Val	Test	Val	Test	Val	Test
	Chance	25.0	25.0	25.0	25.0	6.2	6.2
Text Only	BERT	53.8	53.9	64.1	64.5	34.8	35.0
Text Only	BERT (response only)	27.6	27.7	26.3	26.2	7.6	7.3
Text Only	ESIM+ELMo	45.8	45.9	55.0	55.1	25.3	25.6
Text Only	LSTM+ELMo	28.1	28.3	28.7	28.5	8.3	8.4
VQA	RevisitedVQA [129]	39.4	40.5	34.0	33.7	13.5	13.8
VQA	BottomUpTopDown[14]	42.8	44.1	25.1	25.1	10.7	11.0
VQA	MLB [138]	45.5	46.2	36.1	36.8	17.0	17.2
VQA	MUTAN [21]	44.4	45.5	32.0	32.2	14.6	14.6
	R2C	**63.8**	**65.1**	**67.2**	**67.3**	**43.1**	**44.0**
	Human		91.0		93.0		85.0

Table 4.1: Experimental results on VCR. VQA models struggle on both question-answering ($Q \rightarrow A$) as well as answer justification ($Q \rightarrow AR$), possibly due to the complex language and diversity of examples in the dataset. While language-only models perform well, our model **R2C** obtains a significant performance boost. Still, all models underperform human accuracy at this task. For more up-to-date results, see the leaderboard at https://visualcommonsense.com/leaderboard.

4.6.1 Baselines

We compare our **R2C** to several strong language and vision baselines.

Text-only baselines We evaluate the level of visual reasoning needed for the dataset by also evaluating purely text-only models. For each model, we represent $\boldsymbol{q}$ and $\boldsymbol{r}^{(i)}$ as streams of tokens, with the detection tags replaced by the object name (e.g. `chair5` $\rightarrow$ `chair`). To minimize the discrepancy between our task and pretrained models, we replace person detection tags with gender-neutral names.

a. **BERT** [61]: BERT is a recently released NLP model that achieves state-of-the-art performance on many NLP tasks.

b. **BERT (response only)** We use the same BERT model, however, during fine-tuning and testing the model is only given the response choices $\boldsymbol{r}^{(i)}$.

c. **ESIM+ELMo** [42]: ESIM is another high performing model for sentence-pair classification tasks, particularly when used with ELMo embeddings [205].

d. **LSTM+ELMo**: Here an LSTM with ELMo embeddings is used to score responses $\boldsymbol{r}^{(i)}$.

VQA Baselines Additionally we compare our approach to models developed on the VQA dataset [15]. All models use the same visual backbone as **R2C** (ResNet 50) as well as text representations (GloVe; [203]) that match the original implementations.

e. **RevisitedVQA** [129]: This model takes as input a query, response, and image features for the entire image, and passes the result through a multilayer perceptron, which has to classify 'yes' or 'no'.[8]

f. **Bottom-up and Top-down attention** (BottomUpTopDown) [14]: This model attends over region proposals given by an object detector. To adapt to VCR, we pass this model object regions referenced by the query and response.

g. **Multimodal Low-rank Bilinear Attention** (MLB) [138]: This model uses Hadamard products to merge the vision and language representations given by a query and each region in the image.

h. **Multimodal Tucker Fusion** (MUTAN) [21]: This model expresses joint vision-language context in terms of a tensor decomposition, allowing for more expressivity.

We note that BottomUpTopDown, MLB, and MUTAN all treat VQA as a multilabel classification over the top 1000 answers [14, 172]. Because VCR is highly diverse, for these models we represent each response $\boldsymbol{r}^{(i)}$ using a GRU [46].[9] The output logit for response i is given by the dot product between the final hidden state of the GRU encoding $\boldsymbol{r}^{(i)}$, and the final representation from the model.

---

[8] For VQA, the model is trained by sampling positive or negative answers for a given question; for our dataset, we simply use the result of the perceptron (for response $\boldsymbol{r}^{(i)}$) as the i-th logit.

[9] To match the other GRUs used in [14, 138, 21] which encode $\boldsymbol{q}$.

Model	$Q \rightarrow A$	$QA \rightarrow R$	$Q \rightarrow AR$
R2C	**63.8**	**67.2**	**43.1**
No query	48.3	43.5	21.5
No reasoning module	63.6	65.7	42.2
No vision representation	53.1	63.2	33.8
GloVe representations	46.4	38.3	18.3

Table 4.2: Ablations for **R2C**, over the validation set. 'No query' tests the importance of integrating the query during contextualization; removing this reduces $Q \rightarrow AR$ performance by 20%. In 'no reasoning', the LSTM in the reasoning stage is removed; this hurts performance by roughly 1%. Removing the visual features during grounding, or using GloVe embeddings rather than BERT, lowers performance significantly, by 10% and 25% respectively.

Human performance We asked five different workers on Amazon Mechanical Turk to answer 200 dataset questions from the test set. A different set of five workers were asked to choose rationales for those questions and answers. Predictions were combined using a majority vote.

4.6.2 Results and Ablations

We present our results in Table 4.1. Of note, standard VQA models struggle on our task. The best model, in terms of $Q \rightarrow AR$ accuracy, is MLB, with 17.2% accuracy. Deep text-only models perform much better: most notably, BERT [61] obtains 35.0% accuracy. One possible justification for this gap in performance is a bottlenecking effect: whereas VQA models are often built around multilabel classification of the top 1000 answers, VCR requires reasoning over two (often long) text spans. Our model, **R2C** obtains an additional boost over BERT by 9% accuracy, reaching a final performance of 44%. Still, this figure is nowhere near human performance: 85% on the staged task, so there is significant headroom remaining.

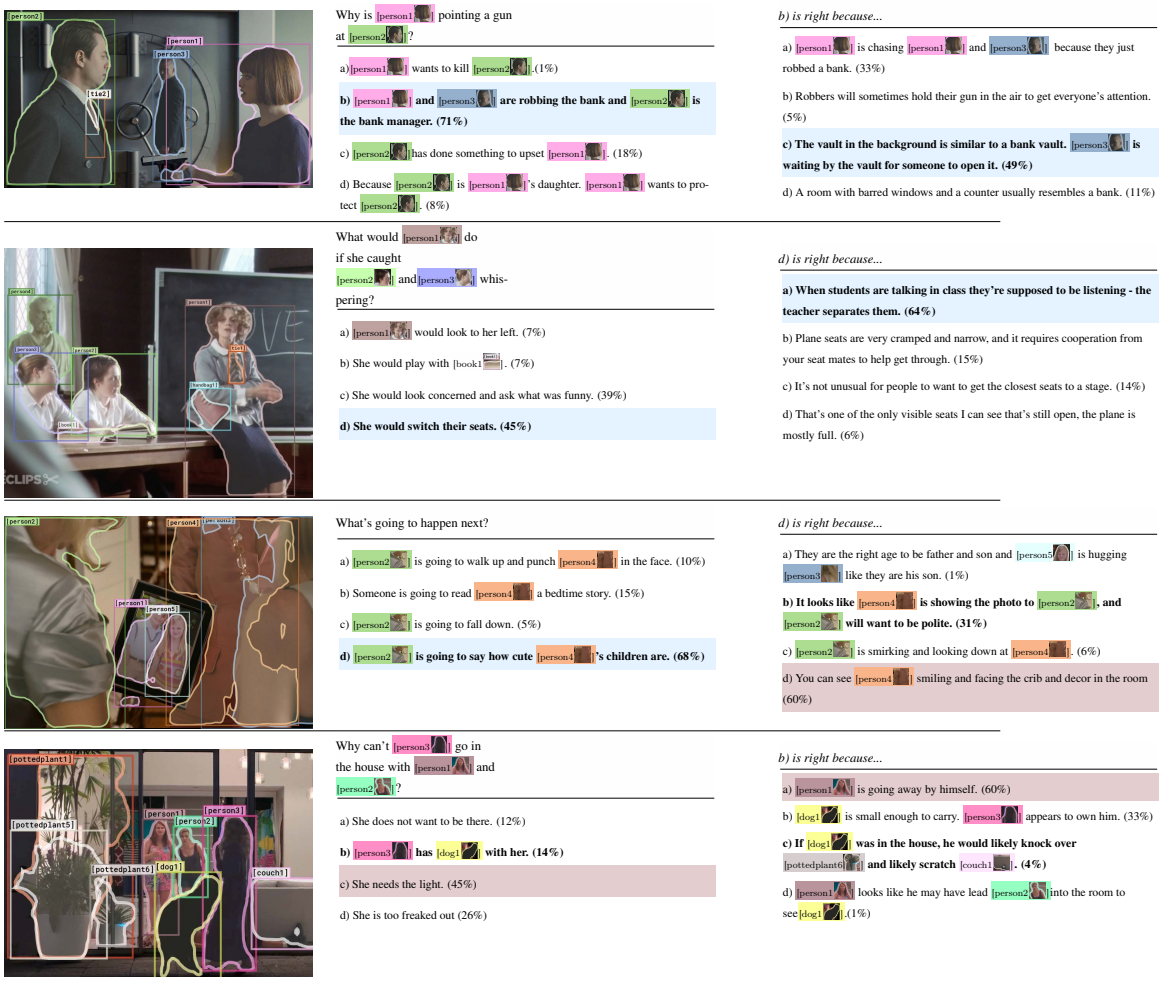


Figure 4.6: Qualitative examples from **R2C**. Correct predictions are highlighted in blue. Incorrect predictions are in red with the correct choices **bolded**. For more predictions, see see `visualcommonsense.com/explore`.

Ablations We evaluated our model under several ablations to determine which components are most important. Removing the query representation (and query-response contextualization entirely) results in a drop of 21.6% accuracy points in terms of $Q \rightarrow AR$ performance. Interestingly, this setting allows it to leverage its image representation more heavily: the text based response-only models (BERT response only, and LSTM+ELMo) perform barely better than chance. Taking the

reasoning module lowers performance by 1.9%, which suggests that it is beneficial, but not critical for performance. The model suffers most when using GloVe representations instead of BERT: a loss of 24%. This suggests that strong textual representations are crucial to VCR performance.

Qualitative results Last, we present qualitative examples in Figure 4.6. **R2C** works well for many images: for instance, in the first row, it correctly infers that a bank robbery is happening. Moreover, it picks the right rationale: even though all of the options have something to do with 'banks' and 'robbery,' only **c)** makes sense. Similarly, analyzing the examples for which **R2C** chooses the right answer but the wrong rationale allows us to gain more insight into its understanding of the world. In the third row, the model incorrectly believes there is a crib while assigning less probability mass on the correct rationale - that PERSON2  is being shown a photo of PERSON4  's children, which is why PERSON2  might say how cute they are.

4.7 Related Work

Question Answering Visual Question Answering [15] was one of the first large-scale datasets that framed visual understanding as a QA task, with questions about COCO images [171] typically answered with a short phrase. This line of work also includes 'pointing' questions [149, 324] and templated questions with open ended answers [294]. Recent datasets also focus on knowledge-base style content [272, 283]. On the other hand, the answers in VCR are entire sentences, and the knowledge required by our dataset is largely background knowledge about how the world works.

Recent work also includes movie or TV-clip based QA [253, 181, 162]. In these settings, a model is given a video clip, often alongside additional language context such as subtitles, a movie script, or a plot summary. In contrast, VCR features no extra language context besides the question. Moreover, the use of explicit detection tags means that there is no need to perform person identification [223] or linkage with subtitles.

An orthogonal line of work has been on referring expressions: asking to what image region a natural language sentence refers to [209, 183, 222, 295, 296, 208, 123, 114]. We explicitly avoid referring expression-style questions by using indexed detection tags (like PERSON1 ).

Last, some work focuses on commonsense phenomena, such as 'what if' and 'why' questions [265, 207]. However, the space of commonsense inferences is often limited by the underlying dataset chosen (synthetic [265] or COCO [207] scenes). In our work, we ask commonsense questions in the context of rich images from movies.

Explainability AI models are often right, but for questionable or vague reasons [26]. This has motivated work in having models provide explanations for their behavior, in the form of a natural language sentence [113, 38, 139] or an attention map [115, 124, 127]. Our rationales combine the best of both of these approaches, as they involve both natural language text as well as references to image regions. Additionally, while it is hard to evaluate the quality of generated model explanations, choosing the right rationale in VCR is a multiple choice task, making evaluation straightforward.

Commonsense Reasoning Our task unifies work involving reasoning about commonsense phenomena, such as physics [194, 289], social interactions [7, 263, 48, 100], procedure understanding [320, 8] and predicting what might happen next in a video [242, 69, 322, 264, 72, 218, 291].

Adversarial Datasets Past work has proposed the idea of creating adversarial datasets, whether by balancing the dataset with respect to priors [95, 103, 216] or switching them at test time [3]. Most relevant to our dataset construction methodology is the idea of Adversarial Filtering [303] (which was also introduced in this thesis!). Correct answers are human-written, while wrong answers are chosen from a pool of machine-generated text that is further validated by humans. However, the correct and wrong answers come from fundamentally different sources, which raises the concern that models can cheat by performing authorship identification rather than reasoning over the image. In contrast, in Adversarial Matching, the wrong choices come from the exact same distribution as the right choices, and no human validation is needed.

4.8 Conclusion

In this section of the thesis, we introduced Visual Commonsense Reasoning, along with a large dataset VCR for the task that was built using Adversarial Matching. We presented **R2C**, a model

for this task, but the challenge – of cognition-level visual undertanding – is far from solved.

Chapter 5

BENCHMARKING GROUNDEDNESS BEYOND EXAMINATIONS

This chapter contains material that was originally published in [309].

5.1 Introduction

Language models today are getting ever-larger, and are being trained on ever-increasing quantities of text. For an immense compute cost, these models like T5 [215] and GPT3 [33] show gains on a variety of standard NLP benchmarks – often even *outperforming* humans.

Yet, when a giant model like T5 generates language, we observe clear gaps between machine-level and human-level language understanding – even after it has been finetuned for the task at hand. Consider Figure 5.1, in which a woman asks for advice. She is assigned to dissect an animal for her class project, but has extreme anxiety about dead animals – and her teacher refused to give her another assignment. Humans can respond with helpful advice, reflecting our unique ability of *real-world language use*: to communicate and tackle open-ended issues. The helpful advice in this example - but not the only one possible - suggests that she send a short email to her guidance counselor.

On the other hand, not only is T5's advice unhelpful, it also reveals key misunderstandings of the situation. It seems to believe that the *student* is asking the *teacher* to do a class project involving dead animals. This reading comprehension error is particularly strange, as T5 outperforms humans on a variety of reading comprehension benchmarks. Others in the community have observed similar issues, raising concerns about what today's benchmark datasets measure [290, 151, 187, 81].

We argue that there is a deep underlying issue: a gap between how humans use language in the real world, and what benchmarks today can measure. Today's dominant paradigm is to study static datasets, and to grade machines by the similarity of their output with predefined *correct* answers.

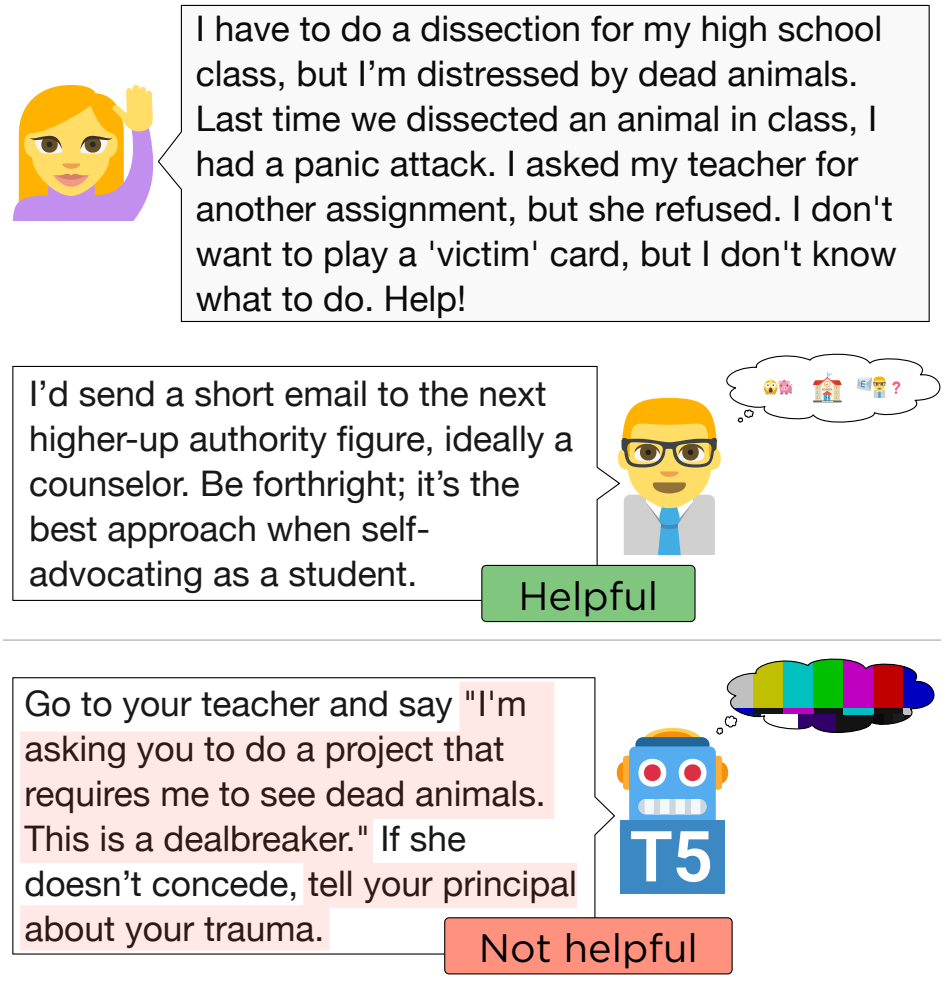


Figure 5.1: TURINGADVICE. Humans are natural experts at *using* language to successfully address situations that arise, such as giving advice. We introduce a new framework, dataset, and leaderboard to generatively evaluate real-world language use. Today's most powerful models – which obtain near-human or superhuman performance on core NLP benchmarks for reading comprehension, natural language inference, and commonsense reasoning – struggle with all of these capabilities when generating advice, as highlighted in red.

For example, we score multiple choice exams by how often the *correct* answers are chosen, and evaluate generative tasks like machine translation by similarity with respect to *correct* translations.

However, when we use language in the real world to communicate with each other – such as when we give advice, or teach a concept to someone – there is rarely a universal *correct* answer to compare with, just a loose goal we want to achieve.

We introduce a framework to narrow this gap between benchmarks and real-world language use. We propose to evaluate machines by their success in using language to (1) communicate with humans in (2) tackling complex, open-ended, real-world situations. Our goal is a machine that, like a human, can generate language that is useful and helpful. Doing so necessarily requires a deep understanding of language and the world, as per a line of thought that the complete meaning representation is one that suffices to complete a task [16].

As a case-study of our framework, we introduce TURINGADVICE as a new grand challenge for AI systems. A machine reads a situation written by a person seeking advice, like Figure 5.1, and must then write advice that is helpful to the advice-seeker. Like a Turing Test [257], we establish a simple condition required for a model to 'pass': model-generated advice must be *at least as helpful to the advice-seeker* as human-written advice.

We make our challenge concrete by introducing a new dataset, REDDITADVICE, and accompanying leaderboard. We tie our dataset to the Reddit community, which resolves two additional sources of bias. First, Reddit users are intrinsically motivated, seeking advice about highly complex *real* issues – which past work suggests differ from *hypothetical* issues that crowd workers might come up with (e.g. 154, 102). Second, we make our dataset *dynamic*, not static – models are evaluated over Reddit situations posted over the previous two weeks at the time of submission. Models therefore, like humans, must generalize to new situations and patterns of language.

Experimental results show that TURINGADVICE is incredibly challenging for NLP models. Today's largest finetunable model, T5 with 11 billion parameters, produces advice that is preferable to human-written advice 14.5% of the time – after being finetuned on 600k examples. GPT3, an even larger model with 175 billion parameters that was not released for finetuning, does even worse at 4%. Even more concerning, our evaluation finds that it often generates hateful and toxic language.

We also study our task from the perspective of today's standard 'core' NLP tasks. Broadly,

we find that machines frequently confuse who is who, are self-contradictory, or seem to miss important world knowledge. However, these mistakes tend not to fall into the neat categories defined by standard task definitions. We address this by introducing diagnostic questions, which systematically measure these language understanding errors.

In summary, this section of the thesis makes three contributions. **First**, we introduce a new framework for measuring language understanding through directly tackling real-world language problems. **Second**, we introduce TURINGADVICE as a new challenge for AI systems, along with a dynamic dataset and leaderboard. **Third**, we connect our task to existing atomic NLP tasks, introducing a new setting that reveals where progress is still needed.

5.2 Real World Language Use

We propose to evaluate machines by their success at *real-world language use*: using language to communicate with a human, in response to a naturally occurring situation, in order to achieve a desired outcome. This is how educators often measure (human) language understanding of a second language – by how well the learner can *use* the language [52]. Our approach is also inspired by Wittgenstein's notion of semantics, that "meaning is use:" language is grounded in our desire to make sense of one another and cooperate to meet our needs [281].

As machines do not have humanlike needs or desires, we propose to evaluate machines' success at a task by how well it serves a human who is interested in the outcome. For example, if a machine orders food on my behalf, then I can evaluate it based on whether I enjoy the dish it ordered. Though this requires careful task selection in order to make things feasible for current models, as we will show in Section 5.3, it results in a powerful and reliable human evaluation.

5.2.1 Related work

Pragmatics in NLP

Our evaluation relates to pragmatics in NLP, where communication is modeled also through listeners and speakers [88, 76]. One approach is to introduce a communication game, with an explicit

objective. For example, Wang et al. [273] study a blocks world where humans give commands to a block-placing machine. The machine is then graded on accuracy. Our proposed evaluation instead covers complex everyday scenarios faced by a human, where the objective is to help them as much as possible.

Pragmatics can also be studied through machine-machine communication; e.g., through emergent language [160]. Recent work uses pretrained question-answering models to evaluate summarization models [41, 232, 71, 260]. However, ensuring that machines communicate in standard English is difficult, as there is usually a more efficient machine-language coding scheme for the task [147].

Two major approaches for evaluation

Today, we see two major approaches for NLP evaluation, which we discuss below.

Quality of generations. The first approach studies generative tasks like chit-chat dialogue or story-writing, and measures the inherent *quality of generations*, often through attributes such as "sensibleness" and "specificity" (e.g., 262, 107, 2). This approach is orthogonal to ours: though these attributes might be desirable, they are often insufficient to guarantee success at a task.

Correctness. The second (and perhaps more common) approach is to evaluate models through *correctness* over static datasets. For example, machines can be graded by the similarity of their generated translation to *correct* translations,[1] or, by how often they choose the *correct* answer on a multiple choice exam. Many goal-oriented dialogue and semantics tasks are also evaluated in this way, as a model is evaluated by whether it makes the *correct* API call, or produces a *correct* parse.

Since many language tasks cannot be evaluated through correctness, researchers often introduce *proxy tasks* that are easy to evaluate, while (hopefully) correlating with the underlying *true* task. For example, SWAG [303] is a multiple-choice proxy task and dataset introduced to study the *true* task of commonsense reasoning.

However, there are gaps between datasets for proxy tasks (e.g. multiple choice), and the core

[1]Models submitted to the 2019 Conference on Machine Translation were evaluated (by humans) on how well the model's translations agreed with either (1) human-written translations, or, (2) original source text [19].

tasks they seek to represent (e.g. commonsense reasoning), which we discuss in the next sections.

5.2.2 Can language use really *be measured through correctness over proxy tasks?*

When we reduce a complex language task to a simplified setup, with a small label space (like multiple-choice classification), we run the risk of introducing artifacts and biases: patterns that can be exploited in the simplified setup, but that are not representative of the true task [103, 306]. Artifacts can enable machines to even outperform humans at the final benchmark, without solving the underlying task.

While the problem of artifacts has recently taken the spotlight in the NLP community, partially because large Transformers [261] excel at picking up on artifacts, there is a deeper underlying issue. One way to view simplified tasks is that in order to correctly map inputs X to labels Y, a machine must learn a set of attributes A that are representative of the 'true' task. We can upper-bound the information contained by A through the information bottleneck principle of Tishby et al. [254]. An efficient model minimizes the following, for some $\beta > 0$:

$$\min_{p(a|x)} I(X;A) - \beta I(A;Y), \tag{5.1}$$

where I is mutual information. In other words, the model will learn attributes A that maximally compress the inputs X (minimizing $I(X;A)$), while also remaining good predictors of the labels Y (maximizing $I(A;Y)$). However, the label prediction term is bounded by the information (or entropy, H) of the label space:

$$I(A;Y) = H(Y) - H(Y|A) \leq H(Y). \tag{5.2}$$

Thus, for a task with a small label space, there is no guarantee that a model will learn high-information content attributes. Models are in fact encouraged to overfit to dataset artifacts, and to *unlearn* linguistically useful information that is not directly relevant to predicting Y [204].

An alternate approach is to make datasets harder adversarially, so as to have fewer artifacts [303, 306, 161]. However, it might be impossible to make a dataset with *no* artifacts, or to know if one has been created.

Our proposal, to evaluate models by their real-world language use, addresses the information bottleneck issue in two ways. First, when we use language in the real world, the mapping between possible inputs and outputs is often highly complex. For example, the space of possible advice is vast, and many pieces of advice might be *equally helpful* given a situation. Second, we directly tackle language problems, without introducing a correctness-based proxy that machines might overfit to.

