FROM PHYSICAL TO SOCIAL COMMONSENSE: NATURAL LANGUAGE AND THE NATURAL WORLD

MAXWELL FORBES

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

Doctor of Philosophy

University of Washington
2021

Reading Committee:
Yejin Choi, Chair
Noah A. Smith
Fei Xia

Program Authorized to Offer Degree:
Paul G. Allen School of Computer Science & Engineering

University of Washington

ABSTRACT

FROM PHYSICAL TO SOCIAL COMMONSENSE: NATURAL LANGUAGE AND THE NATURAL WORLD

Maxwell Forbes

Chair of the Supervisory Committee:
Professor Yejin Choi
Paul G. Allen School of Computer Science & Engineering

Along with the meteoric rise of computation-hungry models, NLP research has also produced new handcrafted datasets. These datasets allow us to study problems that are difficult by web scraping alone. We can use such data to evaluate and extend machine learning models into new areas. One area of natural interest is work that connects NLP to the outside world.

This dissertation describes four projects that present such datasets and computational models. Each project attempts to situate NLP in a context broader than text alone. As a common thread throughout, we make use of commonsense knowledge, either explicitly or implicitly. The first half of the dissertation covers two projects, **Verb Physics** and **Social Chemistry**, which contain explicit representations of commonsense knowledge. Respectively, they capture physical commonsense (e.g., that *my house is bigger than I am*) and social commonsense (e.g., that *it's rude for my roommate to run the blender at 5am*). The second half studies language production and evaluation. In this half, commonsense implicitly informs the work. **Neural Naturalist** addresses language generation from image comparisons. **Scarecrow** focuses on evaluating text generated by large language models.

In the conclusion, we urge the field to embrace *communication*—not merely natural language—and thereby extend the richness of groundings we consider.

To my parents.

ACKNOWLEDGMENTS

First and foremost, thank you to my advisor, Yejin. I have learned countless lessons from you, most intangible and difficult to teach. Most importantly, thank you for deep and enduring support to a degree that I couldn't have understood before doing the PhD.

Heartfelt thanks to those who helped me enter the world of AI research. Working with an undergrad, especially one like me, is a huge donation of one's time—and the undergrad, especially one like me, doesn't always realize it at the time. Thank you to Raj, for taking a chance on me; Luke, for always lending an ear; and Maya, for showing me the ropes. Thanks to Yoav, for support and advice; Kenton, for kindness paired with utmost clarity; and Mike, to whom I owe so much: through years of mentorship, you have treated me as a friend from day one.

In a way of looking, the communities are the experience. I'm so grateful for those I've been a part of. Thank you to my closest PhD comrades, Ari, Chris, Max, Peter, and Rahul—you have made an enormous difference. Thanks to my fellow first-year NLPers and MLBootcampers for your continued camaraderie: Colin, Elizabeth, Gagan, John, Julian, Kelvin, Lucy, Maarten, Mandar, Nelson, and Phoebe. Thanks also to my fellow xlabmates, former and present: Alisa, Antoine, Chloé, Eunsol, Gary, Ge, Hannah, Jack, Jaehun, James, Jan, Karen, Liwei, Melanie, Rachel, Raj, Rowan, Saadia, Sean, Swabha, Vered, Ximing, Xiujun, Yannis, and Yonatan. Along with the rest of Mosaic, special thanks to Nick, for inspiration and engaging conversations, and Jena, for monumental help in our collaborations. At Google, particular thanks to Christine, Serge, Kiat, and Dominic for being so welcoming and supportive. Thanks to the small handful of undergrads I've been lucky to mentor: Jeff, Pooja, and Yao; I have learned a great deal from you. Finally, my local community kept me happy and grounded. To risk naming a few, thank you to Alex and Shelby, Callan and Mere, Cooper and Alexis, Greg, Harnoor and Paige, Jim and Jory, Michael C. and Jolyn, Tyler and Michael G., and Woody and Kristen.

Exiting school at the age of thirty, it's difficult not to trace the threads back even further and see the support and encouragement of past teachers. I have gone to public schools my entire life. To omit any is to do a disservice, but I must offer special thanks for my love of math, music, learning, and communication to Robert Femiano, Doug Swan, Jorge Morales, Marcus Pimpleton, Debbie Meyer Snook, Peter Junkerman, Mark Wangerin, David Katz, and Hélène Martin.

Finally, immense gratitude to my family, a bedrock of love and kindness: Ben and Betsy; Shannon and Jesús; and, of course, to Julie—you have brought me so much happiness in every step of this journey.