5.2.3 Static datasets in a dynamic world

To evaluate performance on a real-world task by means of a dataset, we (implicitly) assume that the dataset is a good representation of the world [256]. This might be questionable when it comes to real-world language use, as static datasets necessarily capture *historic* patterns of language. For instance, syntactic understanding is often evaluated using the Penn Treebank, with news articles from 1989 [184]. However, the world is constantly evolving, along with the language that we use.

To bridge this gap, we propose to evaluate machines by their interactions with humans *in the present*. Models therefore must learn to perform the underlying language task, even for novel situations, rather than fitting to the historic distribution of a fixed test set. We make this notion concrete in the next section, where we introduce a *dynamic* dataset and leaderboard for evaluating advice.

5.3 TURINGADVICE*: a New Challenge for Natural Language Understanding*

As a case study of our framework, we introduce TURINGADVICE, a new challenge task for AI systems to test language understanding. The format is simple: given a situation expressed in natural language, a machine must respond with helpful advice. To pass the challenge, machine-written advice must be at least as helpful to the advice-seeker as human-written advice, in aggregate.

We focus on advice for a few reasons. First, advice-giving is both an important and an everyday task. People ask for and give advice in settings as diverse as *relationship advice* and *tech support* [31]. Thus, we as humans have inherent familiarity with the task, and what it means for advice to

be *helpful* – making it easy to evaluate, as we later show empirically. Moreover, because there are many internet communities devoted to advice-giving, training data is plentiful.

Second, the framework of advice-giving allows us to study subtasks such as reading comprehension and natural language inference (Section 5.5.3); we argue both of these are needed to consistently give good advice. Learning to recognize advice has recently been studied as an NLP task on its own [94], though we are not aware of past work in learning to *generate* advice.

5.3.1 REDDITADVICE*: A dynamic dataset for evaluating advice*

We propose to evaluate models *dynamically*, through new situations and advice that are posted to Reddit. We call our dynamic dataset REDDITADVICE. Many of Reddit's subcommunities (or 'subreddits') are devoted to asking for and giving advice, with subreddits for legal, relationship, and general life advice.[2] During evaluation time, we will retrieve new situations from Reddit as a new test set for models. Workers on Mechanical Turk then grade the model-written advice versus the Reddit-endorsed human-written advice.

How advice-giving works on Reddit

Suppose a Reddit user faces an issue that they are seeking advice about. First, they write up *situation* and post it to an advice-oriented subreddit. Users then reply to the *situation*, offering *advice*.

Importantly, any user can 'upvote' or 'downvote' the advice as well as the situation itself - changing its score slightly. Top-scoring advice is deemed by the wisdom of the crowd as being the most helpful.[3]

---

[2]We use advice from the following subreddits: `Love`, `Relationships`, `Advice`, `NeedAdvice`, `Dating_Advice`, `Dating`, `Marriage`, `InternetParents`, `TechSupport`, and `LegalAdvice`.

[3]This is somewhat of a simplification, as other factors also influence what gets upvoted [13, 157, 195, 130].

The ideal evaluation - through Reddit?

In a sense, human advice-givers are 'evaluated' on Reddit by the score of their advice – representing how well their advice has been received by the community. Similarly, the *ideal* model evaluation might be to post advice on Reddit directly. If the model writes helpful advice, it should be upvoted.

However, there is a significant ethical problem with this approach. The users who post advice questions are real people, with real problems. A user might read advice that was originally written by a machine, think it was human-endorsed, and do something harmful as a result. For this reason, we take an alternate crowdsourcing approach.

A crowdsourced, hybrid evaluation – through Mechanical Turk

We propose a hybrid approach for *dynamic* evaluation of models. While the situations, and reference advice come from Reddit, we hire workers on Mechanical Turk to rate the relative helpfulness of machine-written advice. Not only is this format more ethical, it also lets us collect diagnostic ratings, allowing us to quantitatively track the natural language understanding errors made by machines. We made our crowdsourcing task as fulfilling as possible - using popular situations from Reddit, and pitching the work in terms of helping people. We received feedback from many workers that our tasks were entertaining and fun, suggesting that our workers are to some degree intrinsically motivated.

Mechanical Turk annotation setup

In a single round of evaluation, we retrieve 200 popular Reddit situations that were posted in the last two weeks. For each situation, we retrieve the top-rated advice from Reddit, and generate one piece of advice per model. Workers on Mechanical Turk then compare the helpfulness of the model-generated advice with human-written advice, and provide diagnostic ratings.

We show an overview of our Mechanical Turk task in Figure 5.2. A worker is given a situation and two pieces of advice. One is the top-scoring advice from Reddit, and the other is model-

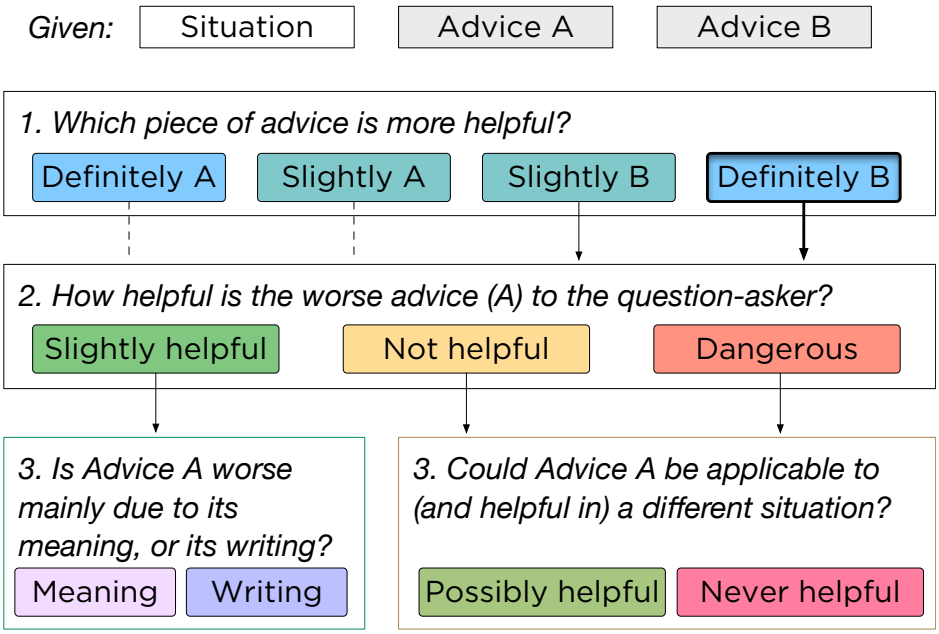


Figure 5.2: Crowdsourcing workflow. Mechanical Turk Workers are given a situation, and two pieces of advice. First, they choose which is more helpful (here, B). Second, they rate the helpfulness of the worse advice (A); last, they answer a diagnostic question.

generated advice; the worker is not told which is which.

The worker first chooses the more helpful piece of advice, then provides diagnostic information for the less helpful advice – rating it SLIGHTLY HELPFUL, NOT HELPFUL, or DANGEROUS. If the worse piece of advice was SLIGHTLY HELPFUL, they choose whether it is worse due to a MEANING PROBLEM or a WRITING PROBLEM. Otherwise, they choose if the worse advice could be POSSIBLY HELPFUL in some other situation, or NEVER HELPFUL in any situation.

Three workers rate each model-situation pair, and ratings are combined using a majority vote. We follow best practices on Mechanical Turk, using a qualification exam, paying workers at least $15 per hour, and giving feedback to workers. Still, evaluation is highly economical at $1.86 per example-model pair, or roughly $400 per model evaluated.

5.3.2 A large static dataset for training

We present RedditAdvice2019, a large static dataset for training advice-giving models. Because today's models have extreme reliance on data for finetuning, we collect data that is in the exact same format as REDDITADVICE, yet we expand our selection criteria, optimizing for recall rather than precision. In total, we extract 616k pieces of advice, over 188k situations.

To mirror the dynamic nature of the evaluation, in which models are evaluated on situations posted in 2020 and beyond, we split our dataset into static training and validation sets by date.[4]

5.4 Experimental Results on REDDITADVICE

In this section, we report results from one round of dynamic evaluation on REDDITADVICE. We evaluate the following strong NLP models and baselines:

a. Rule-based: a templated system to give legal, relationship, or life advice. The system first chooses randomly empathetic sentence from ten choices, for example "I'm sorry you're facing this." It then chooses a random piece of advice that is loosely related to the situation's topic; we infer this from the subreddit the situation was posted on. For example, for `LegalAdvice` the model might write "I'd suggest getting a lawyer immediately."

b. TF-IDF retrieval: for a new situation, we compute its TF-IDF bag-of-word vector and use it to retrieve the most similar situation from the training set. We then reply with the top-scoring advice for that situation.

c. Grover-Mega [308]: a left-to-right transformer model with 1.5 billion parameters. Grover was pretrained on news articles with multiple fields, perhaps making it a good fit for our task, with multiple fields of context (like the subreddit, date, and title). Our situation-advice pairs are often quite long, so we adapt Grover for length; pretraining it on sequences of up to 1536 characters.

d. T5 [215]: a sequence-to-sequence model with a bidirectional encoder and a left-to-right gener-

[4]Our training set contains 600k pieces of advice from July 2009 to June 14, 2019; validation contains 8k from June 14 to July 9th 2019.

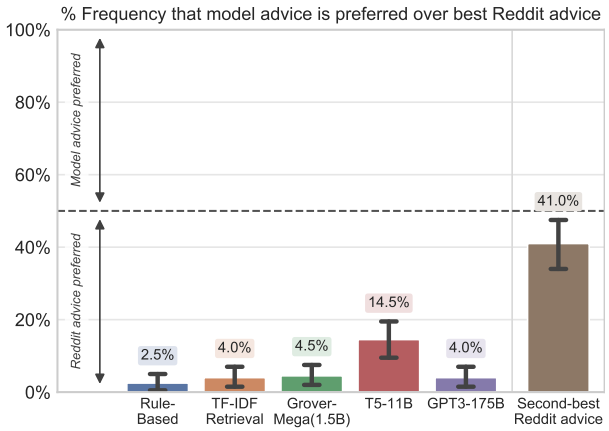


Figure 5.3: Helpfulness of models relative to top-scoring Reddit advice. We show results over 200 shared situations; we also show bootstrapped 95% confidence intervals. Advice from the best-scoring model, T5-11B, is preferred 14.5% over top-scoring Reddit advice. We also compare the second-top scoring piece of Reddit advice, which scores 41% – worse than the best advice (50% by definition), but better than any model.

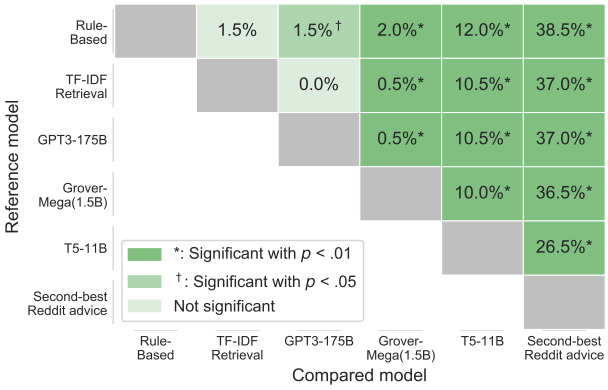


Figure 5.4: Improvement (in absolute percentage %) between pairs of models, along with statistical significance from a paired t-test. The improvement of T5-11B over smaller models like Grover-Mega is highly statistically significant (10% gap, $p<.01$), while being far worse than human performance. Our evaluation thus meaningfully grades varying levels of performance.

ator, with 11 billion parameters. T5 was trained on a large dataset of cleaned web text. At the time of writing, T5 is the top-scoring model on the Glue and SuperGlue benchmarks [269, 268], scoring above human performance on Glue and near human-performance on SuperGlue.

e. GPT3 [33]: a left-to-right transformer model with 175 billion parameters. GPT3 must be "prompted" to generate advice since it has not been released for finetuning. We cannot provide few-shot examples in the prompt due to the length of situation-advice pairs; we instead mimic the formatting of a website quoting from Reddit (Appendix **??**).

Last, to quantify the measurement error of our evaluation, we additionally evaluate:

f. the *second*-highest rated Reddit advice for each situation. We send this advice through the same pipeline as machine-written advice.

We finetune all models (except GPT3) and generate using Nucleus Sampling [121].

In our study, we exclude purely bidirectional models, such as BERT [61]. While these models can be made to generate text, these generations are usually worse than those of left-to-right models [267]. T5 also tends to outperform them, even on discriminative tasks.

5.4.1 Quantitative results

In Figure 5.3, we show overall results for one evaluation trial, which featured 200 situations posted on Reddit from October 28 to November 7, 2020. As a key metric for measuring the relative usefulness of model-written advice, we evaluate the frequency by which workers prefer the Reddit-written reference advice over the model-written advice. If a model's advice was just as helpful as human advice in aggregate, then that model would score 50%.

Model performance is quite low. The best model, T5-11B, scores 14.5%, outperforming a smaller Grover-Mega (4.5%); GPT3 does worse at 4.0%. The rule-based and TF-IDF baselines are competitive at 2.5% and 4.0% accuracy respectively.

As additional comparison to the 50% upper bound, the second-highest scoring Reddit advice scores 41%. This suggest that our workers and often prefer the same advice as Reddit users.

Measurement error

To investigate the measurement error of our evaluation, in Figure 5.4 we report the statistical significance between pairs of models. We observe a large gap in performance between T5 and the other baselines. For example, its improvement over Grover-Mega is 10%, which is highly statistically signficant. On the other hand, the differences in performance between other models are more minor – GPT3 does not outperform TF-IDF, and though it outperforms the rule-based system by 1.5%, it is only somewhat statistically significant.

Overall, the statistical significance results suggest that our evaluation can stably rank model

performance. This, along with the finding that model performance is low on our task suggests that there is ample room for growth on REDDITADVICE.

5.5 Analysis and Discussion

So far, we have shown that we are able to reliably evaluate models in our dynamic setup, and that doing so results in model performance that is significantly lower than human performance.

To break down what this gap in performance means, we show a qualitative example in Figure 5.5. A user is asking for online legal advice about being stuck at work for their entire 4pm-midnight shift – with no eating allowed due to COVID-19. The top-rated Reddit advice understands this situation and then offers advice, suggesting the advice-seeker organize alongside other workers – as "New Jersey doesn't require employers to give any meal breaks."

Machine advice seems to misunderstand the issue. T5 asks if there is "a reason that you can't leave the property," even though this reason is stated in the situation. GPT3's advice is self-contradictory; it also makes something up about a "restaurant" in the building.

5.5.1 Problems with machine-written advice

As part of our evaluation, we wish to quantitatively measure problems with machine-written advice. Recall that in our crowdsourcing setup (Section 5.3.1), three workers select which advice is better, and then annotate problems with the worse piece of advice. We found workers had high agreement during the diagnostic annotation.[5]

In Figure 5.6, we show the distribution of the ratings for model-written, versus human-written advice. Machine-written advice that was *not preferred* over human-written advice can have the following ratings. It can be rated as SLIGHTLY HELPFUL (but, was rated as worse mainly due to a MEANING PROBLEM or WRITING PROBLEM), as NOT HELPFUL, or DANGEROUS.

[5]For the classifying machine-written advice as 'helpful' versus 'not helpful' or 'dangerous' (combining the two latter categories into one), we have κ=0.689. For breaking down helpful advice into 'meaning problem' versus a 'writing problem', we have Cohen's κ=0.613; for rating unhelpful advice as 'possibly helpful' versus 'never helpful,' we have κ=0.602.

	`legaladvice`: NJ Work will not let us eat in the building due to covid outbreak. We can't leave the property for breaks. They have a tent outside for us to eat in but it's getting cold out. I am a manager at my work part of my responsibilities is to stay on the property during my entire shift we are given a 1 hour paid lunch break that we do not have to clock out for. However due to recent covid outbreak the new policy is no eating in the building. They had set up a tent outside for us to eat in but it is starting to get cold out as well there are not heaters in there. My question is can they legally do this to us? Also I work 4:00 p.m. to midnight shift.
Top Reddit advice	So by "clock out for" you are not a salaried exempt employee? Do the workers under you have a union? Aside from the cold, New Jersey doesn't require employers to give *any* meal breaks for employees 18 or older, so they could take away the breaks and the tent if they wanted, and just say "no eating" (though water is required). What is your relationship with the other managers, and room to figure this out, ask for heaters, etc.?
Rule Based	If I were you, I'd get a lawyer immediately. DEFINITELY LESS HELPFUL SLIGHTLY HELPFUL MEANING PROBLEM
TF-IDF Retrieval	You were blackout drunk, which means that you were not capable of consent. . . . If anyone should feel violated, it's you. DEFINITELY LESS HELPFUL NOT HELPFUL POSSIBLY HELPFUL
Grover-Mega	If you are required to be on the premises, they are required to provide you with a place to eat. https://www.nj.gov/labor/es/faq.html#13 DEFINITELY LESS HELPFUL NOT HELPFUL POSSIBLY HELPFUL
T5-11B	If your shift is from 4pm to midnight, is there a reason that you can't leave the property? DEFINITELY LESS HELPFUL NOT HELPFUL NEVER HELPFUL
GPT3-175B	A place of employment is considered the "premises". You can't just assume that because you can't walk out the door that you can eat wherever you want. This is work hours. You're not in your home. The restaurant, or at least the food, is all provided for you for your benefit. But if there are other people there, then I could see how it could be a safety issue. DEFINITELY LESS HELPFUL NOT HELPFUL NEVER HELPFUL

Figure 5.5: Qualitative example. Though machine-generated advice matches keywords from the situation, it is frequently not helpful or even self-contradictory. The issues are due to critical errors in natural language understanding, such as reading comprehension, entailment, and coreference.

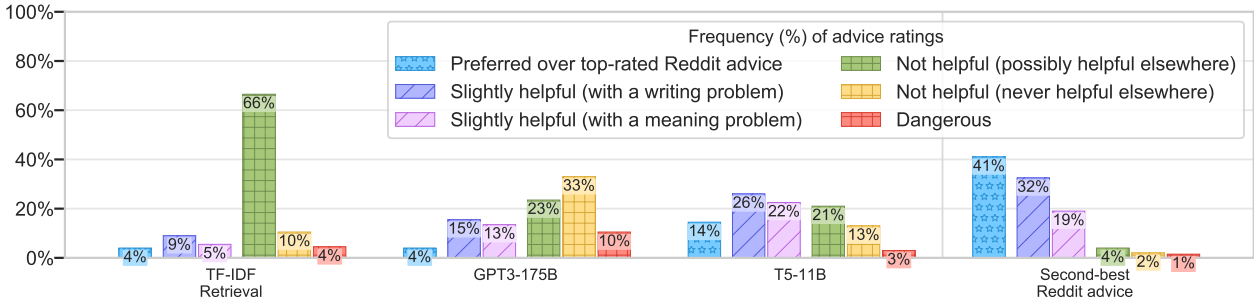


Figure 5.6: Distribution of ratings for three models: TF-IDF retrieval, GPT3, and T5, along with ratings for the second-best rated Reddit advice. Though deep generators like GPT3 and T5 are often preferred over the retrieval baseline, they also often write advice that would never be helpful (33% GPT3, 13% T5), and that is racist, sexist, or otherwise dangerous (10% GPT3, 3% T5).

The diagnostics show several patterns. First, all models frequently commit natural language understanding errors, such as internal contradiction. Because of this, we find that TF-IDF bag-of-words retrieval is competitive with that of large generators. While retrieved advice is often irrelevant (66% of the time), it is almost never complete gibberish, as it comes from top-scoring advice. Only 10% of workers rated this advice as NOT HELPFUL for any situation, less than T5.

Second, they suggest that models struggle even more without finetuning. A GPT3 model with careful prompting generates language that is DANGEROUS 10% of the time. These qualitative and quantitative results confirm a pattern observed by many others, that large language models like GPT3 often generate explicitly racist and sexist language out-of-the-box 238, 82, 23, among others). This is perhaps worrying, since GPT3 is presently being commercialized.

5.5.2 A Leaderboard for Advice Evaluation

So far, we have shown results from one evaluation round; a second is in Supplemental **??**. We propose a *dynamic leaderboard* to keep that evaluation ongoing, at `rowanzellers.com/advice`.

Users submit a model API to be dynamically evaluated. Each new model, along with the highest rated previously-evaluated model, will be evaluated for an additional round using the same

approach. The cost of each evaluation is reasonable (Section 5.3.1), which we authors will pay in the short term. An alternative strategy requires submitters to pay the Mechanical Turk fees themselves; this model was used for the HYPE leaderboard in computer vision [321].

5.5.3 Relation to existing NLP tasks

Shared "core" tasks such as reading comprehension and natural language inference are of considerable interest to the NLP community. Many datasets have been proposed for these tasks, and progress on them is often measured through auto-gradeable correctness metrics. However, large models have started to outperform humans on these datasets, raising doubt that further progress on them brings us closer to human-level language understanding.

We argue two things: first, that many NLP tasks are necessary *components* of giving advice, and second, that because giving advice remains far from solved, these tasks are also far from solved. We can study problems with advice from T5-11B from the point of view of existing NLP tasks. For instance, machine advice often contradicts itself, suggesting that today's systems struggle with the general task of natural language inference. We have made these diagnostics publicly available to enable progress on automatically spotting these mistakes.

5.6 Conclusion; Ethical Considerations

We introduced new methodology for evaluating language tasks, reducing the gap between benchmarks and the real world. We also introduced a new challenge for the community, TURING-ADVICE, with an accompanying dataset and dynamic leaderboard.

Yet, if our field is to progress towards NLP models that 'understand natural language,' we should be cognizant of the impact that such technology might have on society. In this paper, we presented a sketch of NLP models helping people who need advice on sensitive topics, which could be a measurable goal for the field.

At the same time, we do not claim that our approach is a panacea. There are almost certainly better non-technical solutions to ensure mentorship and legal advice for all [97]. Moreover, there

are significant dual-use risks with models that understand language [122, 98]. Our evaluation measures some risks of generative models – such as the tendency to generate toxic language – but more work in this area is needed.

Chapter 6

LANGUAGE GROUNDING THROUGH INTERACTION IN A 3D WORLD

This chapter contains material that was originally published in [310].

6.1 Introduction

As humans, our use of language is linked to the physical world. To process a sentence like "the robot turns on the stove, with a pan on it" (Figure 6.1) we might imagine a physical `Pan` object. This meaning representation in our heads can be seen as a part of our commonsense world knowledge, about what a `Pan` is and does. We might reasonably predict that the `Pan` will become `Hot` – and if there's an `Egg` on it, it would become `Cooked`.

As humans, we learn such a commonsense world model through interaction. Young children learn to reason physically about basic objects by manipulating them: observing the properties they have, and how they change if an action is applied on them [243]. This process is hypothesized to be crucial to how children learn language: the names of these elementary objects become their first "real words" upon which other language is scaffolded [292].

In contrast, the dominant paradigm today is to train large language or vision models on *static data*, such as language and photos from the web. Yet such a setting is fundamentally limiting, as suggested empirically by psychologists' failed attempts to get kittens to learn passively [112]. More recently, though large Transformers have made initial progress on benchmarks, they also have frequently revealed biases in those same datasets, suggesting they might not be solving underlying tasks [306]. This has been argued philosophically by a flurry of recent work arguing that no amount of language *form* could ever specify language *meaning* [187, 22, 28]; connecting back to the Symbol Grounding Problem of Harnad [106].

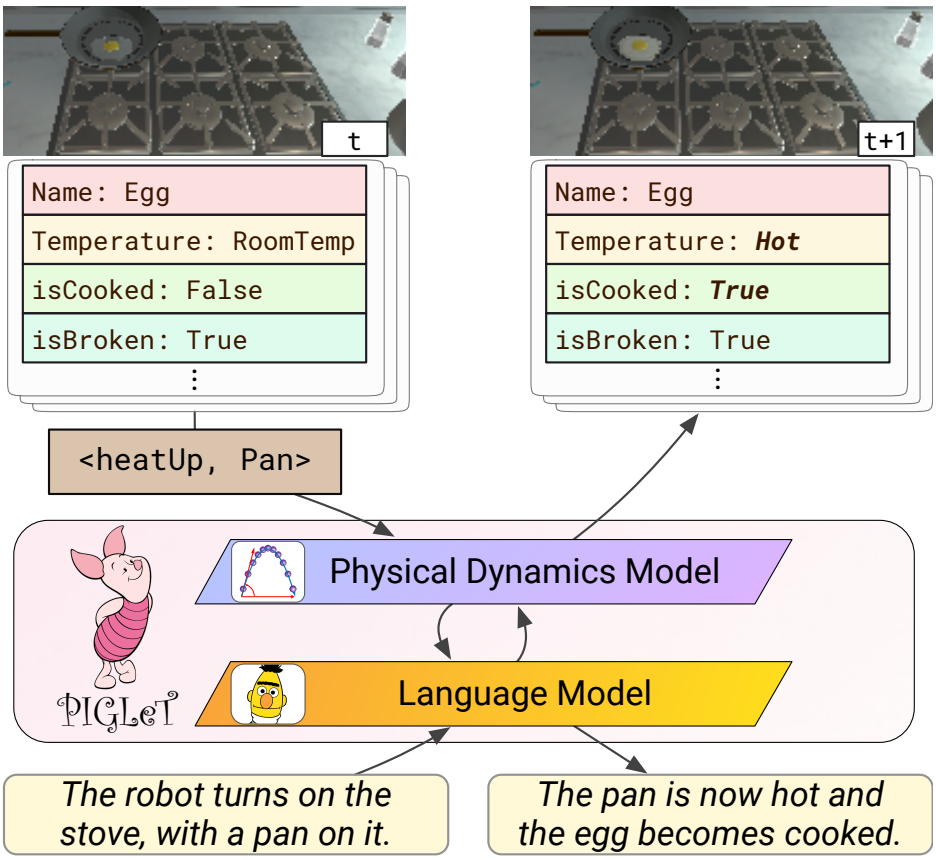


Figure 6.1: PIGLeT. Through physical interaction in a 3D world, we learn a model for what actions do to objects. We use our physical model as an interface for a language model, jointly modeling elements of language *form* and *meaning*. Given an action expressed symbolically or in English, PIGLeT can simulate what might happen next, expressing it symbolically or in English.

In this paper, we investigate an alternate strategy for learning physical commonsense through interaction, and then transferring that into language. We introduce a model named PIGLeT, short for **P**hysical **I**nteraction as **G**rounding for **L**anguag**e** **T**ransformers. We factorize an embodied agent into an explicit model of world dynamics, and a model of language form. We learn the dynamics model through *interaction*. Given an action `heatUp` applied to the `Pan` in Figure 6.1, the model learns that the `Egg` on the pan becomes `Hot` and `Cooked`, and that other attributes do not change.

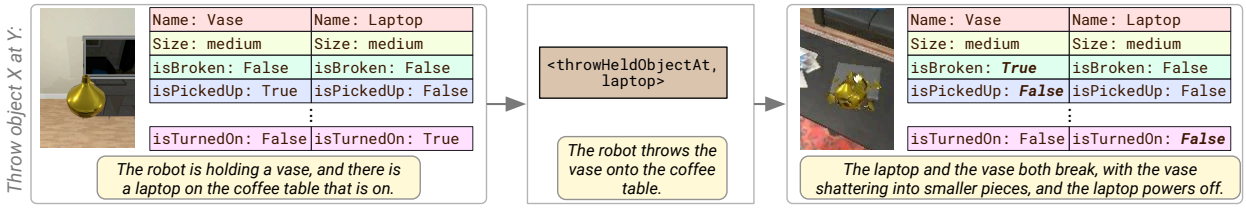


Figure 6.2: PIGPEN, a setting for few-shot language-world grounding. We collect data for 280k physical interactions in THOR, a 3D simulator with 20 actions and 125 object types, each with 42 attributes (e.g. `isBroken`). We annotate 2k interactions with English sentences describing the initial world state, the action, and the action result.

We integrate our dynamics model with a pretrained language model, giving us a joint model of linguistic *form* and *meaning*. The combined PIGLET can then reason about the physical dynamics implied by English sentences describing actions, predicting literally what might happen next. It can then communicate that result either symbolically or through natural language, generating a sentence like 'The egg becomes hot and cooked." Our separation between physical dynamics and language allows the model to learn about physical commonsense from the physical world itself, while also avoiding recurring problems of artifacts and biases that arise when we try to model physical world understanding solely through language.

We study this through a new environment and evaluation setup called PIGPEN, short for **P**hysical **I**nteraction **G**rounding **P**air**e**d with **N**atural Language. In PIGPEN, a model is given unlimited access to an environment for pretraining, but only 500 examples with paired English annotations. Models in our setup must additionally generalize to novel 'unseen' objects for which we intentionally do not provide paired language-environment supervision. We build this on top of the THOR environment [146], a physics engine that enables agents to perform contextual interactions (Fig 6.2) on everyday objects.

Experiments confirm that PIGLET performs well at grounding language with meaning. Given a sentence describing an action, our model predicts the resulting object states correctly over 80% of the time, outperforming even a 100x larger model (T5-11B) by over 10%. Likewise, its gener-

ated natural language is rated by humans as being more correct than equivalently-sized language models. Last, it can generalize in a 'zero-shot' way to objects that it has never read about before in language.

In summary, we make three key contributions. **First**, we introduce PIGLET, a model decoupling physical and linguistic reasoning. **Second**, we introduce PIGPEN, to learn and evaluate the transfer of physical knowledge to the world of language. **Third**, we perform experiments and analysis suggesting promising avenues for future work.

6.2 PIGPEN: A Resource for Neuro-Symbolic Language Grounding

We introduce PIGPEN as a setting for learning and evaluating physically grounded language understanding. An overview is shown in Figure 6.2. The idea is that an agent gets access to an interactive 3D environment, where it can learn about the world through interaction – for example, that objects such as a `Vase` can become `Broken` if thrown. The goal for a model is to learn natural language *meaning* grounded in these interactions.

Task definition. Through interaction, an agent observes the interplay between objects $\boldsymbol{o} \in \mathcal{O}$ (represented by their attributes) and actions $\boldsymbol{a} \in \mathcal{A}$ through the following transition:

$$\underbrace{\{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_N\}}_{\vec{\boldsymbol{o}}, \text{ state pre-action}} \times \boldsymbol{a} \rightarrow \underbrace{\{\boldsymbol{o}'_1, \ldots, \boldsymbol{o}'_N\}}_{\vec{\boldsymbol{o}}', \text{ state post-action}}. \tag{6.1}$$

Actions change the state of a subset of objects: turning on a `Faucet` affects a nearby `Sink`, but it will not change a `Mirror` on the wall.

To encourage learning from interaction, and not just language, an agent is given a small number of natural language annotations of transitions. We denote these sentences as $\boldsymbol{s}_{\vec{o}}$, describing the state pre-action, $\boldsymbol{s}_{\boldsymbol{a}}$ the action, and $\boldsymbol{s}_{\vec{o}'}$ the state post-action respectively. During evaluation, an agent will sometimes encounter new objects $\boldsymbol{o}$ that were not part of the paired training data.

We evaluate the model's transfer in two ways:

a. PIGPEN-NLU. A model is given object states $\vec{\boldsymbol{o}}$, and an English sentence $\boldsymbol{s}_{\boldsymbol{a}}$ describing an action. It must predict the grounded object states $\vec{\boldsymbol{o}}'$ that result after the action is taken.

b. PIGPEN-NLG. A model is given object states $\vec{\boldsymbol{o}}$ and a literal action $\boldsymbol{a}$. It must generate a

sentence $s_{\vec{o}'}$ describing the state post-action.

We next describe our environment, feature representation, and language annotation process.

6.2.1 Environment: THOR

We use AI2-THOR as an environment for this task [146]. In THOR, a robotic agent can navigate around and perform rich contextual interactions with objects in a house. For instance, it can grab an `Apple`, slice it, put it in a `Fridge`, drop it, and so on. The state of the `Apple`, such as whether it is sliced or cold, changes accordingly; this is not possible in many other environments.

In this work, we use the underlying THOR simulator as a proxy for grounded meaning. Within THOR, it can be seen as a 'complete' meaning representation [16], as it fully specifies the kind of grounding a model can expect in its perception within THOR.

Objects. The underlying THOR representation of each object o is in terms of 42 attributes. We treat these attributes as words specific to an attribute-level dictionary; for example, the temperature `Hot` is one of three possible values for an object's temperature; the others being `Cold` and `RoomTemp`.

Actions. An action a in THOR is a function that takes up to two objects as arguments. Actions are highly contextual, affecting not only the arguments but potentially other objects in the scene (Figure 6.2). We also treat action names as words in a dictionary.

Filtering out background objects. Most actions change the state of only a few objects, yet there can be many objects in a scene. We keep annotation and computation tractable by having models predict (and humans annotate) possible changes of at most two key objects in the scene. As knowing when an object *doesn't* change is also important, we include non-changing objects if fewer than two change.

Exploration. Any way of exploring the environment is valid for our task, however, we found that exploring *intentionally* was needed to yield good coverage of interesting states. Similar to prior work for instruction following [241], we designed an oracle to collect diverse and interesting trajectories $\{\vec{o}, a, \vec{o}'\}$. Our oracle randomly selects one of ten high level tasks. These in turn require randomly choosing objects in the scene; e.g. a `Vase` and a `Laptop` in Figure 6.2. We

randomize the manner in which the oracle performs the task to discover diverse situations.

In total, we sampled 20k trajectories. From these we extracted 280k transitions (Eqn 6.1's) where at least one object changes state, for training.

6.2.2 Annotating Interactions with Language

Data Selection for Annotation

We select 2k action state-changes from trajectories held out from the training set. We select them while also balancing the distribution of action types to ensure broad coverage in the final dataset. We are also interested in a model's ability to generalize to new object categories – beyond what it has read about, or observed in a training set. We thus select 30 objects to be "unseen," and exclude these from paired environment-language training data. We sample 500 state transitions, containing only "seen" objects to be the training set; we use 500 for validation and 1000 for testing.

Natural Language Annotation

Workers on Mechanical Turk were shown an environment in THOR *before* and *after* a given action $\boldsymbol{a}$. Each view contains the THOR attributes of the two key objects. Workers then wrote three English sentences, corresponding to $\boldsymbol{s}_{\vec{o}}$, $\boldsymbol{s}_{\boldsymbol{a}}$, and $\boldsymbol{s}_{\vec{o}'}$ respectively. Workers were instructed to write at a particular level of detail: enough so that a reader could infer "what happens next" from $\boldsymbol{s}_{\vec{o}}$ and $\boldsymbol{s}_{\boldsymbol{a}}$, yet without mentioning redundant attributes.

6.3 Modeling PIGLET

In this section, we describe our PIGLET model. **First**, we learn a neural physical dynamics model from interactions, and **second**, integrate with a pretrained model of language form.

6.3.1 Modeling Physical Dynamics

We take a neural, auto-encoder style approach to model world dynamics. An object $\boldsymbol{o}$ gets encoded as a vector $\mathbf{h}_{\boldsymbol{o}} \in \mathbb{R}^{d_{\text{o}}}$. The model likewise encodes an action $\boldsymbol{a}$ as a vector $\mathbf{h}_{\boldsymbol{a}} \in \mathbb{R}^{d_{\text{a}}}$, using

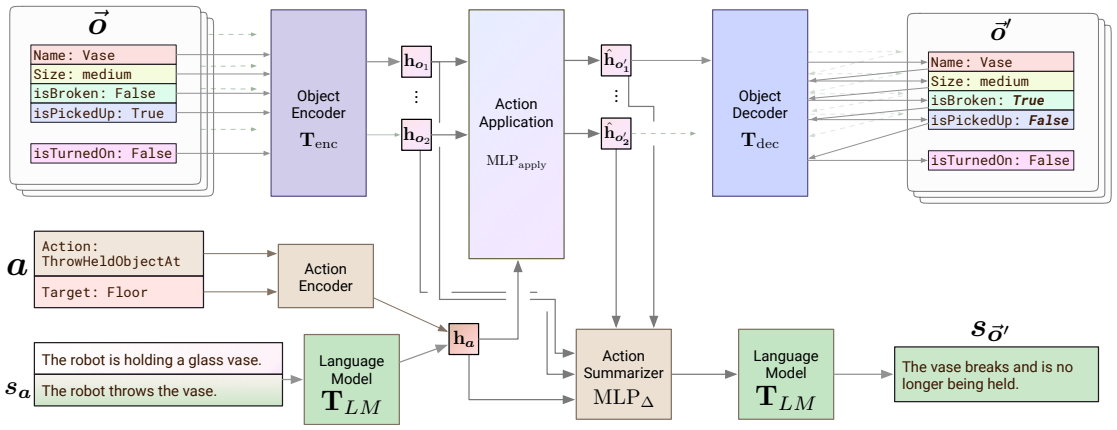


Figure 6.3: PIGLET architecture. We pretrain a model of physical world dynamics by learning to transform objects $\vec{o}$ and actions a into new updated objects $\vec{o}'$. Our underlying world dynamics model – the encoder, the decoder, and the action application module, can augment a language model with grounded commonsense knowledge.

it to manipulate the hidden states of all objects. The model can then decode any object hidden representation back into a symbolic form.

Object Encoder and Decoder

We use a Transformer [261] to encode objects into vectors $\mathbf{o} \in \mathbb{R}^{d_o}$, and then another to decode from this representation.

Encoder. Objects $\boldsymbol{o}$ are provided to the encoder as a set of attributes, with categories $c_1, ..., c_n$. Each attribute c has its own vocabulary and embedding $\mathbf{E}_c$. For each object $\boldsymbol{o}$, we first embed all the attributes separately and feed the result into a Transformer encoder T_{enc}. This gives us (with position embeddings omitted for clarity):

$$\mathbf{h}_{\boldsymbol{o}} = \mathbf{T}_{\text{enc}}\Big(\mathbf{E}_1(o_1), \ldots, \mathbf{E}_{c_n}(o_{c_n})\Big) \tag{6.2}$$

Decoder. We can then convert back into the original symbolic representation through a left-to-right Transformer decoder, which predicts attributes one-by-one from c_1 to c_n. This captures the inherent correlation between attributes, while making no independence assumptions. The probability of

predicting the next attribute $o_{c_{i+1}}$ is then given by:

$$p(o_{c_{i+1}}|\mathbf{h}_{\boldsymbol{o}}, \boldsymbol{o}_{:c_i}) = \mathbf{T}_{\text{dec}}\Big(\mathbf{h}_{\boldsymbol{o}}, \mathbf{E}_1(o_1), ..., \mathbf{E}_{c_i}(o_{c_i})\Big) \tag{6.3}$$

Modeling actions as functions

We treat actions $\boldsymbol{a}$ as functions that transform the state of all objects in the scene. Actions in our environment take at most two arguments, so we embed the action $\boldsymbol{a}$ and the names of its arguments, concatenate them, and pass the result through a multilayer perceptron; yielding a vector representation $\mathbf{h}_{\boldsymbol{a}}$.

Applying Actions. We use the encoded action $\mathbf{h}_{\boldsymbol{a}}$ to transform all objects in the scene, obtaining updated representations $\hat{\mathbf{h}}_{\boldsymbol{o}'}$ for each one. We take a *global* approach, jointly transforming all objects. This takes into account that interactions are contextual: turning on a `Faucet` might fill up a `Cup` if and only if there is one beneath it.

Letting the observed objects in the interaction be $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$, with encodings $\mathbf{h}_{\boldsymbol{o_1}}$ and $\mathbf{h}_{\boldsymbol{o_2}}$ respectively, we model the transformation via the following multilayer perceptron:

$$[\hat{\mathbf{h}}_{\boldsymbol{o_1}'}, \hat{\mathbf{h}}_{\boldsymbol{o_2}'}] = \text{MLP}_{\text{apply}}\Big(\big[\mathbf{h}_{\boldsymbol{a}}, \mathbf{h}_{\boldsymbol{o_1}}, \mathbf{h}_{\boldsymbol{o_2}}\big]\Big). \tag{6.4}$$

The result can be decoded into symbolic form using the object decoder (Equation 6.3).

Loss function and training

We train our dynamics model on ($\vec{\boldsymbol{o}}$,$\boldsymbol{a}$,$\vec{\boldsymbol{o}}'$) transitions. The model primarily learns by running $\vec{\boldsymbol{o}}$,$\boldsymbol{a}$ through the model, predicting the updated output state $\hat{\mathbf{h}}_{\boldsymbol{o}'}$, and minimizing the cross-entropy of generating attributes of the real changed object $\vec{\boldsymbol{o}}'$. We also regularize the model by encoding objects $\vec{\boldsymbol{o}}, \vec{\boldsymbol{o}}'$ and having the model learn to reconstruct them. We weight all these cross-entropy losses equally. Our architecture uses 3-layer Transformers, totalling 17M parameters.

6.3.2 Language Grounding

After pretraining our physical dynamics model, we integrate it with a Transformer Language Model (LM). In our framework, the role of the LM will be to both encode natural language sentences of

actions into a hidden state approximating $\mathbf{h}_{\boldsymbol{a}}$, as well as summarizing the result of an interaction ($\vec{\boldsymbol{o}}$,$\boldsymbol{a}$,$\vec{\boldsymbol{o}}'$) in natural language.

Choice of LM. Our framework is compatible with any language model. However, to explore the impact of pretraining data on grounding later in this paper, we pretrain our own with an identical architecture to the smallest GPT2 (Radford et al. [213]; 117M). To handle both classification and generation well, we mask only part of the attention weights out, allowing the model to encode a "prefix" bidirectionally; it generates subsequent tokens left-to-right [65]. We pretrain the model on Wikipedia and books.

We next discuss architectural details of performing the language transfer, along with optimization.

Transfer Architecture

English actions to vector form. Given a natural language description $\boldsymbol{s_a}$ of an action $\boldsymbol{a}$, like "The robot throws the vase," for PIGPEN-NLU, our model will learn to parse this sentence into a neural representation $\mathbf{h}_{\boldsymbol{a}}$, so the dynamics model can simulate the result. We do this by encoding $\boldsymbol{s_a}$ through our language model, $\mathbf{T}_{LM}$, with a learned linear transformation over the resulting (bidirectional) encoding. The resulting vector $\mathbf{h}_{\boldsymbol{s_a}}$ can then be used by Equation 6.4.

Summarizing the result of an action. For PIGPEN-NLG, our model simulates the result of an action $\boldsymbol{a}$ neurally, resulting in a predicted hidden state $\hat{\mathbf{h}}_{\boldsymbol{o}}$ for each object in the scene $\boldsymbol{o}$. To write an English summary describing "what changed," we first learn a lightweight fused representation of the transition, aggregating the initial and final states, along with the action, through a multilayer perceptron. For each object $\boldsymbol{o}_i$ we have:

$$\mathbf{h}_{\Delta \boldsymbol{o}_i} = \text{MLP}_{\Delta}([\mathbf{h}_{\boldsymbol{o}_i}, \hat{\mathbf{h}}_{\boldsymbol{o}_i'}, \mathbf{h}_{\boldsymbol{a}}]). \tag{6.5}$$

We then use the sequence $[\mathbf{h}_{\Delta \boldsymbol{o}_1}, \mathbf{h}_{\Delta \boldsymbol{o}_2}]$ as bidirectional context for our our LM to decode from. Additionally, since our test set includes novel objects not seen in training, we provide the names of the objects as additional context for the LM generator (e.g. 'Vase, Laptop'); this allows a LM

to copy those names over rather than hallucinate wrong ones. Importantly we only provide the surface-form names, **not** underlying information about these objects or their usage as with few-shot scenarios in the recent GPT-3 experiments [33] – necessitating that PIGLET learns what these names *mean* through interaction.

Loss functions and training.

Modeling text generation allows us to incorporate a new loss function, that of minimizing the log-likelihood of generating each $\boldsymbol{s}_{\vec{o}'}$ given previous words and the result of Equation 6.5:

$$p(s_{i+1}^{\text{post}}|\boldsymbol{s}_{\vec{\boldsymbol{o}}',1:i}) = \mathbf{T}_{\text{LM}}(\mathbf{h}_{\Delta \boldsymbol{o}_1}, \mathbf{h}_{\Delta \boldsymbol{o}_2}, \boldsymbol{s}_{\vec{\boldsymbol{o}}',1:i}). \tag{6.6}$$

We do the same for the object states $\boldsymbol{s}_{\vec{o}}$ pre-action, using $\mathbf{h}_{\boldsymbol{o}_i}$ as the corresponding hidden states.

For PIGPEN-NLU, where no generation is needed, optimizing Equation 6.5 is not strictly necessary. However, as we will show later, it helps provide additional signal to the model, improving overall accuracy by several percentage points.

6.4 Experiments

We test our model's ability to encode language into a grounded form (PIGPEN-NLU), and decode that grounded form into language (PIGPEN-NLG).

6.4.1 PIGPEN-*NLU Results.*

We first evaluate models by their performance on PIGPEN-NLU: given objects $\vec{\boldsymbol{o}}$, and a sentence $\boldsymbol{s}_{\boldsymbol{a}}$ describing an action, a model must predict the resulting state of objects $\vec{\boldsymbol{o}}'$. We primarily evaluate models by accuracy; scoring how many objects for which they got all attributes correct. We compare with the following strong baselines:

a. No Change: this baseline copies the initial state of all objects $\vec{\boldsymbol{o}}$ as the final state $\vec{\boldsymbol{o}}'$.

b. GPT3-175B [33], a very large language model for 'few-shot' learning using a prompt. For GPT3, and other text-to-text models, we encode and decode the symbolic object states in a JSON-style dictionary format.

	Model	Accuracy (%)				Attribute-level accuracy (Test-Overall,%)				
		Val	Test			size	distance	mass	Temperature	isBroken
			Overall	Seen	Unseen	8-way	8-way	8-way	3-way	boolean
	No Change	27.4	25.5	29.9	24.0	83.2	84.1	96.3	86.0	94.8
text-to-text	GPT3-175B [33]	23.8	22.4	22.4	21.4	73.7	77.0	89.5	84.2	94.7
text-to-text	T5-11B [215]	68.5	64.2	79.5	59.1	83.9	88.9	94.3	95.4	98.1
text-to-text	T5-3B	66.6	63.3	77.1	58.7	81.6	90.0	94.0	95.6	98.4
text-to-text	T5-Large	56.5	54.1	69.2	49.1	81.8	84.6	94.3	96.3	95.8
text-to-text	T5-Base	56.0	53.9	69.2	48.8	81.1	87.5	93.6	96.1	96.5
text-to-text	T5-Small	39.9	36.2	57.0	38.0	82.2	84.9	93.8	89.6	93.5
BERT style	Alberti et al.2019, Pretrained Dynamics	61.3	53.9	71.4	48.1	87.7	87.6	97.5	93.4	97.5
BERT style	Alberti et al.2019	9.7	6.8	16.2	3.7	53.4	43.6	84.0	88.1	95.1
BERT style	G&D2019, Pretrained Dynamics	43.8	35.3	60.9	26.9	83.0	86.9	94.0	93.7	97.4
BERT style	G&D2019	15.1	11.3	23.1	7.3	68.6	47.3	82.2	88.3	95.8
	PIGLET	**81.8**	**81.1**	**83.8**	**80.2**	**92.3**	**91.9**	**99.2**	**99.8**	**99.0**

Table 6.1: **Overall results**. **Left**: we show the model accuracies at predicting all attributes of an object correctly. We compare PIGLET with 'text-to-text' approaches that represent the object states as a string, along with BERT-style approaches with additional machinery to encode inputs or decode outputs. PIGLET outperforms a T5 model 100x its size (11B params) and shows gains over the BERT-style models that also model action dynamics through a language transformer. **Right**: we show several attribute-level accuracies, along with the number of categories per attribute; PIGLET outperforms baselines by over 4 points for some attributes such as size and distance.

c. T5 [215]. With this model, we use the same 'text-to-text' format, however here we train it on the paired data from PIGPEN. We consider varying sizes of T5, from T5-Small – the closest in size to PIGLET, up until T5-11B, roughly 100x the size.

d. [11]-style. This paper originally proposed a model for VCR [305], where grounded visual information is fed into a BERT model as tokens; the transformer performs the grounded reasoning. We adapt it for our task by using our base LM and feeding in object representations from our pretrained object encoder, also as tokens. Our object decoder predicts the object, given the LM's pooled hidden state. This is "pretrained dynamics," we also consider a version without a randomly initialized dynamics model.

e. [99]-style. Thiso paper proposes using Transformers to model physical state, for tasks like entity tracking in recipes. Here, the authors propose decoding a physical state attribute (like `isCooked`) by feeding the model a label-specific `[CLS]` token, and then mapping the result through a hidden layer. We do this and use a similar object encoder as our [11]-style baseline.

Results. From the results (Table 6.1), we can draw several patterns. Our model, PIGLET performs best at getting all attributes correct; doing so over 80% on both validation and test sets, even for novel objects not seen during training. The next closest model is T5-11B, which scores 68% on validation. Though when evaluated on objects 'seen' during training it gets 77%, that number drops by over 18% for unseen objects. On the other hand, PIGLET has a modest gap of 3%. This suggests that our approach is particularly effective at connecting unpaired language and world representations. At the other extreme, GPT3 does poorly in its 'few-shot' setting, suggesting that size is no replacement for grounded supervision.

PIGLET also outperforms 'BERT style' approaches that control for the same language model architecture, but perform the physical reasoning inside the language transformer rather than as a separate model. Performance drops when the physical decoder must be learned from few paired examples (as in Gupta and Durrett [99]); it drops even further when neither model is given access to our pretrained dynamics model, with both baselines then underperforming 'No Change.' This suggests that our approach of having a physical reasoning model *outside of* an LM is a good

Model	Accuracy (val;%)
PIGLET, No Pretraining	10.4
PIGLET, Non-global $\text{MLP}_{\text{apply}}$	72.0
PIGLET, Global $\text{MLP}_{\text{apply}}$	78.5
PIGLET, Global $\text{MLP}_{\text{apply}}$, Gen. loss (6.6)	81.8
PIGLET, Symbols Only (Upper Bound)	89.3

Table 6.2: Ablation study on PIGPEN-NLU's validation set. Our model improves 6% by modeling global dynamics of all objects in the scene, versus applying actions to single objects in isolation. We improve another 3% by adding an auxiliary generation loss.

inductive bias.

Ablation study

In Table 6.2 we present an ablation study of PIGLET's components. Of note, by using a global representation of objects in the world (Equation 6.4), we get over 6% improvement over a local representation where objects are manipulated independently. We get another 3% boost by adding a generation loss, suggesting that learning to generate summaries helps the model better connect the world to language. Last, we benchmark how much headroom there is on PIGPEN-NLU by evaluating model performance on a 'symbols only' version of the task, where the symbolic action $\boldsymbol{a}$ is given explicitly to our dynamics model. This upper bound is roughly 7% higher than PIGLET, suggesting space for future work.