CONTENTS

1 INTRODUCTION

1.1 COMMUNICATION AND COMMONSENSE

In this dissertation, we seek to study natural language in its broader context, which we term *the natural world.* Along the way, we will often turn to *commonsense* as a tool to reveal hidden mechanics behind our everyday reasoning and interactions.

To begin, let us first get our bearings with these terms.

TEXT Natural language processing (NLP) primarily studies text. Broadly speaking, the field trains computers to make predictions using text as input, output, or both (Jurafsky and Martin, 2000; Smith, 2011). Historical problems of interest have included syntactic parsing (Charniak, 2000), machine translation (Koehn, 2009), question answering (Rajpurkar et al., 2016), and many others.

COMMUNICATION As humans, our use of natural language is much messier than what is represented in transcribed text. What we say depends on myriad social factors, such as cultural norms, individual ideologies, and the social relations of the participants (Hovy and Yang, 2021). Not just *what* we say but also *how* we say it is also complex, and includes factors like prosody (Trott et al., 2019) and all of body language (Mehrabian and Wiener, 1967). As a backdrop to language use, we have underlying communication goals, though these are rarely stated (Bernstein, 1960).

In fact, the domain of "that which is rarely stated" extends well beyond communication goals. There turns out to be a vast pool of implicit, everyday knowledge that, because we all share, we do not usually describe. The term that has emerged for this kind of knowledge and reasoning is "commonsense."

COMMONSENSE It is tricky even to say exactly what "commonsense" is. As a working definition, let us consider commonsense knowledge to be an understanding of how the world works that we typically do not write down. Two examples are *that my house is bigger than me (physical commonsense)*, and *that it's rude for your roommate to run the blender at 5am (social commonsense).* That commonsense is rarely written down has been termed reporting bias (Gordon and Van Durme, 2013); its tacit nature can be justified as omitting information too obvious to share (Grice, 1975).

Another characteristic that makes commonsense reasoning challenging to study is that it is often defeasible (Rudinger et al., 2020). Statements of commonsense, such as *that it's rude to wake someone up in the middle of the night,* nearly always have exceptions (e.g., *the house is on fire*).

There has been a recent resurgence of NLP efforts to encode commonsense knowledge. After long periods of time between older works such as CYC (Lenat, 1995) and ConceptNet (Liu and Singh, 2004), several commonsense corpora have sprung up for reasoning about events, like Event2Mind (Rashkin et al., 2018) and ATOMIC (Sap et al., 2019b). In addition to these larger resources, researchers have also developed more specialized troves of commonsense data for specific domains, such as defeasible inference (Rudinger et al., 2020) and social intelligence (Sap et al., 2019c).

The spirit of this dissertation is to situate text more broadly in an effort to bring NLP closer to a study of communication. We examine problems where language is used as a medium, but the objects of focus are commonsense knowledge. Other tasks emphasize language more directly, but in service of an objective that includes external phenomena, like images or world knowledge. In these later tasks, commonsense still plays an implicit role.

Given the centrality of commonsense to our work, it is worth a brief detour into how we study commonsense, which is by explicit representation. Recent success in machine learning has been driven by harnessing ever increasing computational power. In the next section, we comment on the potential tension between this trend, and our practice of collecting handcrafted commonsense datasets to study new problems.

1.2 COMPUTATION AND KNOWLEDGE

One lens through which we may study progress in machine learning is through the tension between leveraging computing power[1] and encoding knowledge. Sutton (2019) provides one account of how this has played out in his essay, *The Bitter Lesson:*

> The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin.
> ...
> Seeking an improvement that makes a difference in the shorter term, researchers seek to leverage their human knowledge of the domain, but the only thing that matters in the long run is the leveraging of computation.
> ...
> The bitter lesson is based on the historical observations that 1) AI researchers have often tried to build knowledge into their agents, 2) this always helps in the short term, and is personally satisfying to the researcher, but 3) in the long run it plateaus and even inhibits further progress, and 4) breakthrough progress eventually arrives by an opposing approach based on scaling computation by search and learning. The eventual success is tinged with bitterness, and often incompletely digested, because it is success over a favored, human-centric approach.

1 Here by *computation* we also include *data*, the yin to its yang.

Much of the work presented in this dissertation explicitly encodes commonsense knowledge, and studies its use in AI.[2] But what of the bitter lesson?

In the previous section, we reviewed a brief history of NLP work representing commonsense knowledge. NLP is also no stranger to harnessing computation.