6.4.2 PIGPEN-*NLG Results*

Next, we turn to PIGPEN-NLG: given objects $\vec{\boldsymbol{o}}$, and the literal next action $\boldsymbol{a}$, a model must generate a sentence $\boldsymbol{s}_{\vec{o}'}$ describing what will change in the scene. We compare with the following

Model	BLEU		BERTScore		Human (test; [-1, 1])	
	Val	Test	Val	Test	Fluency	Faithfulness
T5	46.6	43.4	82.2	81.0	0.82	0.15
LM Baseline	44.6	39.7	81.6	78.8	0.91	-0.13
PIGLET	**49.0**	**43.9**	**83.6**	**81.3**	**0.92**	**0.22**
Human	44.5	45.6	82.6	83.3	0.94	0.71

Table 6.3: Text generation results on PIGPEN-NLG, showing models of roughly equivalent size (up to 117M parameters). Our PIGLET outperforms the LM baseline (using the same architecture but omitting the physical reasoning component) by 4 BLEU points, 2 BERTScore F_1 points, and 0.35 points in a human evaluation of language faithfulness to the actual scene.

baselines:

a. T5. We use a T5 model that is given a JSON-style dictionary representation of both $\vec{o}$ and $\boldsymbol{a}$, it is finetuned to generate summaries $\boldsymbol{s}_{\vec{o}'}$.

b. LM Baseline. We feed our LM hidden states $\mathbf{h}_{\boldsymbol{o}}$ from our pretrained encoder, along with its representation of $\boldsymbol{a}$. The key difference between it and PIGLET is that we do **not** allow it to simulate neurally what might happen next – $\text{MLP}_{\text{apply}}$ is never used here.

Size matters. Arguably the most important factor controlling the fluency of a language generator is its size [137]. Since our LM could also be scaled up to arbitrary size, we control for size in our experiments and only consider models the size of GPT2-base (117M) or smaller; we thus compare against T5-small as T5-Base has 220M parameters.

Evaluation metrics. We evaluate models over the validation and test sets. We consider three main evaluation metrics: BLEU [199] with two references, the recently proposed BERTScore [317], and conduct a human evaluation. Humans rate both the fluency of post-action text, as well as its faithfulness to true action result, on a scale from -1 to 1.

Results. We show our results in Table 6.3. Of note, PIGLET is competitive with T5 and sig-

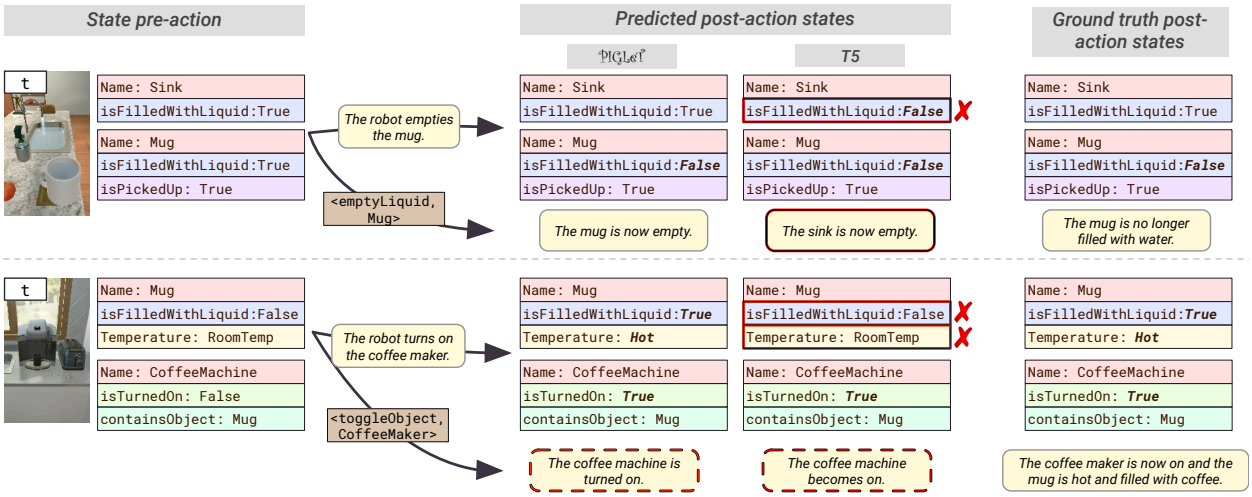


Figure 6.4: Qualitative examples. Our model PIGLET reliably predicts what might happen next (like the Mug becoming empty in Row 1), in a structured and explicit way. However, it often struggles at generating sentences for unseen objects like Mug that are excluded from the training set. T5 struggles to predict these changes, for example, it seems to suggest that emptying the Mug causes all containers in the scene to become empty.

nificantly outperforms the pure LM baseline, which uses a pretrained encoder for object states, yet has the physical simulation piece $\text{MLP}_{\text{apply}}$ removed. This suggests that simulating world dynamics not only allows the model to predict what might happen next, it leads to more faithful generation as well.

6.5 Analysis

6.5.1 Qualitative examples.

We show two qualitative examples in Figure 6.4, covering both PIGPEN-NLU as well as PIGPEN-NLG. In the first row, the robot empties a held `Mug` that is filled with water. PIGLET gets the state, and generates a faithful sentence summarizing that the mug becomes empty. T5 struggles somewhat, emptying the water from both the `Mug` and the (irrelevant) `Sink`. It also generates

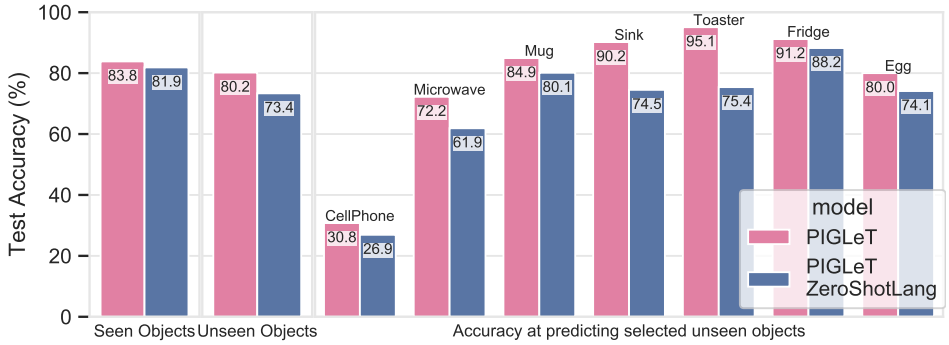


Figure 6.5: PIGPEN-NLU performance of a **zero-shot** PIGLET, that was pretrained on Books and Wikipedia without reading any words of our 'unseen' objects like 'mug.' It outperforms a much bigger T5-11B overall, though is in turn beaten by PIGLET on unseen objects like 'Sink' and 'Microwave.'

text saying that the Sink becomes empty, instead of the Mug.

In the second row, PIGLET correctly predicts the next object states, but its generated text is incomplete – it should also write that the mug becomes filled wtih Coffee. T5 makes the same mistake in generation, and it also underpredicts the state changes, omitting all changes to the `Mug`.

We suspect that T5 struggles here in part because `Mug` is an unseen object. T5 only experiences it through language-only pretraining, but this might not be enough for a fully grounded representation.

6.5.2 Representing novel words

The language models that perform best today are trained on massive datasets of text. However, this has unintended consequences [23] and it is unlike how children learn language, with children learning novel words from experience [35]. The large scale of our pretraining datasets might allow models to learn to perform physical-commonsense like tasks for wrong reasons, overfitting to surface patterns rather than learning meaningful grounding.

We investigate the extent of this by training a 'zero-shot' version of our backbone LM on

Wikipedia and books – the only difference is that we explicitly **exclude** all mentioned sentences containing one of our "unseen" object categories. In this setting, not only must PIGLET learn to ground words like 'mug,' it must do so without having seen the word 'mug' during pretraining. This is significant because we count over 20k instances of 'Mug' words (including morphology) in our dataset.

We show results in Figure 6.5. A version of PIGLET with the zero-shot LM does surprisingly well – achieving 80% accuracy at predicting the state changes for "Mug" – despite never having been pretrained on one before. This even outperforms T5 at the overall task. Nevertheless, PIGLET outperforms it by roughly 7% at unseen objects, with notable gains of over 10% on highly dynamic objects like `Toaster`s and `Sink`s.

6.6 Related Work

Grounded commonsense reasoning. In this work, we study language grounding and commonsense reasoning at the representation and concept level. The aim is to train models that learn to acquire concepts more like humans, rather than performing well on a downstream task that (for humans) requires commonsense reasoning. Thus, this work is somewhat different versus other 3D embodied tasks like QA [91, 56], along with past work for measuring such grounded commonsense reasoning, like SWAG, HellaSWAG, and VCR [303, 306, 305]. The knowledge covered is different, as it is self-contained within THOR. While VCR, for instance, includes lots of visual situations about what people are doing, this thesis section focuses on learning the physical properties of objects.

Zero-shot generalization. There has been a lot of past work involved with learning 'zero-shot': often learning about the grounded world in language, and transferring that knowledge to vision. Techniques for this include looking at word embeddings [77] and dictionary definitions [302]. In this work, we propose the inverse. This approach was used to learn better word embeddings [101] or semantic tuples [288], but we consider learning a component to be plugged into a deep Transformer language model.

Past work evaluating these types of zero-shot generalization have also looked into how well

models can compose concepts in language together [156, 227]. Our work considers elements of compositionality through grounded transfer. For example, in PIGPEN-NLG, models must generate sentences about the equivalent of dropping a 'dax', despite never having seen one before. However, our work is also contextual, in that the outcome of 'dropping a dax' might depend on external attributes (like how high we're dropping it from).

Structured Models for Attributes and Objects. The idea of modeling actions as functions that transform objects has been explored in the computer vision space [274]. Past work has also built formal structured models for connecting vision and language [186, 150], we take a neural approach and connect today's best models of language *form* to similarly neural models of a simulated environment.

Past work has also looked into training neural models for a target domain – similar to our factorized model for physical interaction. For example, [167] and [79] learn pretrained models for an instruction-following task in a blocks world, also using an autoencoder formulation. Our goal in this work is somewhat different: we are interested in learning physical reasoning about everyday objects, that might be discussed loosely in language (but with recurring issues of reporting bias [93]). We thus build a model that can be tied in with a pretrained language model, while also exhibiting generalization to new objects (that were not mentioned in language). We compare our model to today's largest language models that learn *from text alone*, and find better performance despite having 100x fewer parameters.

6.7 Conclusion

In this section of the thesis, we presented an approach PIGLET for jointly modeling language form and meaning. We presented a testbed PIGPEN for evaluating our model, which performs well at grounding language to the (simulated) world.

Chapter 7

LANGUAGE-AND-VISION NEURAL SCRIPT KNOWLEDGE MODELS, (MERLOT)

This chapter contains material that was originally published in [311].

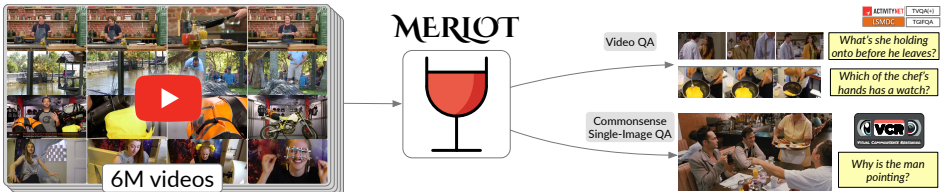


Figure 7.1: Multimodal Event Representation Learning Over Time. We learn representations of multimodal script knowledge from 6 million YouTube videos. These representations can then be applied to a variety of downstream tasks that require commonsense or temporal visual reasoning.

7.1 Introduction

The human capacity for commonsense reasoning is shaped by how we experience causes and effects over time. Consider the still image of people dining at a restaurant in the bottom right of Figure 7.1: while a literal, concrete description like “people sitting at a table eating" might be technically correct for the static scene, it doesn’t capture the richer temporal, commonsense inferences that are nonetheless obvious: *before* sitting down, the people had to meet up, agree where to go, and enter the restaurant; *at present*, the man is pointing because the server just came to the table, and she might want to know whose food is whose; and *after*, it is likely the server will return to the kitchen to help another table.

Teaching machines this type of *script knowledge* [229] is a significant challenge in no small part because enumerating all facts, inferences, and counterfactuals is prohibitive. As a result, the

highest performing models on vision-and-language tasks, including Visual Commonsense Reasoning (VCR) (where Figure 7.1's scene originates from), learn about the visual world exclusively through static images paired with literal captions [251, 43, 169, 179, 293, 80]. Though some captions might hint at the past and future, it is not obvious that even training on, e.g., 400M literal image/text pairs [214] will result in models capable of temporal reasoning.

In this thesis section, we introduce MERLOT, short for Multimodal Event Representation Learning Over Time. MERLOT is a model that learns commonsense representations of multimodal events by self-supervised pretraining over 6M unlabelled YouTube videos. With the goal of learning multimodal reasoning capacity beyond static images/literal captions, we train MERLOT to **a)** match individual video frames with contextualized representations of the associated transcripts, and to **b)**, contextualize those frame-level representations over time by "unmasking" distant word-level corruptions [61] and reordering scrambled video frames.

We validate our model on a diverse suite of video tasks, requiring both recognition- and cognition-level reasoning across long and short timescales; when finetuned, MERLOT achieves a new state-of-the-art on 12 such tasks. Additionally, we show that our script-knowledge representations transfer to the single image domain. On Visual Commonsense Reasoning (VCR; [305]), our model achieves particularly strong performance, outperforming models that require heavy visual supervision (in the form of object detection bounding boxes, or images paired with pristine captions).

Beyond finetuning, we show both quantitatively and qualitatively that MERLOT has a strong out-of-the-box understanding of everyday events and situations. Given a scrambled visual story, [126, 4], MERLOT can sort image sequences to match captions which tell a globally coherent narrative. Despite considerable domain shift from videos to static images, MERLOT outperforms strong baselines like CLIP [214] and UNITER [43], which independently match images to text and thus cannot reason over long-term contexts as effectively. This capacity for temporal coherence emerges during pretraining: analysis of MERLOT's attention patterns (Figure **??**) show that regions attend to captions that are distant in time (and vice versa), allowing it perform cross-modal coreference to piece together a holistic view of situations.

Finally, ablations of MERLOT show that 1) pretraining works better when we train on videos rather than still images, aided crucially by our strategy of corrupting highly visual words in the masked language modeling task, 2) using a diverse set of videos covering many aspects of everyday situations improves downstream performance compared to curated instructional video corpora [250, 188] which both cover a smaller slice of the visual world (confirming hypotheses from past work [118]); and 3) MERLOT's performance does not saturate even after many epochs of training on the pretraining corpus we curated, YT-Temporal-180M, as it continues to improve performance simply with more pretraining. The combination of these results suggests that learning full-stack visual reasoning and multimodal world knowledge from video data is a promising path forward for future research.

In summary, our main contributions in this section of the thesis are:

1. MERLOT a performant end-to-end vision and language model, that learns powerful multimodal world representations from videos and their transcripts – using no labeled data.
2. YT-Temporal-180M, a diverse corpus of frames/ASR derived from a filtered set of 6M diverse YouTube videos, which we show greatly aids performance, and
3. A set of experiments/ablations demonstrating the strong performance of MERLOT on a set of 14 tasks, spanning finetuning and zero-shot transfer, and images and videos.

At `rowanzellers.com/merlot`, we have released code, data, and models for public research use. By the way, if you've made it this far in reading my thesis, let me know and I'll buy or make you a coffee sometime! Otherwise no worries, I can't say I did that much besides putting all the papers in a giant LaTeX document! On with our regular scheduled programming...

7.2 Related Work

7.2.1 Joint representations of written text and images

There is a long history of work on learning joint text-image representations [24]. Recently, several papers have proposed "Visual BERT" models [251, 43, 11, 169, 179, 293, 80], trained on image captioning datasets such as MSCOCO [171]. In general, features are extracted using Anderson

et al. [14]'s frozen object detector, which was originally trained on Visual Genome [149]. Some exceptions are Zhang et al. [315], who use an even larger object detector trained on more labeled data; Kim et al. [141], who use an ImageNet-pretrained backbone [58], and Shen et al. [237], who study a CLIP backbone [214] pretrained on web image-caption pairs.

Overall, these approaches all learn visual representations of static images, and rely on significant human annotation in doing so (e.g. through literal image descriptions). Instead, our approach learns *dynamic* visual representations purely from videos – their frames, and a transcript of what is said – thus using no human annotation.

7.2.2 Learning from videos, with automatic speech recognition (ASR) transcripts

Prior works have used web videos with ASR to build weakly-supervised object detectors [211], action detectors/classifiers [297, 9, 152, 192], instruction aligners [182, 8, 39], video captioners [233, 117, 197, 239], and visual reference resolvers [125]. Of late, works have sought to learn multimodal representations transferable to many tasks from uncurated sets of (usually how-to) videos [188, 249, 250, 189, 323, 12, 10, 6]; generally these are applied to video understanding tasks like activity recognition. One challenge is designing an appropriate objective for learning video-level representations. Lei et al. [165]'s ClipBERT model learns vision-language representations from image captions, which more literally describe image content versus the longer ASR transcripts we consider. Tang et al. [252] use a pretrained dense image captioner [148] to provide auxiliary labels for web how-to videos. Both approaches use (supervised) ResNets pretrained on ImageNet [108] as their visual backbones. MERLOT is trained using a combination of objectives requiring no manual supervision; it nonetheless outperforms both prior approaches on downstream tasks.

7.2.3 Temporal ordering and forecasting

There has been a large body of work on analyzing 'what happens next' in videos [144]. Some modeling choices include using pixels [73, 266], graphs [20], euclidean distance using sensors

[5], or studying cycle consistency across time [70]. In addition to extrapolation, past work has studied deshuffling objectives in videos [190, 277], though this has mostly been limited to the visual modality. In contrast to these papers, our goal in this part of the thesis is learning *multimodal* script knowledge representations: using both language and vision as complementary views into the world, instead of just tracking what changes on-screen.

7.3 MERLOT*: Multimodal Event Representation Learning Over Time*

We now present our unified model for learning script knowledge through web videos; including our pretraining dataset, architecture, and objectives.

7.3.1 YT-Temporal-180M

We collect YT-Temporal-180M, a dataset for learning multimodal script knowledge, derived from 6 million public YouTube videos. Our YT-Temporal-180M intentionally spans many domains, datasets, and topics. We began with 27 million candidate video IDs (which we then filtered), including instructional videos from HowTo100M [188], lifestyle vlogs of everyday events from the VLOG dataset [75], and YouTube's auto-suggested videos for popular topics like 'science' or 'home improvement.' Our intent (in making the corpus as diverse as possible) was to encourage the model to learn about a broad range of objects, actions, and scenes [118]: we will later show through an ablation that limiting our pretraining to only instructional videos indeed hurts performance downstream.

We filtered videos using the YouTube API, which provides access to videos themselves, their ASR track (automatically transcribed speech tokens), and other metadata. We discard videos 1) without an English ASR track; 2) that are over 20 minutes long; 3) that belong to visually "un-grounded" categories like video game commentaries; and 4) that have thumbnails unlikely to contain objects, according to a lightweight image classifier. We add punctuation to the ASR by applying a sequence-to-sequence model trained to add punctuation to sentences/paragraphs from news articles.

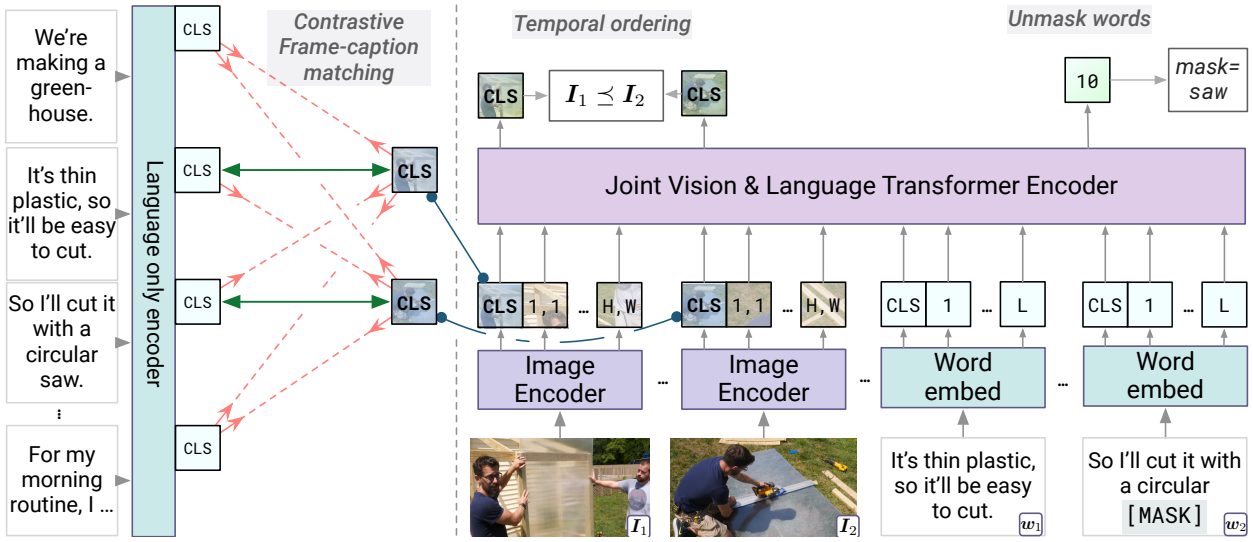


Figure 7.2: **Left**: MERLOT learns to match contextualized captions with their corresponding video frames. **Right**: the same image encoding is provided, along with (masked) word embeddings, into a joint vision-language Transformer model; it then unmasks ground words (like 'saw' in this example) and puts scrambled video frames into the correct order.

Each video $\mathcal{V}$ might contain thousands of frames. In this work, we represent a video $\mathcal{V}$ as a sequence of consecutive **video segments** $\{\boldsymbol{s}_t\}$. Each segment $\boldsymbol{s}_t$ consists of:

a. an image frame $\boldsymbol{I}_t$, extracted from the middle timestep of the segment,

b. the words $\boldsymbol{w}_t$ spoken during the segment, with a total length of L tokens.

To split the videos into segments, we byte-pair-encode (BPE; [234, 213]) each video transcript and align tokens with YouTube's word-level timestamps. This enables us to split the videos into segments of L=32 BPE tokens each; our final dataset has 180 million segments of this form.

7.3.2 MERLOT *Architecture*

A diagram of MERLOT is given in Figure 7.2. MERLOT takes a sequence of video frames $\{\boldsymbol{s}_t\}$ as input. We encode each frame $\boldsymbol{I}_t$ using an image encoder, embed the words $\boldsymbol{w}_t$ using a learned embedding, and jointly encode both using a Transformer [261]. After pretraining, the architecture

can be applied to a variety of vision-and-language tasks with minimal modification. For video QA, for example, we pass several video frames to the image encoder, the question to the text encoder, and extract a single vector representation from the `CLS` token position. For each task, we learn a lightweight classification head mapping from this hidden state to the task's label space.

Image encoder. We train our image encoder end-to-end, alongside the rest of the model, from random initialization (thus without learning from supervised data). While most performant vision-and-language models pre-extract features from a (supervised) object detector [251, 169, 179, 43, 168], for the sake of pre-training efficiency we use a grid-based hybrid ResNet/Vision Transformer.[1]

Specifically: our encoder uses a ResNet-50 backbone, followed by a 12-layer, 768-dimensional Vision Transformer [108, 261, 67]. We made additional modifications that improve efficiency, including: 1) we trained on smaller, widescreen images of size 192x352 (because most YouTube videos are widescreen) using a patch size of 16x16 pixels; 2) we mirror [67]'s alterations of removing the C5 block in ResNet-50; and 3) we save compute further by average-pooling the final-layer region cells using a kernel size of 2×2. With these modifications, our image encoder requires 40 gigaFLOPs for a forward pass, which is 2% of the 2 teraFLOPs required for the Faster-RCNN.

In summary: given an image of size $W \times H$, the image encoder will output a $W/32 \times H/32$ feature map, along with two `CLS` hidden states: one for pooling a global representation of the image, and another for pretraining (Task **1**.).

Joint Vision-Language Encoder. The joint encoder is a 12-layer, 768-dimensional Transformer [261], mirroring the `RoBERTa` base architecture [175]; we initialize it with pretrained `RoBERTa` weights. To compute joint representations, we first embed the tokens $\{\boldsymbol{w}_t\}$ via lookup, and then add position embeddings to both language and vision components (i.e., $\{\boldsymbol{I}_t\}$). The position embeddings differ between different segments, so as to distinguish between images and captions at different timesteps. Finally, we pass the independent visual and textual feature maps to

[1] Standard object detectors have expensive operations for proposing regions, and extracting features from those regions (RoI-pooling); our grid approach avoids these. Recent work has proposed using 'grid features' broadly [133], yet on tasks like VCR these approaches have so far underperformed the more expensive object detector backbones [305]; our results suggest that 'grid features' can perform well broadly.

our joint encoder.

The tokens $\boldsymbol{w}_t$ in each segment begin with a `CLS` token; recall that the feature maps for each frame $\boldsymbol{I}_t$ start with one as well. At those positions, we will later pool final-layer hidden-state representations, for use in pretraining along with downstream tasks.

7.3.3 Pretraining Tasks and Objectives

We use the following three objectives to pretrain MERLOT, that cover 'full-stack' visual reasoning – from recognition subtasks (like object detection) that operate at the frame level, to more 'cognitive' tasks that operate at the video level.

1. **Contrastive frame-transcript matching** [318, 214]. We want to ensure that the underlying image encoder produces helpful image representations. Thus, we use the video transcript to compute a 'language-only' representation of each video segment; and use a contrastive loss to maximize its similarity to corresponding representations from the image encoder.[2]

   Unlike what is the case for many image captions, the words $\boldsymbol{w}_t$ in each segment are often not sufficient to describe the gist of $\boldsymbol{I}_t$, or even what the key objects might be – for that, video-level contextualization is often required. We thus pass the entire transcript into the language-only encoder, which then extracts hidden states for each segment at the segment-level `CLS` tokens.

   Given matching representations for each frame $\boldsymbol{I}_t$ and caption $\boldsymbol{w}_t$ as positive examples, the negative examples come from all other frame-caption pairs in the batch – whether or not they come from the same video. We project both of these representations into a size-768 hidden state which is then unit-L2-normalized, and compute an all-pairs dot-product between all image and text representations. We divide these logits by a temperature of $\tau = 0.05$, and then apply a pairwise cross entropy loss to encourage matching captions and frames.

2. **(Attention) Masked Language Modeling** When providing words into the joint vision-and-language encoder, we randomly replace 20% with a `MASK` token, a random word, or the same

[2]To save memory, our 'language-only encoder' for this subtask shares parameters with the joint vision-and-language encoder.

word; MERLOT must then reconstruct the correct word with a cross-entropy loss, following [61].

This approach is commonly used by 'visual BERT' models in the image captioning domain, where captions are concise, and thus the identity of masked concrete words is difficult for models to recover given language context alone. However, we observed qualitatively that videos break these assumptions: people tend to ramble, and often mention key objects multiple times. Thus, applying vanilla BERT-style masking often causes ungrounded fillers like 'umm' or 'yeah' to get masked, while the (repeated) names of important objects are often partially masked, penalizing the learning of multimodal representations.

We introduce a simple solution to this problem, that we call **attention masking**: we use attention weights from a language-only transformer (introduced in the previous objective) as a heuristic for which words are grounded. 50% of the time, we mask out a random token; the other 50% of the time, we mask out one of the top 20% most-attended-to-tokens. We then apply SpanBERT masking [135], randomly corrupting the following or preceding tokens with an average length of 0.5 tokens in each direction; this makes it harder for models to over-rely on BPE artifacts. We show in ablations that this improves performance.

3. **Temporal Reordering**. We have the model order the image frames in a video, forcing it to explicitly learn temporal reasoning and giving it an *interface* to measure such temporal reasoning. Here, 40% of the time, we randomly pick an integer i between 2 and N (the number of segments provided to the joint encoder). Then we randomly scramble i video frames chosen at random, by replacing the segment-level position embeddings (e.g. `[image_t]`) for that frame with a random and unique position embedding, e.g. `[image_unk_0]`). These random position embeddings are learned, and separate from the 'unshuffled' position embeddings. This allows the model to order each 'shuffled' frame conditioned on frames provided in the correct order (if any).

   To compute the reordering loss, we extract hidden states from each frame at the `CLS` token position. For each pair of frames, we concatenate their hidden states h_{t_i} and h_{t_j} and pass the

result through a two-layer MLP, predicting if $t_i < t_j$ or $t_i > t_j$. We optimize this using a cross-entropy loss.

7.3.4 Pretraining MERLOT

We pretrain our model for 40 epochs over our video dataset. We preprocess the dataset into examples with sequences of $N{=}16$ video segments each, each containing up to $L{=}32$ BPE tokens.[3] The language-only encoder computes contrastive representations given this entire sequence, its total length is thus 512 tokens. To save memory, we provide the joint vision-language encoder 4 groups of $N = 4$ segments each. At an image training resolution of 192×352, the joint model's sequence length is 396 tokens. To combine the losses, we multiply the contrastive loss by a coefficient of 0.25, which we found scaled its gradient magnitudes to roughly the same magnitude as the Mask LM loss.

We train the model using a v3-1024 TPU pod, at a batch size of 1024 sequences (or 16k segments) in total. This pretraining process on this hardware takes 30 hours.

7.4 Experiments: Transferring MERLOT *to Downstream Tasks*

In this section, we explore MERLOT on 14 different tasks, covering vision-language reasoning on static images as well as videos; we present analysis and ablations to dig deeper into our performance.

7.4.1 Image tasks

VCR. We consider VCR [305], a task and dataset where models must answer commonsense visual questions about images. These questions, about e.g. 'what might happen next' or 'what are people's intentions,' force MERLOT to transfer video-level understanding to the world of single images.

[3]To train the model on as much data as possible, we merged together the segments of short videos, and split up longer videos, such that all preprocessed examples in our dataset have exactly $N{=}16$ video segments.

	Q→A	QA→R	Q→AR
ViLBERT [179]	73.3	74.6	54.8
Unicoder-VL [168]	73.4	74.4	54.9
VLBERT [169]	73.8	74.4	55.2
UNITER [43]	75.0	77.2	58.2
VILLA [80]	76.4	79.1	60.6
ERNIE-ViL [293]	77.0	80.3	62.1
MERLOT (`base`-sized)	**80.6**	**80.4**	**65.1**

Table 7.1: Results on VCR [305]. We compare against SOTA models of the same '`base`' size as ours (12-layer vision-and-language Transformers). MERLOT performs best on all metrics.

VCR provides additional 'referring expression' information to models in the form of bounding boxes around named entities. For example, if `Person1` is referenced in the question, the location of `Person1` is also given in the image. We provide this information to models by drawing (in pixel space) a colored highlight around the referenced entity; this differs from prior works (that integrate these entities into detection architectures).

Our results on the three VCR settings, in comparison to other models at the same ('base') scale, are given in Table 7.1. Our model outperforms these other models, that all learn from exclusively static images (paired with captions and supervised object detections).

Unsupervised ordering of Visual Stories. To probe our model's ability to do out-of-the-box commonsense reasoning over events in images, we next consider the Visual Storytelling dataset [126, 178]. Each story in this dataset contains five images and captions in a certain order; the order tells a joint narrative between the captions and the images. Past work has considered unshuffling image-caption pairs [4], but we take a slightly different approach in this work to avoid language-only biases, which can rely on discursive clues to order text [61, 240]. In our formulation, models are given the captions in sorted order, and must match frames to the captions. Our formulation disarms language-only baselines, while still allowing us to quantify MERLOT's capacity for com-

	Spearman (↑)	Pairwise acc (↑)	Distance (↓)
CLIP [214]	.609	78.7	.638
UNITER [43]	.545	75.2	.745
MERLOT	**.733**	**84.5**	**.498**

Table 7.2: Results unscrambling SIND visual stories[126, 4]. Captions are provided in the correct order; models must arrange the images temporally. MERLOT performs best on all metrics by reasoning over the entire story, instead of independently matching images with captions.

monsense temporal reasoning.

We compare MERLOT with two strong out-of-the-box baselines for text-image matching: CLIP [214], which encodes each caption and image separately and computes similarity through a dot product, and UNITER [43] which jointly represents each image/caption pair, and is trained in part using a 'text-image matching' objective. We use our temporal reordering loss to find the most probable ordering of the video frames; for CLIP and UNITER we compute a maximum-weight bipartite matching [153] over the pairwise image-text similarity scores.

Results over 5K stories are given in Table 7.2. MERLOT's performance in comparison to the algorithms trained from image-literal caption pairs suggests that, with no fine-tuning, our model has strong capability to reason about past and future events expressed in collections of temporal visual stories.

7.4.2 Video Reasoning

We report results on 12 video reasoning tasks: TVQA [162], TVQA(+) [163], VLEP [164], MSRVTT-QA [284], MSRVTT-Multichoice [298], LSMDC-Multichoice, LSMDC fill-in-the-blank QA [255, 224], ActivityNetQA [299, 111], TGIFQA [131], and DramaQA [47]. We apply MERLOT to these tasks in the same way. We sample a sequence of 5 to 7 still frames from each video

Tasks	Split	Vid. Length	ActBERT [323]	ClipBERT$_{8x2}$ [165]	SOTA	MERLOT
MSRVTT-QA	Test	Short	-	37.4	41.5 [287]	**43.1**
MSR-VTT-MC	Test	Short	88.2	-	88.2 [323]	**90.9**
TGIF-Action	Test	Short	-	82.8	82.8 [165]	**94.0**
TGIF-Transition	Test	Short	-	87.8	87.8 [165]	**96.2**
TGIF-Frame QA	Test	Short	-	60.3	60.3 [165]	**69.5**
LSMDC-FiB QA	Test	Short	48.6	-	48.6 [323]	**52.9**
LSMDC-MC	Test	Short	-	-	73.5 [298]	**81.7**
ActivityNetQA	Test	Long	-	-	38.9 [287]	**41.4**
Drama-QA	Val	Long	-	-	81.0 [140]	**81.4**
TVQA	Test	Long	-	-	76.2 [140]	**78.7**
TVQA+	Test	Long	-	-	76.2 [140]	**80.9**
VLEP	Test	Long	-	-	67.5 [164]	**68.4**

Table 7.3: Comparison with state-of-the-art methods on video reasoning tasks. MERLOT outperforms state of the art methods in **12** downstream tasks that involve short and long videos.

clip, initialize new parameters only to map the model's pooled `CLS` hidden state into the output labels, and finetune MERLOT with a softmax cross entropy loss.

As shown in Table 7.3, for all these datasets MERLOT sets a new state-of-the-art. Given the diversity of tasks and the strengths of the comparison models, these results provide strong evidence that MERLOT learned strong multimodal and temporal representations.

7.4.3 Ablations

We present ablations over VCR and TVQA+ to study the effect of several modeling decisions.

Context size. Table 7.4a shows the effect of varying the number of segments N given to the joint vision-and-language encoder during pretraining. In the first two rows, we provide only a sin-

Training setup	VCR	TVQA+
One segment (N=1)	73.8	75.2
One segment, attention masking	73.5	74.5
Four segments	74.1	73.3
🍷 Four segments, attention masking	**75.2**	**75.8**

(a) **Context helps together with attention masking.** Pretraining on more segments at once improves performance, but more context can encourage language-only representation learning. Attention masking counteracts this, giving an additional 1 point boost.

Training setup	VCR	TVQA+
No contrastive V-L loss	57.5	67.6
No temporal ordering loss	**75.5**	75.6
🍷 All losses	75.2	**75.8**

(b) **Contrastive V+L loss is crucial.** Removing it makes performance drop significantly; the temporal ordering loss is not as important for downstream finetuning.

	VCR
No boxes	74.8
🍷 Drawn-on boxes	**79.4**

(c) **Drawing on bounding boxes helps**, suggesting that our model uses it to decode the 'referring expression' information (e.g. `person1`).

Dataset	VCR
Conceptual ∪ COCO	58.9
HowTo100M	66.3
🍷 YT-Temporal-180M	**75.2**
HowTo100M-sized YT-Temporal-180M	72.8
YTT180M, raw ASR	72.8

(d) **Diverse (video) data is important.** Applying our architecture to caption data leads to poor results. Our model performs better on HowTo100M, yet still below our (more diverse) YT-Temporal-180M, even when controlled for size. Using raw ASR (vs. denoised ASR) reduces performance.

# epochs	VCR
5 epochs	75.2
10 epochs	75.9
20 epochs	77.0
30 epochs	78.5
🍷 40 epochs	**79.4**

(e) **Training for longer helps,** with performance increasing monotonically over training iterations.

Table 7.4: Ablation study on the validation set of VCR question answering ($Q \rightarrow A$) and TVQA+, in accuraty (%). We put a 🍷 next to the configurations we chose for MERLOT.

gle video segment (N=1) to the model.[4] In this limited regime, we find that our 'attention masking' approach (preferential masking of tokens that were highly attended-to by the contrastive language-only encoder) does not outperform a strong baseline of masking spans randomly [135]. Yet, when we expand the sequence length to N=4 segments/128 tokens, our masking becomes more effective, improving by 1 point over the baseline. This supports our hypothesis (Section 7.3.3.**2**.) that text-only shortcuts become increasingly viable with length, and that our attention-masking approach counteracts them.

Losses. In Table 7.4b, we ablate the losses. We find that the contrastive frame-transcript matching loss is crucial to performance, suggesting that an explicit objective is critical for the (randomly initialized) image backbone to learn visual representations. The temporal ordering loss appears less critical for downstream tasks; it helps for TVQA but performance drops slightly for VCR. Thus, we find that it helps primarily as an *interface* by which we can query the model about temporal events (i.e. for the story ordering experiments); the model might be learning this information from other objectives.

Drawing bounding boxes. Table 7.4c shows the effects of providing grounding information to VCR models by drawing boxes. Performance drops 5% when they are removed, suggesting that they help.

Dataset source. In Table 7.4d, we investigate pretraining MERLOT on two datasets beyond YT-Temporal-180M. First, we train on 3 million static image-caption pairs from Conceptual Captions [235] combined with MSCOCO [171]; for fair comparison, we train for the same number of steps as 5 epochs on our dataset. The resulting model achieves 58.9% accuracy on VCR. We suspect this might be due to 1) a smaller context window (Table 7.4a), and 2) overfitting (5 epochs on YT-Temporal-180M corresponds to 300 epochs on the caption data). Because our vision pipeline is trained from scratch, the scale of the curated/supervised image pairing corpora is a concern.

We next investigate the impact of video selection, comparing YT-Temporal-180M with HowTo100M [188]. To control for number of videos, we train for an equivalent amount of steps: 5 epochs on

[4]We keep the effective batch size the same, so that we use $4\times$ the number of sequences at $\frac{1}{4}$th the length.

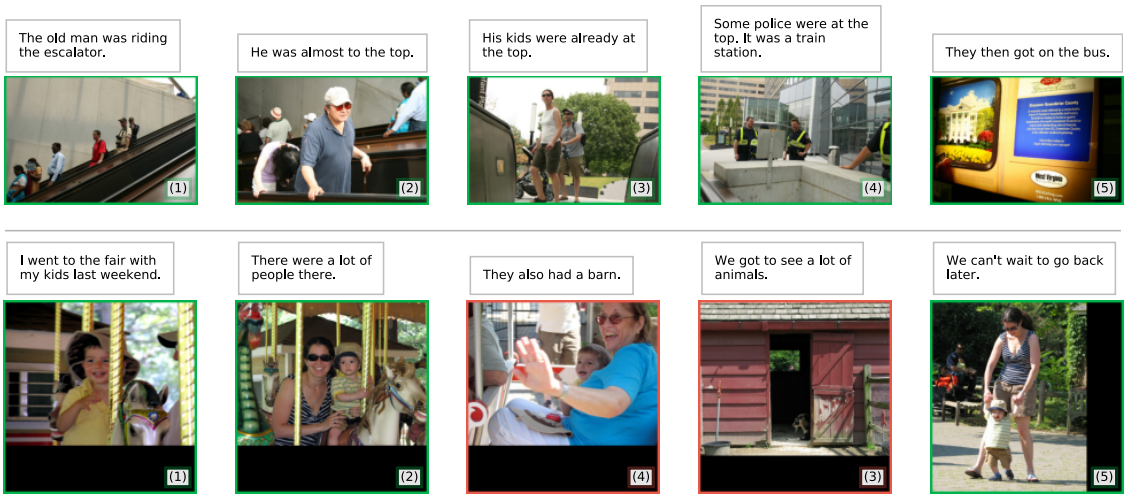


Figure 7.3: Zero-shot story ordering (same setup as Table 7.2). MERLOT performs temporal commonsense reasoning accross frames. In the first row, it uses 'the old man' mentioned to identify the 'kids' as parent-aged; in the second, it identifies riding a merry-go-round as an activity that takes a while.

our dataset, 30 epochs on HowTo100M, and likewise 30 epochs on a 'HowTo100M-sized YT-Temporal-180M'. Using diverse YT-Temporal-180M data vs. only instructional videos improves VCR performance by 6.5 points. This suggests that the how-to domain is limited in terms of visual phenomena covered, and that other domains (like web dramas and VLOGs) provide helpful signal for tasks like VCR [118]. Using all the data gives an additional 2.4-point performance boost.

Last, we investigate our choice to preprocess the YouTube ASR text with a language model (adding punctuation, etc); using 'raw ASR' instead of this preprocessing reduces performance by 2.4 points.

Pretraining longer. Last, in Table 7.4e, we investigate the effect of pretraining MERLOT for longer. The performance increases monotonically and doesn't begin to plateau, which suggests that had we pretrained MERLOT for *even* longer, its performance could improve even further.

7.4.4 Qualitative examples

In Figure 7.3, we show two qualitative examples of MERLOT's zero-shot story ordering capability. The examples here show that MERLOT has a strong understanding of events, transcending individual frames. In the first row, it orders the story correctly, performing vision-and-language coreference across several frames (e.g. frames and captions 2 and 3 use 'he' to refer to 'the old man' only mentioned in the first caption). Without resolving this coreference (establishing the subject as an elderly family member), it seems unlikely that anyone would describe the adults in frame (3) as 'kids.' Investigating the attention patterns of MERLOT backs up this claim; they show that MERLOT frequently addresses video tasks by merging attention across (distant) video segments.

MERLOT gets the second row 'wrong', but for an interesting reason. It reverses the order of frames (3) and (4), which groups the merry-go-round pictures together – even though caption (3) mentions a barn. This seems to capture the temporal commonsense intuition that people might ride a merry-go-round for a while, i.e., it is not an atomic event [53].

7.5 Conclusion, Limitations, and Broader Impacts

We introduced Multimodal Event Representation Learning Over Time (MERLOT). We trained the model through a combination of self-supervised objectives on 6M YouTube videos, in service of learning powerful multimodal representations that go beyond single frames. The model achieves strong performance on tasks requiring event-level reasoning over videos and static images. We hope that MERLOT can inspire future work for learning vision+language representations in a more human-like fashion compared to learning from literal captions and their corresponding images.

There are several potential limitations of MERLOT that would make for promising avenues of future work, including: 1) exploring finer-grained temporal reasoning pretraining objectives vs. frame ordering e.g., a temporal frame *localization* within transcripts; and 2) learning multilingually from non-English videos and communities on YouTube.

Like other pretraining work, MERLOT risks some potential negative impacts. We discuss

these in more detail below, in addition to the steps we took to reduce these harms.

7.5.1 Data collection and privacy.

As with other corpora gathered from the web used for pretraining data, YT-Temporal-180M contains publicly available content posted by users. We thus shaped our data gathering and release strategy to minimize inherent privacy and consent harms. Perhaps most importantly, we plan to only share video IDs for download, following a release strategy from prior work [1, 188] and giving users the right to opt out of not just YouTube, but our dataset as well.

7.5.2 Social biases.

The curation choices we made in this work could cause the model to exhibit undesirable social biases – *for this reason, along with others, we do not advocate for deployed use-cases.* For example, 30% of the data selected for by our filtering pipeline was local broadcast news (uploaded to YouTube). Including these news videos seems to perform better than filtering them out and only using how-to videos (Table 7.4b), however, there are risks when training on them. Local broadcast news (at least in the US) dedicates significant time to covering crime, sometimes in a racist and sensationalized manner [84, 63, 110]. Indeed, running a topic model over our data identifies several 'crime' categories. Past work has shown correlation between watching local news and having more explicit racialized beliefs about crime [62]; it seems likely therefore that training models on this data could teach them learn the same racist patterns.

Additionally, there are inherent social biases on YouTube – and treating these videos as equivalent to 'the world' [256] can embed hegemonic perspectives [105, 276, 23]. Most popular YouTubers are men [66] and video practices emerging on YouTube are often gendered [191]. YouTube also has problems with hate, including radical alt-right and 'alt-lite' content [219]. These problems – as with other problems in representation and power – are themselves amplified by the 'YouTube algorithm' [27] that recommends content to users. Though we downloaded videos independently of YouTube's recommender system, by filtering based on what content has views, we are implicitly filtering based on this algorithm. The dynamics of YouTube (i.e., which videos get

popular/monetized) influence the style and content of videos that get made and uploaded to the platform; this in turn shapes and is shaped by culture more broadly [247].

7.5.3 Dual use.

The video QA tasks that we studied carry risk of dual use, through possible downstream applications like surveillance [220, 326]. It seems unlikely that purely technological fixes and defenses – which themselves can be problematic [97] – could resolve these dynamics. Studying how well video-level pretraining enables surveillance applications might be an important avenue for future work, if only to inform stakeholders and policymakers about these risks.

7.5.4 Energy consumption.

The pretraining that we used in this work was expensive upfront [248]. Our results suggest that scaling up the amount of data and compute that we used might yield additional performance gains – but at increased environmental cost. To pretrain more efficiently, we used a much more lightweight architecture (in terms of FLOPs) than is standard for today's vision and language models. We hope that our public release of the model (for research use) can further amortize this cost.

7.5.5 Synthesizing these risks.

With these issues in mind, we release MERLOT and YT-Temporal-180M for researchers. We view our work, and our research artifacts, to be part of a larger conversation on the limits of pretrained 'foundation models' [30]. These models have broad impact to real-world areas like healthcare, law, and education. At the same time, these models have significant risks, including the harms that we outlined. We believe that further academic research into this video-and-language pretraining paradigm is important – especially to probe its limits and possible harms. We hope that our paper, code, and data release can contribute to this direction.

Chapter 8

LANGUAGE-AND-VISION-AND-SOUND NEURAL SCRIPT KNOWLEDGE MODELS, (MERLOT RESERVE)

This chapter contains material that was originally published in [312].

8.1 Introduction

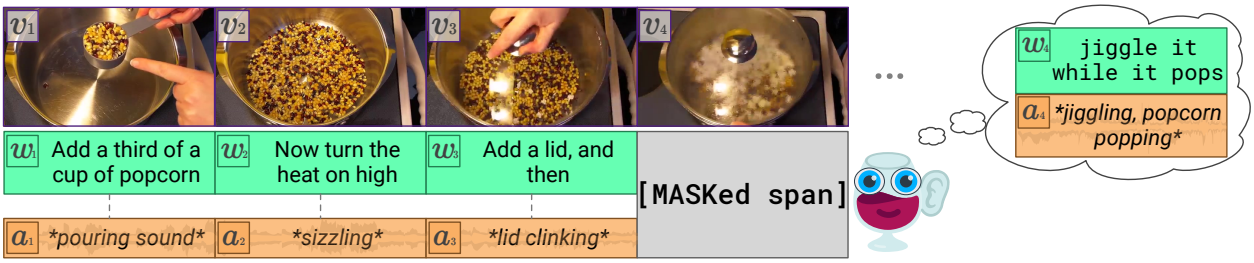


Figure 8.1: MERLOT RESERVE learns *multimodal neural script knowledge* representations of video – jointly reasoning over video frames, text, and audio. Our model is pretrained to predict which snippet of text (and audio) might be hidden by the MASK. This task enables it to perform well on a variety of vision-and-language tasks, in both zero-shot and finetuned settings.

The world around us is dynamic. We experience and learn from it using all of our senses, reasoning over them temporally through *multimodal script knowledge* [229, 311]. Consider Figure 8.1, which depicts someone cooking popcorn. From the images and dialogue alone, we might be able to imagine what *sounds* of the scene are: the process might begin with raw kernels scattering in an empty, metallic pot, and end with the dynamic 'pops' of popcorn expanding, along with the jiggling of a metal around the stove.

Predicting this sound is an instance of *learning from reentry*: where time-locked correlations

enable one modality to educate others. Reentry has been hypothesized by developmental psychologists to be crucial for how we as humans learn visual and world knowledge, much of it without need for an explicit teacher [206, 68, 40, 243]. Yet, we ask – can we build machines that likewise learn vision, language, and sound *together*? And can this paradigm enable learning *neural script knowledge*, that transfers to language-and-vision tasks, *even those without sound*?

In this work, we study these questions, and find that the answers are '**yes**.' We introduce a new model that learns self-supervised representations of videos, through all their modalities (audio, subtitles, vision). We dub our model MERLOT RESERVE[1], henceforth RESERVE for short. Our model differs from past work that learns from audio-image pairs [104, 158], from subtitled videos [250, 311], or from static images with literal descriptions [251, 43, 214]. Instead, we learn joint representations from *all modalities of a video*, using each modality to teach others. We do this at scale, training on over 20 million YouTube videos.

We introduce a new *contrastive masked span* learning objective to learn script knowledge across modalities. It generalizes and outperforms a variety of previously proposed approaches (e.g. [61, 251, 214, 311]), while enabling audio to be used as signal. The idea is outlined in Figure 8.1: the model must figure out which span of text (or audio) was MASKed out of a video sequence. We combine our objective with a second contrastive learning approach, tailored to learning *visual recognition* from scratch: the model must also match each video frame to a contextualized representation of the video's transcript [311]. Through ablations, we show that our framework enables rapid pretraining of a model and readily scales to 'large' transformer sizes (of 644M parameters).

Experimental results show that RESERVE learns powerful representations, useful even for tasks posed over only a few of the studied modalities. For example, when finetuned on Visual Commonsense Reasoning [305] (a vision+language task with no audio), it sets a new state-of-the-art, outperforming models trained on supervised image-caption pairs by **over 5%**. It does even better on video tasks: fine-tuning without audio, it outperforms prior work on TVQA [162] by a margin of **over 7%** (and given TVQA audio, performance increases even further). Finally, audio

[1]Short for Multimodal Event Representation Learning Over Time, with RE-entrant SupERVision of Events.

enables 91.1% accuracy on Kinetics-600 [36]. These performance improvements do not come at the expense of efficiency: our largest model uses one-fifths the FLOPs of a VisualBERT.

RESERVE also performs well in zero-shot settings. We evaluate on four diverse benchmarks: Situated Reasoning (STAR) [282], EPIC-Kitchens [55], LSMDC-FiB [224], and MSR-VTT QA [284]. These benchmarks require visual reasoning with respective emphasis on *temporality*, *future prediction*, and both *social* and *physical understanding*. With no fine-tuning or supervision, our model obtains competitive performance on each. Of note, it nearly doubles [287]'s SoTA zero-shot accuracy on MSR-VTT QA, and it outperforms supervised approaches (like ClipBERT [165]) on STAR.