COMPUTATION The rise of statistical natural language processing itself was possible due to increased computational power (Manning and Schutze, 1999). Even focusing only the most recent trends, we see the rise of pretrained word representations of word2vec (Mikolov et al., 2013), GloVe (Pennington, Socher, and Manning, 2014), and ELMo (Peters et al., 2018); masked language model representations from BERT (Devlin et al., 2019) and its descendants like RoBERTa (Liu et al., 2019a); and models of increasingly gargantuan scale trained with forward language modeling losses, like the GPT family (Brown et al., 2020; Radford et al., 2018a; 2019b), and encoder-deocder models like T5 (Raffel et al., 2019). These papers report staggering performance metrics on suites of NLP benchmark tasks, several of which eclipse some measures of human performance (Wang et al., 2019).[3]

Given the remarkable success of such recent efforts that leverage computation, are our efforts to encode commonsense knowledge in vain? Will the ocean of computation ultimately blast apart any sandcastles of painstakingly collected data?

We must truthfully confess that we do not know.

However, there is one line of *The Bitter Lesson* that easily overlooked:

> These two need not run counter to each other, but in practice they tend to.

Behind the scenes, commonsense knowledge research in the last few years has been grappling with the rise of powerful computation-driven models. Several approaches have emerged that propose how to leverage the strengths of both computation and knowledge to achieve what neither has been able to alone. We present here a brief overview of four such approaches.

1. One opportunity for specialized knowledge is to be used for **evaluating** models in new and interesting domains. While computation can scale the size and capabilities of large models, it does not help us assess how well a model actually understands[4] a domain that we care about, but for which natural data simply does not exist (Sap et al., 2019a).

2 *AI* and *machine learning* will be used interchangeably, with apologies.

3 Despite temptation, we omit here a discussion of *solving the dataset but not the task*, but mention it as a vital caveat to any notions of models reaching human performance.

4 Words like *understand* will be consciously used in this dissertation, again with apologies, when the intuitive benefit outweighs the sloppiness of anthropomorphization.

2. Closely related to evaluation is the practice of **probing** (Jawahar, Sagot, and Seddah, 2019; Tenney, Das, and Pavlick, 2019), an activity that has emerged recently due to the prevalence of black-box neural network models. Rather than asking how well a model performs at a commonsense task, we may use the structure of the data to define dimensions along which to test a model's decision boundaries. In other words, specialized knowledge allows us to select dimensions to investigate based on real world interest, rather than due to a model's particular structure or behavior (Ilharco et al., 2021).

3. The most straightforward merging of large pretrained models and specialized data is **finetuning**, a practice currently so common it is easy to overlook. In this, we take a general-purpose model, which benefited from large amounts of computation to during initial training, and train for comparatively few additional cycles on our smaller collection of knowledge. This practice is commonplace broadly in NLP, where it is used for specific domains or tasks (Devlin et al., 2019), and commonsense knowledge tasks are no exception (Bhagavatula et al., 2020).

4. Finally, perhaps the most recent and exciting merger of computation and knowledge is through **model to data distillation**. Unlike model distillation (Hinton, Vinyals, and Dean, 2015), where a model is shrunk, or data distillation (Wang et al., 2018), where data is shrunk, distilling a model to data allows for contextually relevant and (hopefully) plausible data to be generated on-the-fly. In this approach, a smaller source of carefully constructed knowledge is used to prime large pretrained model. The model then generates data either to augment the original data source (West et al., 2021), or provide complementary but different information (Liu et al., 2021).

1.3 SCOPE OF THIS DISSERTATION

The preceding sections describe the broader aim of this dissertation, which is to study more phenomena surrounding language usage, with a particular eye towards commonsense. They also paint a picture of why we might still care about handcrafted knowledge even in an era of enormous computational resources.

We now discuss the work to be presented. The scope of this dissertation is four places where language meets other aspects of the world: naive physics, normative behaviors, visual stimuli, and event factuality.

In each of these studies, commonsense knowledge plays a different role. In the first two, it is explicitly represented, either by formal logic or freely structured text. In the latter two, it is used implicitly as a mechanism for producing or evaluating language.

The chapters are as follows:

EXPLICIT COMMONSENSE

- Chapter 2 presents **Verb Physics**, a study of how everyday language can reveal physical commonsense knowledge if we know how to read between the lines.

  This chapter was previously published as: Maxwell Forbes and Yejin Choi (2017). "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)*.

- Chapter 3 presents **Social Chemistry**, a project that seeks to capture the rich domain of social, moral, and ethical norms in a collection of pithy rules-of-thumb.

  This chapter was previously published as: Maxwell Forbes, Jena D. Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi (2020). "Social Chemistry 101: Learning to Reason about Social and Moral Norms." In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

IMPLICIT COMMONSENSE

- Chapter 4 presents **Neural Naturalist**, in which annotators use an implicit commonsense understanding of similarity to construct comparisons between two images.

  This chapter was previously published as: Maxwell Forbes, Christine Kaeser-Chen, Piyush Sharma, and Serge Belongie (2019). "Neural Naturalist: Generating Fine-Grained Image Comparisons." In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

- Chapter 5 presents **Scarecrow**, where humans critique machine written text, using a commonsense understanding of how the world works to tease apart discrepancies between what is stated and the truth.

  At the time of writing, this chapter is under peer review, but a preprint has been published as: Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A. Smith, and Yejin Choi (2021). *Scarecrow: A Framework for Scrutinizing Machine Text*. arXiv: 2107.01294 [cs.CL].

Part I

EXPLICIT COMMONSENSE

2

VERB PHYSICS

Reading and reasoning about natural language text often requires trivial knowledge about everyday physical actions and objects. For example, given a sentence, *"Tyle could fit the trophy into the suitcase,"* we can infer that the trophy must be smaller than the suitcase, even though it is not stated explicitly. This reasoning requires knowledge about the action *"fit"*—in particular, typical preconditions that need to be satisfied in order to perform the action. In addition, reasoning about the applicability of various physical actions in a given situation often requires background knowledge about objects in the world, for example, that people are usually *smaller* than houses, that cars generally move *faster* than humans walk, or that a brick probably is *heavier* than a feather.

In fact, the potential use of such knowledge about everyday actions and objects can go beyond language understanding and reasoning. Many open challenges in computer vision and robotics may also benefit from such knowledge, as shown in work that requires visual reasoning and entailment (Izadinia et al., 2015; Zhu, Fathi, and Fei-Fei, 2014). Ideally, an AI system should acquire such knowledge through direct physical interactions with the world. However, such a physically interactive system does not seem feasible in the foreseeable future.

In this chapter, we present an approach to acquire high-level physical knowledge from unstructured natural language text as an alternative knowledge source. In particular, we focus on acquiring relative physical knowledge of actions and objects organized along five dimensions: size, weight, strength, rigidness, and speed. Figure 2.1 illustrates example knowledge of (1) relative physical relations of object pairs and (2) physical implications of actions when applied to those object pairs.

While natural language text is a rich source to obtain broad knowledge about the world, compiling trivial commonsense knowledge from unstructured text is a nontrivial feat. The central challenge lies in *reporting bias*: people rarely states the obvious (Gordon and Van Durme, 2013; Misra et al., 2016; Sorower et al., 2011; Van Durme, 2010; Zhang et al., 2017), since it goes against Grice's conversational maxim on the quantity of information (Grice, 1975).

We demonstrate that it is possible to overcome reporting bias and still extract the unspoken knowledge from language. The key insight is this: there is consistency in the way people describe how they interact with the world, which provides vital clues to reverse engineer the common knowledge shared among people. More concretely, we frame knowledge acquisition as joint inference over two closely related puzzles: inferring relative physical knowledge about object pairs while simultaneously reasoning about physical implications of actions.

Importantly, four of five dimensions of knowledge in our study—weight, strength, rigidness, and speed—are either not visual or not easily recognizable by image recognition using currently available computer vision techniques. Thus, our work provides