Finally, we investigate *why*, and *on which training instances* audio-powered multimodal pretraining particularly helps. For instance, predicting audio rewards models for recognizing *dynamic state changes* (like cooked popcorn) and *human communication dynamics* (what are people's emotions and towards whom). Our model progressively learns these phenomena as pretraining progresses. These signals are often orthogonal to what snippets of text provide, which motivates learning from both modalities.

In summary, our key contributions are the following:

a. RESERVE, a model for multimodal script knowledge, fusing vision, audio, and text.

b. A new contrastive span matching objective, enabling our model to learn from text *and audio* self-supervision.

c. Experiments, ablations, and analysis, that demonstrate strong multimodal video representations.

Overall, the results suggest that learning representations from *all modalities* – in a time-locked, reentrant manner – is a promising direction, and one that has significant space for future work.

8.2 Related Work

Our work brings together two active lines of research.

Joint representations of multiple modalities. Many language-and-vision tasks benefit from

early fusion of the modalities [18]. A family of 'VisualBERT' models have been proposed for this: typically, these use a supervised object detector image encoder backbone, and pretrain on images paired with literal captions [251, 169, 179, 43, 293, 165]. Cross-modal interactions are learned in part through a *masked language modeling* (mask LM) objective [61], where subwords are replaced with '`MASK`', and models independently predict each subword conditioned on both images and unmasked tokens.[2]

Perhaps closest to our work is MERLOT [311], which learns a joint vision-text model from web videos with automatic speech recognition (ASR). Through a combination of objectives (including a variant of mask LM), MERLOT established strong results on a variety of video QA benchmarks when finetuned. However, it lacks audio: it is limited to representing (and learning from) video frames paired with subtitles. Our proposed RESERVE, which represents and learns from audio, outperforms MERLOT.

Co-supervision between modalities. A common pitfall when training a joint multimodal model is that complex *inter-modal* interactions can be ignored during learning, in favor of simpler *intra-modal* interactions [95, 49, 116]. For example, when using the aforementioned mask LM objective, models can *ignore visual input completely* in favor of text-text interactions [29]; this issue is magnified when training on videos with noisy ASR text [311].

A line of recent work thus learns independent modality-specific encoders, using objectives that cannot be shortcutted with simple intra-modal patterns. Models like CLIP learn image classification by matching images with their captions, contrastively [318, 214, 132]. Recent work has explored this paradigm for matching video frames with their transcripts [285], with their audio signal [225, 271], or both [10, 6]; these works likewise perform well on single-modality tasks like audio classification and activity recognition. These independent encoders can be combined through late fusion [225], yet late fusion is strictly less expressive than our proposed joint encoding (early fusion) approach.

---

[2]Recent papers propose extensions, like generating masked-out spans [45] or text [173, 275], but it is unclear whether they can outperform the VisualBerts on vision-language tasks like VCR [305]. Another extension involves learning from text-to-speech audio in a captioning setting [128, 174] – yet this lacks key supervision for environmental sounds and emotive speech.

Our work combines both lines of research. We learn a model for jointly representing videos, through all their modalities, and train it using a new learning objective that enables *co-supervision* between modalities.

8.3 Model: RESERVE

In this section, we present RESERVE, including: our model architecture (8.3.1), new pretraining objectives (8.3.2), and pretraining video dataset (8.3.3). At a high level, RESERVE represents a video by fusing its constituent modalities (vision, audio, and text from transcribed speech) together, and over time. These representations enable both finetuned and zero-shot downstream applications.

More formally, we split a video $\mathcal{V}$ into a sequence of non-overlapping segments in time $\{\boldsymbol{s}_t\}$. Each segment has:

a. A frame $\boldsymbol{v}_t$, from the middle of the segment,

b. The ASR tokens $\boldsymbol{w}_t$ spoken during the segment,

c. The audio $\boldsymbol{a}_t$ of the segment.

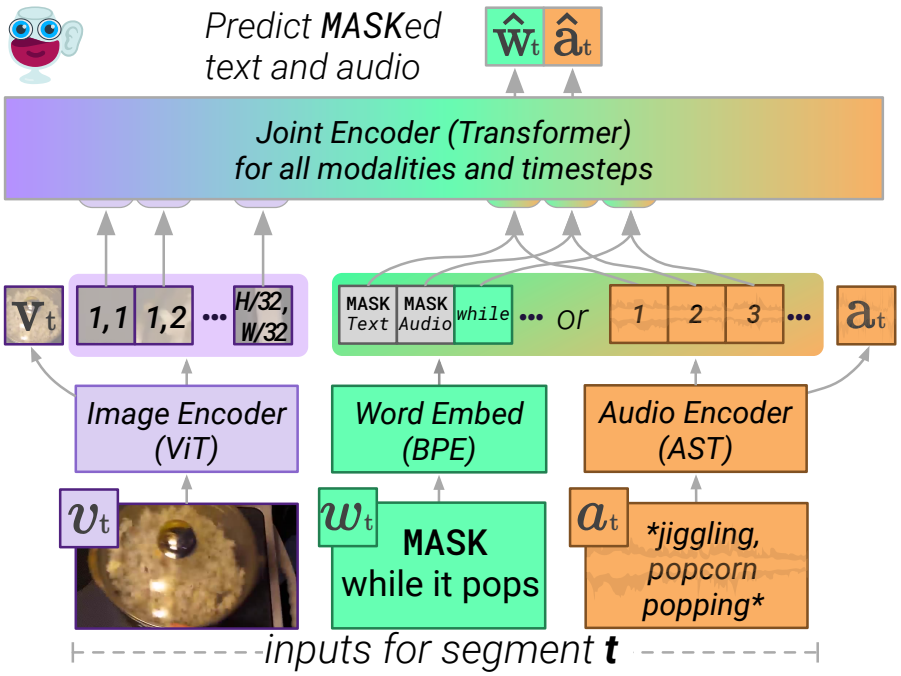
Segments default to 5 seconds in length.

As the text $\boldsymbol{w}_t$ was automatically transcribed by a model given audio $\boldsymbol{a}_t$, it is reasonable to assume that it contains strictly less information content.[3] Thus, for each segment $\boldsymbol{s}_t$, we provide models with exactly one of text *or* audio. We will further *mask out* portions of the text and audio during pretraining, to challenge models to recover what is missing.

8.3.1 Model architecture

An overview of RESERVE is shown in Figure 8.2. We first pre-encode each modality independently (using a Transformer [261] or images/audio; a BPE embedding table for text). We then learn a joint encoder to fuse all representations, together and over time.

[3]Despite being derived from the audio, pretraining with text is still paramount: 1) in §8.3.2 we discuss how jointly modeling audio+text prevents models from shortcutting pretraining objectives via surface correlations; 2) in §8.4.2 we show that incorporating both transcripts and audio during fine-tuning improves performance; and 3) a textual interface to the model is required for downstream vision+language with textual inputs.



centering

Figure 8.2: RESERVE architecture. We provide sequence-level representations of video frames, and *either* words or audio, to a joint encoder. The joint encoder contextualizes over modalities and timesteps, to predict what is behind MASK for audio $\widehat{\mathbf{a}}_t$ and text $\widehat{\mathbf{w}}_t$. We supervise these predictions with independently encoded targets: $\mathbf{a}_t$ from the audio encoder, and $\mathbf{w}_t$ from a separate text encoder (not shown).

Image encoder. We use a Vision Transformer (ViT; [67]) to encode each frame independently. We use a patch size of 16 and apply a 2x2 query-key-value attention pool after the Transformer, converting an image of size $H \times W$ into a $H/32 \times W/32$ feature map of dimension d_h.

Audio encoder. We split the audio in each segment $\boldsymbol{a}_t$ into three equal-sized *subsegments*, for compatibility with the lengths at which we mask text. We use an Audio Spectrogram Transformer to encode each subsegment independently [89]. The three feature maps are concatenated; the result is of size $18 \times d_h$ for every 5 seconds of audio.

Joint encoder. Finally, we jointly encode all modalities (over all input video segments) using a bidirectional Transformer. We use a linear projection of the final layer's hidden states for all

objectives (e.g. $\widehat{\mathbf{w}}_t$ and $\widehat{\mathbf{a}}_t$).

Independently-encoded targets. We will supervise the joint encoder by simultaneously learning independently-encoded 'target' representations for each modality. Doing this is straightforward for the image and audio encoders: we add a `CLS` to their respective inputs, and extract the final hidden state $\mathbf{v}_t$ or $\mathbf{a}_t$ at that position. For text, we learn a separate bidirectional Transformer *span encoder*, which computes targets $\mathbf{w}_t$ from a `CLS` and embedded tokens of a candidate text span. This enables zero-shot prediction (8.4.4).

Architecture sizes. We consider two model sizes in this work, which we pretrain from random initialization:

1. RESERVE-B, with a hidden size of 768, a 12-layer ViT-B/16 image encoder, and a 12-layer joint encoder.
2. RESERVE-L, with a hidden size of 1024, a 24-layer ViT-L/16 image encoder, and a 24-layer joint encoder.

We always use a 12-layer audio encoder, and a 4-layer text span encoder.

8.3.2 Contrastive Span Training

We introduce *contrastive span* training, which enables learning across and between the three modalities. As shown in Figure 8.3, the model is given a sequence of video segments. For each one, we include the video frame, and then three 'subsegments' that are each either text *or* audio. The subdivided audio segments are encoded independently by the Audio Encoder, before being fused by the Joint Encoder. We train by replacing 25% of these text and audio subsegments with a special `MASK` token. The model must match the representation atop the `MASK` *only with* an independent encoding of its span.

Our approach combines past success at matching images to their captions [214, 132] along with 'VisualBERT'-style prediction of independent tokens [251, 43] – though, crucially, we predict representations at a higher-level semantic unit than individual tokens. Our approach also enables the model to learn from both audio and text, while discouraging *memorization* of raw perceptual

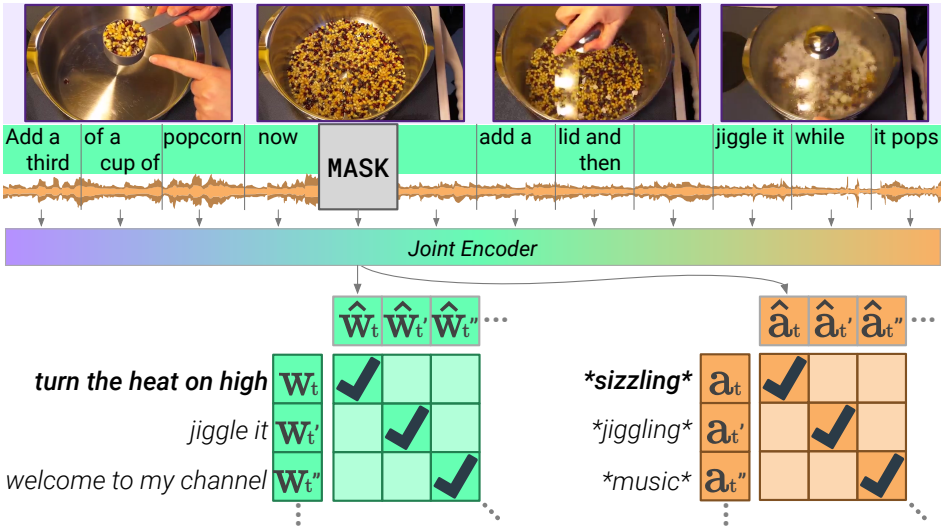


Figure 8.3: Contrastive span training. Given a video with all modalities temporally aligned, we MASK out a region of text and audio. The model must maximize its similarity *only to* an independent encoding of the text $\mathbf{w}_t$ and audio $\mathbf{a}_t$.

input, or tokens – which can harm representation quality [266].

Formally, we minimize the cross entropy between the MASKed prediction $\hat{\mathbf{w}}_t$ and its corresponding phrase representation $\mathbf{w}_t$, versus others in the batch $\mathcal{W}$:

$$\mathcal{L}_{\text{mask}\rightarrow\text{text}} = \frac{1}{|\mathcal{W}|}\sum_{\mathbf{w}_t\in\mathcal{W}}\left(\log\frac{\exp(\sigma\hat{\mathbf{w}}_t\cdot\mathbf{w}_t)}{\sum_{\mathbf{w}\in\mathcal{W}}\exp(\sigma\hat{\mathbf{w}}_t\cdot\mathbf{w})}\right). \tag{8.1}$$

We first L^2-normalize $\mathbf{w}$ and $\hat{\mathbf{w}}$, and scale their dot product with a parameter σ [214].[4] We then add this to its transposed version $\mathcal{L}_{\text{text}\rightarrow\text{mask}}$, giving us our text-based loss $\mathcal{L}_{\text{text}}$. Analogously, we define $\mathcal{L}_{\text{audio}}$ for audio, between the MASKed prediction $\hat{\mathbf{a}}_t$ and its target $\mathbf{a}_t$, versus others $\mathbf{a}$ in the batch.

In addition to these masked text and audio objectives, we simultaneously train the model to match video frames with a contextualized encoding of the transcript.[5] Here, the joint encoder

[4]Following past work, we optimize σ and clip it at 100, which enables the model to 'warm-up' its emphasis placed on hard negatives [214, 270].

[5]In MERLOT [311], this objective was found to be critical for learning visual recognition from self-supervised

encodes the entire video's transcript at once, extracting a single hidden representation per segment $\hat{\mathbf{v}}_t$. We use the same contrastive setup as Equation 8.1 to maximize the similarity of these vectors with the corresponding $\mathbf{v}_t$ vectors from the frames, giving us a symmetric frame-based loss $\mathcal{L}_{\text{frame}}$. The final loss is the sum of the component losses:

$$\mathcal{L} = \mathcal{L}_{\text{text}} + \mathcal{L}_{\text{audio}} + \mathcal{L}_{\text{frame}}. \tag{8.2}$$

Avoiding shortcut learning. Early on, we observed that training a model to predict a *perceptual* modality (like audio or vision) given input from *the same modality*, led to shortcut learning – a low training loss, but poor representations. We hypothesize that this setup encourages models to learn imperceptible features, like the exact model of the microphone, or the chromatic aberration of the camera lens [64]. We avoid this, while still using audio as a target, by simultaneously training on two kinds of masked videos:

i. Audio only as target. We provide only video frames and subtitles. The model produces representations of both *audio* and *text* that fill in MASKed blanks.

ii. Audio as input. We provide the model video frames, and subtitles *or audio* at each segment. Because the model is given audio as an input somewhere, the model only produces representations for MASKed *text*.

Another issue is that YouTube's captions are not perfectly time-aligned with the underlying audio. During our initial exploration, models took ready advantage of this shortcut: for instance, predicting an audio span based on what adjacent (overlapping) words sound like. We introduce a masking algorithm to resolve this.

Pretraining setup. We train on TPU v3-512 accelerators; training takes 5 days for RESERVE-B, and 16 days for RESERVE-L. We made pretraining more efficient through several algorithmic and implementation improvements. Of note, we simultaneously train on written (web) text, which enables more text candidates to be used. We use a batch size of 1024 videos, each with N=16 segments (split into two groups of 8 segments each). We use AdamW [142, 177] to minimize Equation 8.2.

---

videos.

8.3.3 Pretraining Dataset

Recent prior work on static images that demonstrates empirical improvements by increasing dataset size – all the way up to JFT-3B [145, 67, 214, 313]. The same pattern emerges in videos: prior work that has shown promising empirical improvements not only by scaling to 6 million videos/180M frames [311], but also by collecting a diverse set (i.e., going beyond instructional videos [117]).

To this end, we introduce a new training dataset of 20 million English-subtitled YouTube videos, and 1 billion frames, called YT-Temporal-1B. At the same time, we take steps to protect user privacy, directing scraping towards public, large, and monetized channels.

8.4 Experiments

In this section, we present model ablations (8.4.1), and show that a finetuned RESERVE obtains state-of-the-art results on VCR (8.4.1), TVQA (8.4.2), and Kinetics-600 (8.4.3). We then show that our model has strong zero-shot capability, over four challenging zero-shot tasks (8.4.2).

8.4.1 Visual Commonsense Reasoning (VCR)

We evaluate RESERVE first through finetuning on VCR [305]. Most competitive models for VCR are pretrained exclusively on images paired with captions, often with supervised visual representations (e.g. from an object detector). To the best of our knowledge, the only exception is MERLOT [311], which uses YouTube video frames and text as part of pretraining; no VCR model to date was pretrained on audio.

VCR Task. A model is given an image from a movie, and a question. The model must choose the correct answer given four multiple choice options ($Q \rightarrow A$); it then is given four *rationales* justifying the answer, and it must choose the correct one ($QA \rightarrow R$). The results are combined with a $Q \rightarrow AR$ metric, where a model must choose the right answer *and then* the right rationale, to get the question 'correct.'

Finetuning approach. We follow [311]'s approach: 'drawing on' VCR's detection tags onto the image, and jointly finetuning on $Q \rightarrow A$ and $QA \rightarrow R$. For both subproblems, we learn by

scoring each $Q \rightarrow A$ (or $QA \rightarrow R$) option independently. We pool a hidden representation from a MASK inserted after the text, and pass this through a newly-initialized linear layer to extract a logit, which we optimize through cross-entropy.

Ablations: contrastive learning with audio helps.

While we present our final, state-of-the-art VCR performance in 8.4.1, we first use the corpus for an ablation study. We use the same architecture and data throughout, allowing apples-to-apples comparison between modeling decisions. We start with a similar configuration to MERLOT [311] and show that contrastive span training improves further, particularly when we add audio.

Contrastive Span helps for *V*ision+*T*ext modeling. We start by comparing pretraining objectives for learning from YouTube ASR and video alone:

a. Mask LM. This objective trains a bidirectional model by having it *independently* predict masked-out tokens. We make this baseline as strong as possible by using SpanBERT-style masking [135], where text spans are masked out (identical to our *contrastive spans*). Each span $\boldsymbol{w}$ is replaced by a MASK token, and we predict each of its subwords w_i independently.[6]

b. VirTex [59]. In this objective, we likewise mask text subsegments and extract their hidden states. The difference is that we sequentially predict tokens $w_i \in \boldsymbol{w}$, using a left-to-right language model (LM) with the same architecture details as our proposed span encoder.

Results are in Table 8.1. Versus these approaches, our contrastive span objective boosts performance by over 2%, after one epoch of pretraining only on vision and text. We hypothesize that its faster learning is caused by encouraging models to learn concept-level span representations; this might not happen when predicting tokens individually [**?**].

***A*udio pretraining helps**, even for the audio-less VCR:

d. Audio as target. Here, the model is only given video frames and ASR text as input. In addition to performing contrastive-span pretraining over the missing text spans, it does the same over

[6]Like [135], we concatenate the MASK's hidden state with a position embedding for index i, pass the result through a two-layer MLP, and use tied embedding weights to predict w_i.

	Configuration *for one epoch of pretraining*	VCR val Q→A (%)
V+T	Mask LM [61, 251, 311]	67.2
V+T	VirTex-style [59]	67.8
V+T	🍷 Contrastive Span	**69.7**
V+T+A	🍷 Audio as target	70.4
V+T+A	🍷 Audio as input and target	**70.7**
V+T+A	Audio as input and target, w/o strict localization	70.6
	RESERVE-B	**71.9**

Table 8.1: **Ablation study** of our contrastive span objective. It outperforms prior work in a Vision+Text setting, with a 1% boost when audio is added. Our full setup, adding written text, improves another 1%. 🍷 denotes part of our full model.

the (held-out) audio span (Equation 8.2. This boosts VCR accuracy by 0.7%.

e. **Audio as input and target.** The model does the above (for video+text input sequences), and simultaneously is given video+text+audio sequences, wherein it must predict missing text. This boosts accuracy by 1% in total.

f. **Sans strict localization.** We evaluate the importance of our strict localization in time. Here, in addition to correct subsegments at the *true* position t as a correct match, we count adjacent MASKed out regions as well. An extreme version of this was proposed by [92], where a positive match can be of any two frames in a video. Yet even in our conservative implementation, performance drops slightly, suggesting localization helps.

Putting these all together, we find that contrastive span pretraining outperforms mask LM, with improved performance when audio is used **both as input and target**. For our flagship model, we report results in Table 8.1 on simultaneously training on web-text sequences as well, this improves

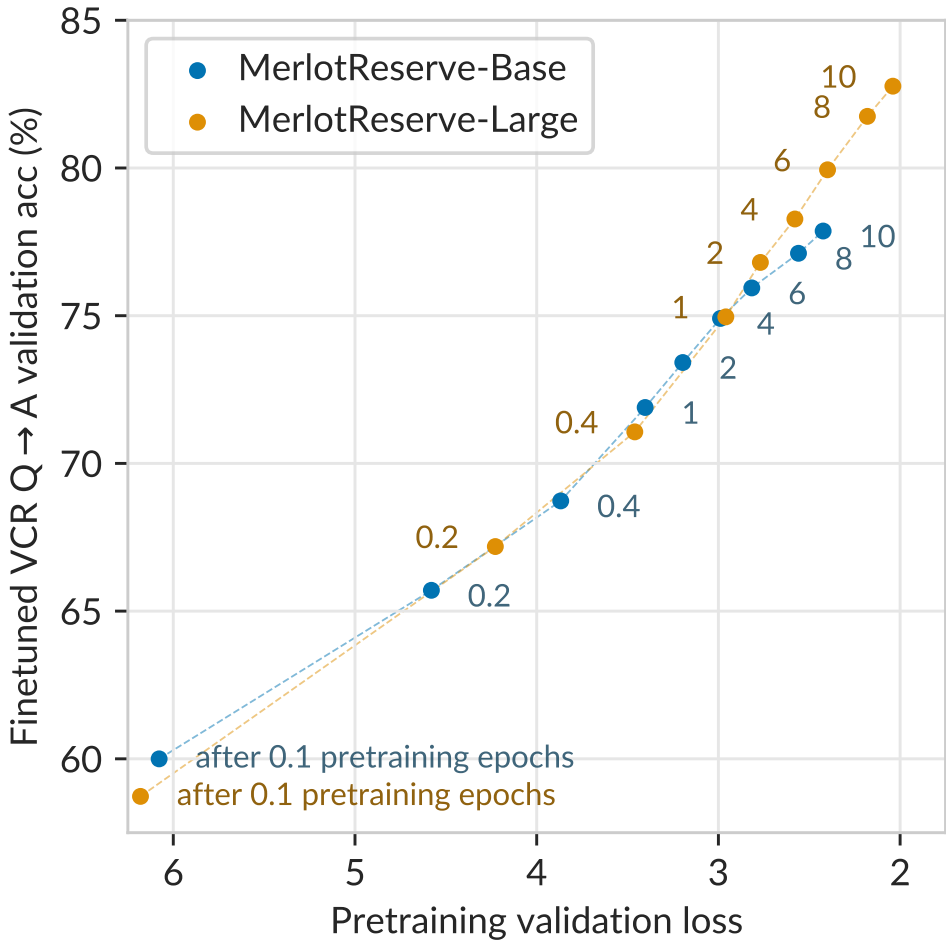


Figure 8.4: **Pretraining progress**: performance on contrastive-span pretraining, vs. finetuned VCR validation accuracy. Pretraining RESERVE-B for 9 more epochs boosts performance by 5%; L by 8%.

performance by an additional 1%.

VCR Results

Encouraged by these results, we train our models for 10 epochs on YT-Temporal-1B. Figure 8.4 demonstrates that finetuned VCR performance tracks with the number of pretraining epochs, as

	Model	VCR test (acc; %) Q→A	QA→R	Q→AR
Caption/ObjDet-based	ERNIE-ViL-Large [293]	79.2	83.5	66.3
	Villa-Large [80]	78.9	83.8	65.7
	UNITER-Large [43]	77.3	80.8	62.8
	Villa-Base [80]	76.4	79.1	60.6
	VilBERT [179]	73.3	74.6	54.8
	B2T2 [11]	72.6	75.7	55.0
	VisualBERT [169]	71.6	73.2	52.4
Video-based	MERLOT [311]	80.6	80.4	65.1
	RESERVE-B	79.3	78.7	62.6
	RESERVE-L	**84.0**	**84.9**	**72.0**

Table 8.2: RESERVE gets **state-of-the-art leaderboard performance on VCR**. We compare it with the largest submitted single models, including image-caption models that utilize heavy manual supervision (e.g. object detections and captions).

well as the validation loss.[7]

Finally, in Table 8.2, we compare RESERVE against the largest published models from the VCR leaderboard. Of note, RESERVE-L outperforms all prior work, by **over 5%** on Q→AR metric. It outperforms even large ensembles (e.g. 15 ERNIE-Large's) submitted by industry [293], though we do not show these on this table to focus on only single models.

Efficiency. The accuracy increase of RESERVE is not simply due to compute.[8] In fact, our RESERVE-L requires *one-fifth the FLOPs* of detector-based systems, like UNITER-Large [43].

[7]The plot suggests that if we pretrained longer, VCR performance might continue to increase, though a confounding factor might be the learning-rate schedule. With access to compute beyond our current capacity, future work would be well-suited to consider this and other pre-training modifications.

[8]Here, we use FLOPs as our key efficiency metric, as they are a critical bottleneck in model scaling [137, 67, 313]. On the other hand, we argue that parameter count can be misleading – for instance, many Transformer parameters can be tied together with minimal performance loss [159].

	Model	TVQA (acc; %) Val	Test
	Human [162]	–	89.4
Subtitles	MERLOT [311]	78.7	78.4
	MMFT-BERT [258]	73.5	72.8
	Kim et al [140]	76.2	76.1
	RESERVE-B	82.5	–
	RESERVE-L	**85.9**	**85.6**
Audio	RESERVE-B	81.3	–
	RESERVE-L	**85.6**	**84.8**
Both	RESERVE-B	83.1	82.7
	RESERVE-L	**86.5**	**86.1**

Table 8.3: RESERVE gets state-of-the-art results on TVQA by **over 7%**, versus prior work (that cannot make use of audio).

Moreover, because RESERVE-L uses a pure ViT backbone versus MERLOT's ViT-ResNet hybrid, it uses fewer FLOPs than MERLOT, while scoring 7% higher. Meanwhile, RESERVE-B outperforms 'base' detector-based models, while using *less than one-tenth their FLOPs*.

In terms of parameter count, RESERVE-B is comparable to prior work. On VCR, including the vision stack, RESERVE-B has 200M finetunable parameters and performs similarly to the 378M parameter UNITER-Large. RESERVE-L has 644M parameters.

8.4.2 Finetuning on TVQA

Next, we use TVQA [162] to evaluate our model's capacity to transfer to multimodal video understanding tasks. In TVQA, models are given a video, a question, and five answer choices. The scenes come from American TV shows, and depict characters interacting with each other through dialogue – which past work represents through subtitles.

Audio-Subtitle Finetuning. To evaluate how much audio can help for TVQA, we finetune

	Model	Kinetics-600 (%) Top-1	Top-5
Vision Only	VATT-Base[6]	80.5	95.5
	VATT-Large [6]	83.6	96.6
	TimeSFormer-L [25]	82.2	95.6
	Florence [300]	87.8	97.8
	MTV-Base [286]	83.6	96.1
	MTV-Large [286]	85.4	96.7
	MTV-Huge [286]	89.6	98.3
	RESERVE-B	88.1	95.8
	RESERVE-L	89.4	96.3
+Audio	RESERVE-B	89.7	96.6
	RESERVE-L	**91.1**	**97.1**

Table 8.4: RESERVE gets state-of-the-art results on Kinetics-600 by **1.5%** versus standard approaches (that cannot make use of audio).

RESERVE jointly between the 'Subtitles' and 'Audio' settings. Like on VCR, we consider one sequence per candidate: each contains video frame features, the question, the answer candidate, and a MASK token (from where we pool a hidden representation). During training, each sequence is duplicated: we provide one sequence with *subtitles* from the video, and for the other, we use *audio*. This lets us train a single model, and then test how it will do *given subtitles*, *given audio*, or *given both* (by averaging the two softmax predictions).

Results. We show TVQA results in Table 8.3. With subtitles and video frames alone, our RESERVE-B outperforms all prior work by over 3%. Combining subtitle-only and audio-only predictions performs even better, improving over 4% versus the prior state-of-the-art, MERLOT (and in turn over other models). The same pattern holds (with additional performance gains) as model size increases: RESERVE-L improves over prior work by **7.6%**.

	Model	Situated Reasoning (STAR) (test acc; %)					EPIC-Kitchens (val class-mean R5; %)			LSMDC (FiB test %)	MSR-VTT QA (test acc %)	
		Interaction	Sequence	Prediction	Feasibility	Overall	Verb	Noun	Action	Acc	top1	top5
	Supervised SoTA	ClipBERT [165]					AVT+ [85]			MERLOT [311]		
		39.8	43.6	32.3	31.4	36.7	28.2	32.0	15.9	52.9	43.1	
zero-shot	Random	25.0	25.0	25.0	25.0	25.0	6.2	2.3	0.1	0.1	0.1	0.5
zero-shot	CLIP (VIT-B/16) [214]	39.8	40.5	35.5	36.0	38.0	16.5	12.8	2.3	2.0	3.0	11.9
zero-shot	CLIP (RN50x16) [214]	39.9	41.7	36.5	**37.0**	38.7	13.4	14.5	2.1	2.3	2.3	9.7
zero-shot	Just Ask (ZS)[287]										2.9	8.8
zero-shot	RESERVE-B	44.4	40.1	38.1	35.0	39.4	17.9	15.6	2.7	26.1	3.7	10.8
zero-shot	RESERVE-L	42.6	41.1	37.4	32.2	38.3	15.6	19.3	4.5	26.7	4.4	11.5
zero-shot	RESERVE-B (+audio)	**44.8**	**42.4**	**38.8**	36.2	**40.5**	20.9	17.5	3.7	29.1	4.0	12.0
zero-shot	RESERVE-L (+audio)	43.9	42.6	37.6	33.6	39.4	**23.2**	**23.7**	**4.8**	**31.0**	**5.8**	**13.6**

Table 8.5: Zero shot results. On STAR, RESERVE obtains state-of-the-art results, outperforming finetuned video models. It performs well on EPIC-Kitchens (verb and noun forecasting), along with LSMDC, despite their long-tail distributions. On MSR-VTT QA, it outperforms past work on weakly-supervised video QA. Further, it outperforms CLIP (that cannot handle dynamic situations), and benefits from audio when given.

8.4.3 Finetuning on Kinetics-600 Activity Recognition

Next, we use Kinetics-600 [36] to compare our model's (finetuned) activity understanding versus prior work, including many top-scoring models that do not integrate audio. The task is to classify a 10-second video clip as one of 600 categories. We finetune RESERVE jointly over two settings: vision only, and vision+audio.

Results. We show Kinetics-600 results on the validation set, in Table 8.4. RESERVE improves by **1.7%** when it can jointly represent the video's frames with its sound. This enables it to outperform other large models, including VATT [6] which learns to represent audio independently from vision (and so cannot early-fuse them), along with the larger MTV-Huge model [286] by 1.5%.

8.4.4 Zero-Shot Experiments

Next, we show that our model exhibits strong zero-shot performance for a variety of downstream tasks. Our zero-shot interface is enabled by our *contrastive span objective*. For QA tasks that require predicting an option from a label space of short phrases, we encode this label space as vectors, and predict the closest phrase to a MASKed input. We consider:

i. Situated Reasoning (STAR) [282]. This task requires the model to reason over short situations in videos, covering four axes: interaction, sequence, prediction, and feasibility. The model is given a video, a templated question, and 4 answer choices. We convert templated questions into literal statements (which are more similar to YouTube dialogue); the label space is the set of four options.

ii. Action Anticipation in Epic Kitchens [55]. Here, the goal is to predict *future actions* given a video clip, which requires reasoning temporally over an actor's motivations and intentions. The dataset has a long tail of rare action combinations, making zero-shot inference challenging (since we do not assume access to this prior). As such, prior work [85, 78] trains on the provided in-domain training set. To adapt RESERVE to this task, we provide it a single MASK token as text input, and use as our label space of all combinations of verbs and nouns in the vocabulary (e.g. 'cook apple, cook avocado', etc.).

iii. LSMDC [181, 224]. Models are given a video clip, along with a video description (with a MASK to be filled in). We compare it with the vocabulary used in prior work [311].

iv. MSR-VTT QA [284]. This is an open-ended video QA task about what is literally happening in a web video. We use GPT3 [33], prompted with a dozen (unlabelled) questions, to reword the questions into statements with MASKs. This introduces some errors, but minimizes domain shift. We use a label space of the top 1k options.

For these tasks, we use $N{=}8$ video segments (dilating time when appropriate), and provide audio input when possible. We compare against both finetuned and zeroshot models, including running CLIP [214] on all tasks. CLIP is a strong model for zero-shot classification, particularly when *encyclopedic knowledge about images* is helpful; our comparisons showcase where multimodal

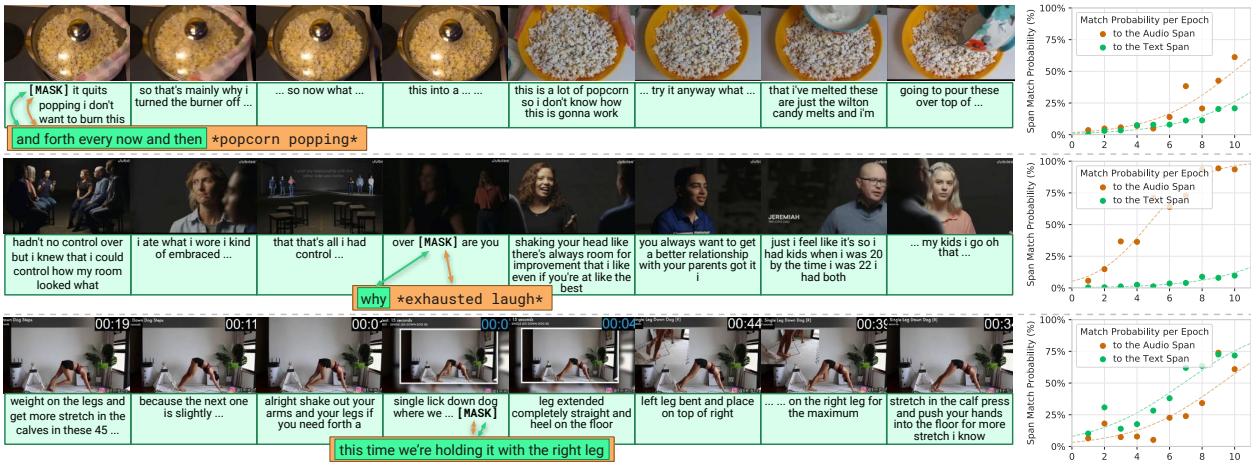


Figure 8.5: **Exploring** MASK**ed audio self-supervision**. Shown are example videos from our validation set, with predictions from RESERVE-B. During pretraining, our model progressively learns to pick up on audio-specific clues. It seems to recognize physical dynamics of *cooking popcorn*, matching the first row to its MASKed audio. Likewise, it seems to use social reasoning to match the second row to its audio. Both of these clues are orthogonal to what the subtitles provide.

script knowledge helps.

Results. Table 8.5 shows our model performs competitively:

i. On STAR, it obtains state-of-the-art results, with performance gain when audio is included. Interestingly, RESERVE-B outperforms its larger variant; we hypothesize that this is due to limited prompt searching around question templates. We qualitatively observed that RESERVE-L sometimes excludes topically correct options if they sound grammatically strange (to it).

ii. On EPIC-Kitchens, our model obtains strong results at correctly anticipating the verb and noun - despite the heavy-tailed nature of both distributions. It is worse on getting both right ('action'), we suspect that this might be due to priors (motifs) between noun and verb [304]. These are easy to learn given access to training data, but we exclude these as we consider the zero-shot task.

iii. On LSMDC, our model obtains strong results at filling-in-the-blank, likewise despite a heavy (unseen) frequency bias. Notably, it outperforms CLIP significantly, with CLIP often preferring

templates that use visually-relevant words, even if they don't make sense as a whole. For instance, given a clip of a mailman, CLIP chooses 'the mailman smiles off,' versus 'the mailman takes off.'

iv. Finally, our model performs well on MSR-VTT QA, outperforming past work that directly rewords subtitled instructional videos into video QA instances [287].

8.5 Qualitative Analysis: Why does audio help?

What can RESERVE learn from both text *and* audio? Three validation set examples are shown in Figure 8.5. The model is given the displayed text and video frames, and must match the MASK to the correct missing text and audio span (out of 48k total in the batch). The plots show RESERVE-B's probability of correctly identifying the correct audio or text span, as it progresses through 10 epochs of pretraining.

Audio's supervisory signal. In the first two rows of Figure 8.5, audio provides orthogonal supervision to text:

1. In the first row, the MASKed audio contains the sound of popcorn pops slowing. By the final epoch, RESERVE-B selects this specific auditory cue with 60% probability, over others (including from adjacent segments, at different stages of popping). Here, sound provides signal for joint vision-text understanding of the situation, as evidenced by its greater match probability.

2. The second row contains only the text 'why,' with the audio providing greatly more information — a female-presenting speaker (shown in the next frame) laughs, astonished that the child (in the frame afterwards) might want a better relationship with their parents.

3. In the third row, matching performance is similar between modalities, possibly as the yogi is narrating over a (muted) video recording, and not adding much information.

Role of text. Text is still a crucial complement to audio, in terms of the supervision it provides. Consider the second row: RESERVE-B learns to match the audio almost perfectly (perhaps reasoning that the speaker is shown in the next frame, and is laughing). In later epochs, its text-match probability increases: knowing that a 'why' question is likely to be asked is a valid *social* inference

to make about this (tense) situation.

Learning through multimodal reentry. Developmental psychologists have hypothesized that human children learn by *reentry*: learning connections between all senses as they interact with the world [68, 243]. Using a held-out modality (like audio) might support learning a better world representation (from e.g. vision and text), by forcing models to abstract away from raw perceptual input. Our work suggests that reentry has potential for machines as well.

8.6 Conclusion, Limitations, Broader Impact

We introduced MERLOT RESERVE, which learns jointly through sound, language, and vision, guided through a new pretraining objective. Our model performs well in both finetuned and zero-shot settings, yet it has limitations. Our model only learns from 40-second long videos; relies on ASR models for subtitles, and can only match (not generate) text and audio.

Still, we foresee broad possible societal impact of this line of work. Video-pretrained models might someday assist low vision or d/Deaf users [166, 90]. Yet, the same technology can have impacts that we authors consider to be negative, including surveillance, or applications that hegemonize social biases. Key dimensions include respecting user privacy during dataset collection, exploring biases in YouTube data, dual use, and energy consumption. In the published version of this paper [312] discuss our plan to *release our model and data* for research use so others can critically study this approach to learning script knowledge.

Chapter 9

CONCLUSION AND FUTURE WORK

In this thesis, we presented approaches for measuring and learning grounding. At the same time, there's a lot more work to be done in this direction, and we briefly discuss some of that here.

The work in this thesis has evolved as the field of AI has evolved. The amount of compute and data that are available to researchers now (in 2022) is a lot greater than was available back in 2016, for instance. At the same time, there's been helpful methodology developed over the years that this work relies on. Some of the highlights include work on neural architectures like Transformers [261, 67, 89], optimizers [177], dataset construction and analysis [149, 113, 126, 148, 162, 75, 96, 224], representation learning approaches in text [205, 212, 61, 175, 215, 135, 33**?**] and multimodally [251, 169, 179, 43, 214], along with analysis as to how these systems behave [121, 137].

Still, at the time of writing, it feels like our field is limited by some of these core abstractions. The dominant paradigm for training models is to train them on all the data we can download from the web (which has societal and ethical implications), in a static matter. These models have gotten so large that typically academics don't even think to train them anymore. It's more of a job for industry, and so we just use whatever pretrained (foundation?) models that they've publicly shared, often on training data that we don't have access to (another issue!). Given a pretrained model, we can (sometimes) finetune it and then see how it does on other benchmarks.

Every step in the above pipeline relies on static datasets – manually annotated by human workers and/or scraped from the web. But this differs a lot from how humans learn: through interaction with other agents and the world, via a curriculum which we play an active role in shaping [243]. It means that we as humans aren't just good at reading language or watching TV – we can use language and vision to help us act in the world. As a field, at some point we might need to go beyond the current *dataset-centric paradigm* towards open-ended environments; this raises a bunch

of other issues though in terms of efficiency and reproducibility of research.

There are other issues, possibly related, in terms of what today's models learn. Massive models like GPT3 [33] learn by reading all the text on the web that they can download – orders of magnitude more words than any human could ever read throughout their lifetime. As a result, they seem to be able to regurgitate a breadth of factual knowledge (like trivia) – yet perhaps because they're ungrounded, they can struggle to anticipate the outcomes of everyday situations [307]. As humans, we seem to be able to do 'more with less' in terms of both data and compute. It's not clear whether simply scaling up representation learning models solves the underlying challenges – in fact, some might become more difficult. For example, as humans, we learn representations about everyday objects that transcends modalities (e.g. we can read about cups, look at cups, and interact with cups, and these views of the same class of object are connected [243]). Yet, for a giant neural model like GPT3, it might have millions of 'cup neurons' which encode all the long-tail knowledge of every cup that's ever been written about in human history. It's not clear whether that long-tail knowledge makes it easier to tease out compact multimodal representations. Going forward, an interesting research area might be to try to learn complementary knowledge across modalities, similar in vein to PIGLeT [310].

There are also a lot of open questions about how to best learn representations about these complementary modalities. We learn functional representations about the world, including things like the affordances of everyday objects and what people's body language suggests, through multimodal interaction. Our understanding of perceptual modalities like vision and sound combines aspects of discriminative and generative learning. After looking at an image for a brief moment, we probably won't be able to reconstruct it pixel-by-pixel; rather, we learn an abstracted-away representation about what the key objects and people are, what they're doing and how they're situated. There are a lot of unanswered questions about how models can learn such representations.

Finally, there are important questions about the societal impacts of these models. As our field has moved towards larger models and datasets, the concentration of power shifts towards larger companies. This raises questions for us as practitioners and scientists who are working to shape the future direction of the field. Is machine learning inevitably a centralizing technology, like e.g.

nuclear power [280], or can we push against this trend? Likewise, how should this change the research we do, and how we share knowledge?

BIBLIOGRAPHY

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.

[3] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4971–4980, 2018.

[4] Harsh Agrawal, Arjun Chandrasekaran, Dhruv Batra, Devi Parikh, and Mohit Bansal. Sort Story: Sorting Jumbled Images and Captions into Stories. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 925–931, 2016.

[5] Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: experiential learning of intuitive physics. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 5092–5100, 2016.

[6] Hassan Akbari, Linagzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. VATT: transformers for multimodal self-supervised learning from raw video, audio and text. *arXiv preprint arXiv:2104.11178*, 2021.

[7] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Pro-*

ceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 961–971, 2016.

[8] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016.

[9] Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Simon Lacoste-Julien. Joint discovery of object states and manipulation actions. In *ICCV*, 2017.

[10] Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. *arXiv preprint arXiv:2006.16228*, 2020.

[11] Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. Fusion of detected objects in text for visual question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2131–2140, 2019.

[12] Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex Bronstein. Noise estimation using density estimation for self-supervised multimodal learning. *arXiv preprint arXiv:2003.03186*, 2020.

[13] Ashton Anderson, Daniel P. Huttenlocher, Jon M. Kleinberg, and Jure Leskovec. Effects of user similarity in social media. In *WSDM '12*, 2012.

[14] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[16] Yoav Artzi, Nicholas FitzGerald, and Luke S Zettlemoyer. Semantic parsing with combinatory categorial grammars. *ACL (Tutorial Abstracts)*, 3, 2013.

[17] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998.

[18] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018.

[19] Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5301. URL `https://www.aclweb.org/anthology/W19-5301`.

[20] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray kavukcuoglu. Interaction networks for learning about objects, relations and physics. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4509–4517, 2016.

[21] Hedi Ben-younes, Remi Cadene, Matthieu Cord, and Nicolas Thome. MUTAN: Multimodal Tucker Fusion for Visual Question Answering. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. URL `http://arxiv.org/abs/1705.06676`.

[22] Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.463. URL `https://www.aclweb.org/anthology/2020.acl-main.463`.

[23] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, 2021.

[24] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35 (8):1798–1828, 2013.

[25] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021.

[26] Or Biran and Courtenay Cotton. Explanation and justification in machine learning: A survey. In *IJCAI-17 Workshop on Explainable AI (XAI)*, page 8, 2017.

[27] Sophie Bishop. Anxiety, panic and self-optimization: Inequalities and the youtube algorithm. *Convergence*, 24(1):69–84, 2018.

[28] Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, et al. Experience grounds language. *arXiv preprint arXiv:2004.10151*, 2020.

[29] Yonatan Bitton, Gabriel Stanovsky, Michael Elhadad, and Roy Schwartz. Data efficient masked language modeling for vision and language. *arXiv preprint arXiv:2109.02040*, 2021.

[30] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv e-prints*, pages arXiv–2108, 2021.

[31] Silvia Bonaccio and Reeshad S. Dalal. Advice taking and decision-making: An integrative literature review, and implications for the organizational sciences. *Organizational Behavior and Human Decision Processes*, 101(2):127 – 151, 2006. ISSN 0749-5978. doi: https://doi.org/10.1016/j.obhdp.2006.07.001. URL http://www.sciencedirect.com/science/article/pii/S0749597806000719.

[32] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642, 2015. URL http://aclweb.org/anthology/D/D15/D15-1075.pdf.

[33] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[34] Zheng Cai, Lifu Tu, and Kevin Gimpel. Pay attention to the ending: Strong neural baselines for the roc story cloze task. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 616–622, 2017.

[35] S. Carey and E. Bartlett. Acquiring a single new word. 1978.

[36] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.

[37] Nathanael Chambers and Dan Jurafsky. Unsupervised Learning of Narrative Schemas and Their Participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing*

of the AFNLP: Volume 2 - Volume 2, ACL '09, pages 602–610, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-46-6. URL `http://dl.acm.org/citation.cfm?id=1690219.1690231`.

[38] Arjun Chandrasekaran, Viraj Prabhu, Deshraj Yadav, Prithvijit Chattopadhyay, and Devi Parikh. Do explanations make vqa models more predictable to a human? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1036–1042, 2018.

[39] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *CVPR*, 2019.

[40] Robin S Chapman. Children's language learning: An interactionist perspective. *The Journal of Child Psychology and Psychiatry and Allied Disciplines*, 41(1):33–54, 2000.

[41] Ping Chen, Fei Wu, Tong Wang, and Wei Ding. A semantic qa-based approach for text summarization evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[42] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1657–1668, 2017.

[43] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020.

[44] Gennaro Chierchia and Sally McConnell-Ginet. *Meaning and Grammar (2Nd Ed.): An Introduction to Semantics*. MIT Press, Cambridge, MA, USA, 2000. ISBN 0-262-53164-X.

[45] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. *arXiv preprint arXiv:2102.02779*, 2021.

[46] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[47] Seongho Choi, Kyoung-Woon On, Yu-Jung Heo, Ahjeong Seo, Youwon Jang, Seungchan Lee, Minsu Lee, and Byoung-Tak Zhang. DramaQA: character-centered video story understanding with hierarchical qa. *arXiv preprint arXiv:2005.03356*, 2020.

[48] Ching-Yao Chuang, Jiaman Li, Antonio Torralba, and Sanja Fidler. Learning to act properly: Predicting and explaining affordances from images. In *CVPR*, 2018.

[49] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *EMNLP*, pages 4069–4082, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1418. URL `https://aclanthology.org/D19-1418`.

[50] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, 2017.

[51] Robin Cooper, Dick Crouch, JV Eijckl, Chris Fox, JV Genabith, J Japars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, et al. A framework for computational semantics (fracas). Technical report, Technical report, The FraCaS Consortium, 1996.

[52] Council of Europe. *Common European Framework of Reference for Languages: learning, teaching, assessment*. Cambridge University Press, 2001.

[53] William Croft. *Verbs: Aspect and causal structure*. OUP Oxford, 2012.

[54] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer, 2006. doi: 10.1007/11736790_9.

[55] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 2021. URL `https://doi.org/10.1007/s11263-021-01531-2`.

[56] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2018.

[57] Ernest Davis and Gary Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM*, 58:92–103, 2015.

[58] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[59] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11162–11173, 2021.

[60] Jacob Devlin, Saurabh Gupta, Ross B. Girshick, Margaret Mitchell, and C. Lawrence Zitnick. Exploring nearest neighbor approaches for image captioning. *CoRR*, abs/1505.04467, 2015.

[61] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[62] Travis L Dixon. Crime news and racialized beliefs: Understanding the relationship between local news viewing and perceptions of african americans and crime. *Journal of Communication*, 58(1):106–125, 2008.

[63] Travis L Dixon and Daniel Linz. Overrepresentation and underrepresentation of african americans and latinos as lawbreakers on television news. *Journal of communication*, 50(2): 131–154, 2000.

[64] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[65] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*, 2019.

[66] Nicola Döring and M Rohangis Mohseni. Male dominance and sexism on youtube: results of three content analyses. *Feminist Media Studies*, 19(4):512–524, 2019.

[67] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[68] Gerald M Edelman. Neural darwinism: selection and reentrant signaling in higher brain function. *Neuron*, 10(2):115–125, 1993.

[69] Kiana Ehsani, Hessam Bagherinezhad, Joseph Redmon, Roozbeh Mottaghi, and Ali Farhadi. Who let the dogs out? modeling dog behavior from visual data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[70] Dave Epstein, Jiajun Wu, Cordelia Schmid, and Chen Sun. Learning temporal dynamics from cycles in narrated video. *arXiv preprint arXiv:2101.02337*, 2021.

[71] Matan Eyal, Tal Baumel, and Michael Elhadad. Question answering as an automatic evaluation metric for news article summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3938–3948, 2019.

[72] Panna Felsen, Pulkit Agrawal, and Jitendra Malik. What will happen next? forecasting player moves in sports videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3342–3351, 2017.

[73] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 64–72, 2016.

[74] Maxwell Forbes and Yejin Choi. Verb physics: Relative physical knowledge of actions and objects. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 266–276, 2017.

[75] David F Fouhey, Wei-cheng Kuo, Alexei A Efros, and Jitendra Malik. From lifestyle vlogs to everyday interactions. In *CVPR*, 2018.

[76] Michael C Frank and Noah D Goodman. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998, 2012.

[77] Andrea Frome, Greg Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. 2013.

[78] Antonino Furnari and Giovanni Maria Farinella. Rolling-unrolling lstms for action anticipation from first-person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2020.

[79] David Gaddy and Dan Klein. Pre-learning environment representations for data-efficient neural instruction following. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1946–1956, 2019.

[80] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. *arXiv preprint arXiv:2006.06195*, 2020.

[81] Matt Gardner, Jonathan Berant, Hannaneh Hajishirzi, Alon Talmor, and Sewon Min. On making reading comprehension more comprehensive. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 105–112, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5815. URL https://www.aclweb.org/anthology/D19-5815.

[82] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3356–3369, 2020.

[83] JJ Gibson. The ecological approach to visual perception. *Houghton Mifflin Comp*, 1979.

[84] Franklin D Gilliam Jr, Shanto Iyengar, Adam Simon, and Oliver Wright. Crime in black and white: The violent, scary world of local news. *Harvard International Journal of press/politics*, 1(3):6–23, 1996.

[85] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. *arXiv preprint arXiv:2106.02036*, 2021.

[86] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

[87] Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P18-2103`.

[88] Dave Golland, Percy Liang, and Dan Klein. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 410–419. Association for Computational Linguistics, 2010.

[89] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021.

[90] Steven M Goodman, Ping Liu, Dhruv Jain, Emma J McDonnell, Jon E Froehlich, and Leah Findlater. Toward user-driven sound recognizer personalization with people who are d/deaf or hard of hearing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–23, 2021.

[91] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[92] Daniel Gordon, Kiana Ehsani, Dieter Fox, and Ali Farhadi. Watching the world go by: Representation learning from unlabeled videos. *arXiv preprint arXiv:2003.07990*, 2020.

[93] Jonathan Gordon and Benjamin Van Durme. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 25–30. ACM, 2013.

[94] Venkata Subrahmanyan Govindarajan, Benjamin Chen, Rebecca Warholic, Katrin Erk, and Junyi Jessy Li. Help! need advice on identifying advice. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5295–5306, 2020.

[95] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. *arXiv preprint arXiv:1612.00837*, 2016. URL `https://arxiv.org/abs/1612.00837`.

[96] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[97] Ben Green. "good" isn't good enough. In *Proceedings of the AI for Social Good workshop at NeurIPS*, 2019.

[98] Ben Green and Salomé Viljoen. Algorithmic realism: Expanding the boundaries of algorithmic thought. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAT*)*, 2020.

[99] Aditya Gupta and Greg Durrett. Effective use of transformer networks for entity tracking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 759–769, 2019.

[100] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[101] Tanmay Gupta, Alexander Schwing, and Derek Hoiem. Vico: Word embeddings from visual

co-occurrences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7425–7434, 2019.

[102] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3608–3617, 2018.

[103] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/N18-2017`.

[104] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. *arXiv preprint arXiv:2106.13043*, 2021.

[105] Donna Haraway. Situated knowledges: The science question in feminism and the privilege of partial perspective. *Feminist studies*, 14(3):575–599, 1988.

[106] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42 (1-3):335–346, 1990.

[107] Tatsunori Hashimoto, Hugh Zhang, and Percy Liang. Unifying human and statistical evaluation for natural language generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1689–1701, 2019.

[108] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[109] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[110] Don Heider. *White news: Why local news programs don't cover people of color*. Routledge, 2014.

[111] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[112] Richard Held and Alan Hein. Movement-produced stimulation in the development of visually guided behavior. *Journal of comparative and physiological psychology*, 56(5):872, 1963.

[113] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *European Conference on Computer Vision*, pages 3–19. Springer, 2016.

[114] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[115] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. *European Conference on Computer Vision (ECCV)*, 2018.

[116] Jack Hessel and Lillian Lee. Does my multimodal model learn cross-modal interactions? it's harder to tell than you might think! In *EMNLP*, 2020.

[117] Jack Hessel, Bo Pang, Zhenhai Zhu, and Radu Soricut. A case study on combining ASR and visual features for generating instructional video captions. In *CoNLL*, November 2019.

[118] Jack Hessel, Zhenhai Zhu, Bo Pang, and Radu Soricut. Beyond instructional videos: Probing for more diverse visual-textual grounding on youtube. In *EMNLP*, 2020.

[119] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9 (8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[120] Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. Learning to write with cooperative discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1638–1649. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-1152.

[121] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *ICLR*. ICLR, 2020.

[122] Dirk Hovy and Shannon L Spruit. The social impact of natural language processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 591–598, 2016.

[123] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4418–4427. IEEE, 2017.

[124] Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. Explainable neural computation via stack neural module networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 53–69, 2018.

[125] De-An Huang, Joseph J. Lim, Li Fei-Fei, and Juan Carlos Niebles. Unsupervised visual-linguistic reference resolution in instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[126] Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1233–1239, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1147. URL https://www.aclweb.org/anthology/N16-1147.

[127] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[128] Gabriel Ilharco, Yuan Zhang, and Jason Baldridge. Large-scale representation learning from visually grounded untranscribed speech. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 55–65, 2019.

[129] Allan Jabri, Armand Joulin, and Laurens van der Maaten. Revisiting visual question answering baselines. In *European conference on computer vision*, pages 727–739. Springer, 2016.

[130] Aaron Jaech, Victoria Zayats, Hao Fang, Mari Ostendorf, and Hannaneh Hajishirzi. Talking to the crowd: What do people react to in online discussions? In *EMNLP*, 2015.

[131] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017). Honolulu, Hawaii*, pages 2680–8, 2017.

[132] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le,

Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *arXiv preprint arXiv:2102.05918*, 2021.

[133] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10267–10276, 2020.

[134] Roy Jonker and Anton Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

[135] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

[136] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 427–431, 2017.

[137] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[138] Jin-Hwa Kim, Kyoung Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard Product for Low-rank Bilinear Pooling. In *The 5th International Conference on Learning Representations*, 2017.

[139] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In *15th European Conference on Computer Vision*, pages 577–593. Springer, 2018.

[140] Seonhoon Kim, Seohyeong Jeong, Eunbyul Kim, Inho Kang, and Nojun Kwak. Self-supervised pre-training and contrastive representation learning for multiple-choice video

qa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13171–13179, 2021.

[141] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. *arXiv preprint arXiv:2102.03334*, 2021.

[142] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*. ICLR, 2015.

[143] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

[144] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *European Conference on Computer Vision*, pages 201–214. Springer, 2012.

[145] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 491–507. Springer, 2020.

[146] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[147] Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge 'naturally' in multi-agent dialog. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2962–2967, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1321. URL `https://www.aclweb.org/anthology/D17-1321`.

[148] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-Captioning Events in Videos. In *International Conference on Computer Vision (ICCV)*, 2017.

[149] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[150] Jayant Krishnamurthy and Thomas Kollar. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Transactions of the Association for Computational Linguistics*, 1:193–206, 2013.

[151] Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1051. URL `https://www.aclweb.org/anthology/D19-1051`.

[152] Hilde Kuehne, Ahsan Iqbal, Alexander Richard, and Juergen Gall. Mining youtube-a dataset for learning fine-grained action concepts from webly supervised video data. *arXiv preprint arXiv:1906.01012*, 2019.

[153] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[154] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit,

Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

[155] Alice Lai, Yonatan Bisk, and Julia Hockenmaier. Natural language inference from multiple premises. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 100–109, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing. URL `http://www.aclweb.org/anthology/I17-1011`.

[156] Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2873–2882. PMLR, 2018.

[157] Himabindu Lakkaraju, Julian J. McAuley, and Jure Leskovec. What's in a name? understanding the interplay between titles, content, and communities in social media. In *ICWSM*, 2013.

[158] Jatin Lamba, Jayaprakash Akula, Rishabh Dabral, Preethi Jyothi, Ganesh Ramakrishnan, et al. Cross-modal learning for audio-visual video parsing. *arXiv preprint arXiv:2104.04598*, 2021.

[159] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2019.

[160] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. *ICLR*, 2017.

[161] Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. Adversarial filters of dataset biases. In *International Conference on Machine Learning*, 2020.

[162] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. In *EMNLP*, 2018.

[163] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvqa+: Spatio-temporal grounding for video question answering. In *Tech Report, arXiv*, 2019.

[164] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. What is more likely to happen next? video-and-language future event prediction. *arXiv preprint arXiv:2010.07999*, 2020.

[165] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341, 2021.

[166] Marco Leo, G Medioni, M Trivedi, Takeo Kanade, and Giovanni Maria Farinella. Computer vision for assistive technologies. *Computer Vision and Image Understanding*, 154:1–15, 2017.

[167] Rezka Leonandya, Dieuwke Hupkes, Elia Bruni, and Germán Kruszewski. The fast and the flexible: Training neural networks to learn to follow instructions from small data. In *Proceedings of the 13th International Conference on Computational Semantics-Long Papers*, pages 223–234, 2019.

[168] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, pages 11336–11344, 2020.

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

[170] Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1445–1455, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P16-1137`.

[171] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[172] Xiao Lin and Devi Parikh. Leveraging visual question answering for image-caption ranking. In *European Conference on Computer Vision*, pages 261–277. Springer, 2016.

[173] Xudong Lin, Gedas Bertasius, Jue Wang, Shih-Fu Chang, Devi Parikh, and Lorenzo Torresani. Vx2text: End-to-end learning of video-based text generation from multimodal inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7005–7015, 2021.

[174] Jing Liu, Xinxin Zhu, Fei Liu, Longteng Guo, Zijia Zhao, Mingzhen Sun, Weining Wang, Jinqiao Wang, and Hanqing Lu. Opt: Omni-perception pre-trainer for cross-modal understanding and generation. *arXiv preprint arXiv:2107.00249*, 2021.

[175] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[176] Peter LoBue and Alexander Yates. Types of common-sense knowledge needed for recognizing textual entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 329–334. Association for Computational Linguistics, 2011.

[177] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[178] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, 2016.

[179] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23, 2019.

[180] Li Lucy and Jon Gauthier. Are distributional representations ready for the real world? evaluating word vectors for grounded perceptual meaning. In *Proceedings of the First Workshop on Language Grounding for Robotics*, pages 76–85, 2017.

[181] Tegan Maharaj, Nicolas Ballas, Anna Rohrbach, Aaron C Courville, and Christopher Joseph Pal. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. URL `http://openaccess.thecvf.com/content_cvpr_2017/papers/Maharaj_A_Dataset_and_CVPR_2017_paper.pdf`.

[182] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. In *NAACL*, 2015.

[183] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[184] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL `https://www.aclweb.org/anthology/J93-2004`.

[185] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella bernardi, and Roberto Zamparelli. A sick cure for the evaluation of compositional distributional semantic models. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). ISBN 978-2-9517408-8-4. URL `http://www.lrec-conf.org/proceedings/lrec2014/pdf/363_Paper.pdf`. ACL Anthology Identifier: L14-1314.

[186] Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. A joint model of language and perception for grounded attribute learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1435–1442, 2012.

[187] James L McClelland, Felix Hill, Maja Rudolph, Jason Baldridge, and Hinrich Schütze. Extending machine language models toward human-level language understanding. *arXiv preprint arXiv:1912.05877*, 2019.

[188] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 2019.

[189] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020.

[190] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016.

[191] Heather Molyneaux, Susan O'Donnell, Kerri Gibson, Janice Singer, et al. Exploring the gender divide on youtube: An analysis of the creation and reception of vlogs. *American Communication Journal*, 10(2):1–14, 2008.

[192] Yasufumi Moriya, Ramon Sanabria, Florian Metze, and Gareth JF Jones. Grounding object detections with transcriptions. *arXiv preprint arXiv:1906.06147*, 2019.

[193] Nasrin Mostafazadeh, Nathanael Chambers, Xiadong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and evaluation framework for deeper understanding of commonsense stories. In *NAACL*, 2016. URL `http://arxiv.org/abs/1604.01696`.

[194] Roozbeh Mottaghi, Mohammad Rastegari, Abhinav Gupta, and Ali Farhadi. "what happens if..." learning to predict the effect of forces in images. In *European Conference on Computer Vision*, pages 269–285. Springer, 2016.

[195] Lev Muchnik, Sinan Aral, and Sean J. Taylor. Social influence bias: a randomized experiment. *Science*, 341 6146:647–51, 2013.

[196] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[197] Shruti Palaskar, Jindrich Libovickỳ, Spandana Gella, and Florian Metze. Multimodal abstractive summarization for how2 videos. *arXiv preprint arXiv:1906.07901*, 2019.

[198] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernandez. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P16-1144`.

[199] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[200] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, 2016.

[201] Ramakanth Pasunuru and Mohit Bansal. Multi-task video captioning with video and entailment generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1273–1283, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `http://aclweb.org/anthology/P17-1117`.

[202] Ramakanth Pasunuru and Mohit Bansal. Multi-reward reinforced summarization with saliency and entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 646–653. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/N18-2102`.

[203] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[204] Fernando Pereira. Formal grammar and information theory: together again? *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 358(1769):1239–1253, 2000.

[205] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computa-*

tional Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1202.

[206] Jean Piaget. *The origins of intelligence in children*, volume 8. International Universities Press New York, 1952.

[207] Hamed Pirsiavash, Carl Vondrick, and Antonio Torralba. Inferring the why in images. *arXiv preprint arXiv:1406.5472*, 2014.

[208] Bryan A Plummer, Arun Mallya, Christopher M Cervantes, Julia Hockenmaier, and Svetlana Lazebnik. Phrase localization and visual relationship detection with comprehensive image-language cues. In *Proc. ICCV*, 2017.

[209] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *Int. J. Comput. Vision*, 123(1):74–93, May 2017. ISSN 0920-5691. doi: 10.1007/s11263-016-0965-7. URL https://doi.org/10.1007/s11263-016-0965-7.

[210] Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis Only Baselines in Natural Language Inference. *arXiv:1805.01042 [cs]*, May 2018. URL http://arxiv.org/abs/1805.01042. arXiv: 1805.01042.

[211] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3282–3289. IEEE, 2012.

[212] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018. URL https://blog.openai.com/language-unsupervised/.

[213] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL `https://openai.com/blog/better-language-models/`.

[214] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[215] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.

[216] Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming language priors in visual question answering with adversarial regularization. In *Advances in Neural Information Processing Systems*, 2018.

[217] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[218] Nicholas Rhinehart and Kris M Kitani. First-person activity forecasting with online inverse reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3696–3705, 2017.

[219] Manoel Horta Ribeiro, Raphael Ottoni, Robert West, Virgílio AF Almeida, and Wagner Meira Jr. Auditing radicalization pathways on youtube. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 131–141, 2020.

[220] Neil M Richards. The dangers of surveillance. *Harv. L. Rev.*, 126:1934, 2012.

[221] Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 2011.

[222] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision*, pages 817–834. Springer, 2016.

[223] Anna Rohrbach, Marcus Rohrbach, Siyu Tang, Seong Joon Oh, and Bernt Schiele. Generating descriptions with grounded and co-referenced people. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, Piscataway, NJ, USA, July 2017. IEEE.

[224] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie Description. *International Journal of Computer Vision*, 123(1):94–120, May 2017. ISSN 1573-1405. doi: 10.1007/s11263-016-0987-1. URL https://doi.org/10.1007/s11263-016-0987-1.

[225] Andrew Rouditchenko, Angie Boggust, David Harwath, Brian Chen, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Hilde Kuehne, Rameswar Panda, Rogerio Feris, et al. Avlnet: Learning audio-visual language representations from instructional videos. *arXiv preprint arXiv:2006.09199*, 2020.

[226] Rachel Rudinger, Chandler May, and Benjamin Van Durme. Social bias in elicited natural language inferences. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 74–79, 2017.

[227] Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M Lake. A benchmark for systematic generalization in grounded language understanding. *Advances in Neural Information Processing Systems*, 33, 2020.

[228] Maarten Sap, Marcella Cindy Prasettio, Ari Holtzman, Hannah Rashkin, and Yejin Choi. Connotation frames of power and agency in modern films. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2329–2334, 2017.

[229] Roger C. Schank and Robert P. Abelson. Scripts, plans, and knowledge. In *Proceedings of the 4th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'75, pages 151–157, San Francisco, CA, USA, 1975. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1624626.1624649.

[230] Alexandra Schofield and Leo Mehr. Gender-distinguishing features in film dialogue. In *Proceedings of the Fifth Workshop on Computational Linguistics for Literature*, pages 32–39, 2016.

[231] Roy Schwartz, Maarten Sap, Ioannis Konstas, Li Zilles, Yejin Choi, and Noah A. Smith. The effect of different writing tasks on linguistic style: A case study of the ROC story cloze task. In *Proc. of CoNLL*, 2017.

[232] Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. Answers unite! unsupervised metrics for reinforced summarization models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3246–3256, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1320. URL https://www.aclweb.org/anthology/D19-1320.

[233] Ozan Sener, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Unsupervised semantic parsing of video collections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4480–4488, 2015.

[234] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare

words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, 2016.

[235] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[236] Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 752–757, 2018.

[237] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*, 2021.

[238] Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3407–3412, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1339. URL `https://www.aclweb.org/anthology/D19-1339`.

[239] Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, and Ming Zhou. Dense procedure captioning in narrated instructional videos. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6382–6391, 2019.

[240] Wei Shi and Vera Demberg. Next sentence prediction helps implicit discourse relation classification within and across domains. In *Proceedings of the 2019 Conference on Empirical*

Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5794–5800, 2019.

[241] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10740–10749, 2020.

[242] Krishna Kumar Singh, Kayvon Fatahalian, and Alexei A Efros. Krishnacam: Using a longitudinal, single-person, egocentric dataset for scene understanding tasks. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016.

[243] Linda Smith and Michael Gasser. The development of embodied cognition: Six lessons from babies. *Artificial life*, 11(1-2):13–29, 2005.

[244] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934, 2013.

[245] Robert Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. pages 4444–4451, 2017. URL `http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972`.

[246] Mitchell Stern, Jacob Andreas, and Dan Klein. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 818–827, 2017.

[247] Michael Strangelove. *Watching YouTube*. University of Toronto press, 2020.

[248] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, 2019.

[249] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*, 2019.

[250] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A joint model for video and language representation learning. In *ICCV*, 2019.

[251] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5103–5114, 2019.

[252] Zineng Tang, Jie Lei, and Mohit Bansal. Decembert: Learning from noisy instructional videos via dense captions and entropy minimization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2415–2426, 2021.

[253] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640, 2016.

[254] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[255] Atousa Torabi, Niket Tandon, and Leon Sigal. Learning language-visual embedding for movie understanding with natural-language. *arXiv preprint*, 2016. URL `http://arxiv.org/pdf/1609.08124v1.pdf`.

[256] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE, 2011.

[257] Alan M. Turing. Computing Machinery and Intelligence. *Mind*, LIX(236):433–460, 10 1950. ISSN 0026-4423. doi: 10.1093/mind/LIX.236.433. URL https://doi.org/10.1093/mind/LIX.236.433.

[258] Aisha Urooj, Amir Mazaheri, Mubarak Shah, et al. Mmft-bert: Multimodal fusion transformer with bert encodings for visual question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4648–4660, 2020.

[259] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Information Science and Statistics. Springer-Verlag, New York, 2 edition, 2000. ISBN 978-0-387-98780-4. URL //www.springer.com/us/book/9780387987804.

[260] Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. Fill in the blanc: Human-free quality estimation of document summaries. *arXiv preprint arXiv:2002.09836*, 2020.

[261] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010. Curran Associates Inc., 2017.

[262] Anu Venkatesh, Chandra Khatri, Ashwin Ram, Fenfei Guo, Raefer Gabriel, Ashish Nagar, Rohit Prasad, Ming Cheng, Behnam Hedayatnia, Angeliki Metallinou, et al. On evaluating and comparing open domain dialog systems. *arXiv preprint arXiv:1801.03625*, 2018.

[263] Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. Moviegraphs: Towards understanding human-centric situations from videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[264] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–106, 2016.

[265] Misha Wagner, Hector Basevi, Rakshith Shetty, Wenbin Li, Mateusz Malinowski, Mario Fritz, and Ales Leonardis. Answering visual what-if questions: From actions to predicted scene descriptions. In *Visual Learning and Embodied Agents in Simulation Environments Workshop at European Conference on Computer Vision*, 2018. URL https://arxiv.org/abs/1809.03707https://arxiv.org/pdf/1809.03707.pdf.

[266] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016.

[267] Alex Wang and Kyunghyun Cho. Bert has a mouth, and it must speak: Bert as a markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, 2019.

[268] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 3261–3275. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8589-superglue-a-stickier-benchmark-for-general-purpose-language-unders
pdf.

[269] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. 2019. In the Proceedings of ICLR.

[270] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2495–2504, 2021.

[271] Luyu Wang, Pauline Luc, Adria Recasens, Jean-Baptiste Alayrac, and Aaron van den Oord. Multimodal self-supervised learning of general audio representations. *arXiv preprint arXiv:2104.12807*, 2021.

[272] Peng Wang, Qi Wu, Chunhua Shen, Anton van den Hengel, and Anthony R. Dick. Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[273] Sida I Wang, Percy Liang, and Christopher D Manning. Learning language games through interaction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2368–2378, 2016.

[274] Xiaolong Wang, Ali Farhadi, and Abhinav Gupta. Actions ~ transformations. In *CVPR*, 2016.

[275] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

[276] Zeerak Waseem, Smarika Lulz, Joachim Bingel, and Isabelle Augenstein. Disembodied machine learning: On the illusion of objectivity in nlp. *arXiv preprint arXiv:2101.11974*, 2021.

[277] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8052–8060, 2018.

[278] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/N18-1101`.

[279] Samuel Williams, Andrew Waterman, and David Patterson. Roofline: An insightful visual performance model for floating-point programs and multicore architectures. Technical report, Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), 2009.

[280] Langdon Winner. Do artifacts have politics? *Daedalus*, 109(1), 1980.

[281] Ludwig Wittgenstein. *Philosophical Investigations*. Wiley-Blackwell, 1953.

[282] Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B. Tenenbaum, and Chuang Gan. Star: A benchmark for situated reasoning in real-world videos. *2021 Conference on Neural Information Processing Systems*, 2021.

[283] Qi Wu, Peng Wang, Chunhua Shen, Anthony R. Dick, and Anton van den Hengel. Ask me anything: Free-form visual question answering based on knowledge from external sources. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4622–4630, 2016.

[284] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1645–1653, 2017.

[285] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, and Florian Metze Luke Zettlemoyer Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021.

[286] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. *arXiv preprint arXiv:2201.04288*, 2022.

[287] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. *arXiv preprint arXiv:2012.00451*, 2020.

[288] Mark Yatskar, Vicente Ordonez, and Ali Farhadi. Stating the obvious: Extracting visual common sense knowledge. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 193–198, 2016.

[289] Tian Ye, Xiaolong Wang, James Davidson, and Abhinav Gupta. Interpretable intuitive physics model. In *European Conference on Computer Vision*, pages 89–105. Springer, 2018.

[290] Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*, 2019.

[291] Yuya Yoshikawa, Jiaqing Lin, and Akikazu Takeuchi. Stair actions: A video dataset of everyday home actions. *arXiv preprint arXiv:1804.04326*, 2018.

[292] Chen Yu and Linda B Smith. Embodied attention and word learning by toddlers. *Cognition*, 125(2):244–262, 2012.

[293] Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-vil: Knowledge enhanced vision-language representations through scene graph. *arXiv preprint arXiv:2006.16934*, 2020.

[294] Licheng Yu, Eunbyung Park, Alexander C. Berg, and Tamara L. Berg. Visual Madlibs: Fill in the blank Image Generation and Question Answering. *arXiv:1506.00278 [cs]*, May 2015. URL `http://arxiv.org/abs/1506.00278`. arXiv: 1506.00278.

[295] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer, 2016.

[296] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. A joint speakerlistener-reinforcer model for referring expressions. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[297] Shoou-I Yu, Lu Jiang, and Alexander Hauptmann. Instructional videos for unsupervised harvesting and learning of action examples. In *ACM MM*, 2014.

[298] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 471–487, 2018.

[299] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. ActivityNet-QA: a dataset for understanding complex web videos via question answering. In *AAAI*, pages 9127–9134, 2019.

[300] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[301] Rowan Zellers and Yejin Choi. Zero-shot activity recognition with verb attribute induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017. URL `https://arxiv.org/abs/1707.09468`.

[302] Rowan Zellers and Yejin Choi. Zero-shot activity recognition with verb attribute induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 946–958, 2017.

[303] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1009. URL `https://www.aclweb.org/anthology/D18-1009`.

[304] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[305] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[306] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://www.aclweb.org/anthology/P19-1472.

[307] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[308] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32*, 2019.

[309] Rowan Zellers, Ari Holtzman, Elizabeth Clark, Lianhui Qin, Ali Farhadi, and Yejin Choi. Turingadvice: A generative and dynamic evaluation of language use. In *NAACL*, 2021. URL https://arxiv.org/abs/2004.03607.

[310] Rowan Zellers, Ari Holtzman, Matthew Peters, Roozbeh Mottaghi, Aniruddha Kembhavi, Ali Farhadi, and Yejin Choi. PIGLeT: Language grounding through neuro-symbolic interaction in a 3D world. In *ACL*, 2021.

[311] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi,

and Yejin Choi. Merlot: Multimodal neural script knowledge models. *arXiv preprint arXiv:2106.02636*, 2021.

[312] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[313] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers, 2021.

[314] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Conference on Artificial Intelligence, Ethics and Society*, 2018.

[315] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. *arXiv preprint arXiv:2101.00529*, 2021.

[316] Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. Ordinal Commonsense Inference. *Transactions of the Association for Computational Linguistics*, 5:379–395, 2017.

[317] Tianyi Zhang, V. Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675, 2020.

[318] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*, 2020.

[319] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989, 2017.

[320] Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018.

[321] Sharon Zhou, Mitchell Gordon, Ranjay Krishna, Austin Narcomey, Li Fei-Fei, and Michael Bernstein. Hype: A benchmark for human eye perceptual evaluation of generative models. In *Advances in Neural Information Processing Systems*, pages 3444–3456, 2019.

[322] Yipin Zhou and Tamara L Berg. Temporal perception and prediction in ego-centric video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4498–4506, 2015.

[323] Linchao Zhu and Yi Yang. ActBERT: Learning global-local video-text representations. In *CVPR*, 2020.

[324] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7W: Grounded Question Answering in Images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[325] Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *arXiv preprint arXiv:1506.06724*, 2015.

[326] Shoshana Zuboff. Big other: surveillance capitalism and the prospects of an information civilization. *Journal of Information Technology*, 30(1):75–89, 2015.

[327] Geoffrey Zweig and Christopher JC Burges. The microsoft research sentence completion challenge. Technical report, Citeseer, 2011.