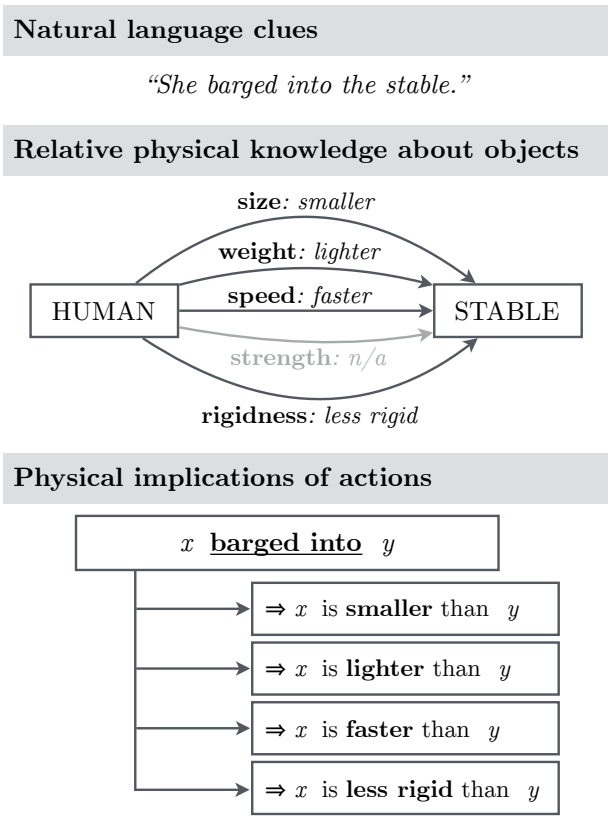


Figure 2.1: An overview of our approach. A verb's usage in language (top) implies physical relations between objects it takes as arguments. This allows us to reason about properties of specific objects (middle), as well as the knowledge implied by the verb itself (bottom).

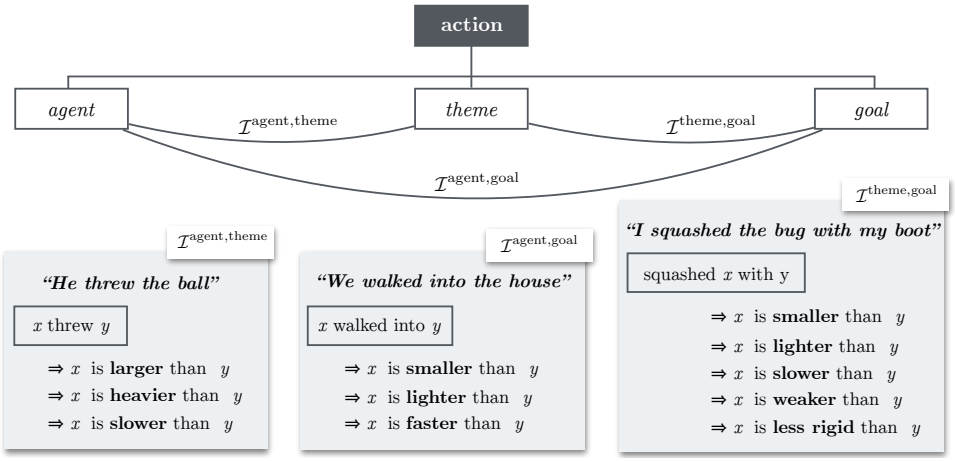


Figure 2.2: Example physical implications represented as frame relations between a pair of arguments.

unique value to complement recent attempts to acquire commonsense knowledge from web images (Bagherinezhad et al., 2016; Izadinia et al., 2015; Sadeghi, Kumar Divvala, and Farhadi, 2015).

In sum, the contributions described in this chapter are threefold:

- We introduce a new task in the domain of commonsense knowledge extraction from language, focusing on the physical implications of actions and the relative physical relations among objects, organized along five dimensions.
- We propose a model that can infer relations over grounded object pairs together with first order relations implied by physical verbs.
- We develop a new dataset VERBPHYSICS that compiles crowdsourced knowledge of actions and objects.[1]

The rest of the chapter is organized as follows. We first provide the formal definition of knowledge we aim to learn in Section 2.1. We then describe our data collection in Section 2.2 and present our inference model in Section 2.3. Empirical results are given in Section 2.4 and discussed in Section 2.5. We review related work in Section 2.6 and conclude in Section 2.7.

2.1 REPRESENTATION OF RELATIVE PHYSICAL KNOWLEDGE

2.1.1 *Knowledge Dimensions*

We consider five dimensions of relative physical knowledge in this work: *size, weight, strength, rigidness,* and *speed.* "Strength" in our work refers to the physical durability of an object (e.g., "diamond" is stronger than "glass"), while "rigidness" refers to the

1 https://uwnlp.github.io/verbphysics/

physical flexibility of an object (e.g., "glass" is more rigid than a "wire"). When considered in verb implications, *size, weight, strength,* and *rigidness* concern individual-level semantics; the relative properties implied by verbs in these dimensions are true in general. On the other hand, *speed* concerns stage-level semantics; its implied relations hold only during a window surrounding the verb (Carlson, 1977).

2.1.2 *Relative physical knowledge*

Let us first consider the problem of representing relative physical knowledge between two objects. We can write a single piece of knowledge like "A person is larger than a basketball" as

$$\texttt{person} >^{\text{size}} \texttt{basketball}$$

Any propositional statement can have exceptions and counterexamples. Moreover, we need to cope with uncertainties involved in knowledge acquisition. Therefore, we assume each piece of knowledge is associated with a probability distribution. More formally, given objects x and y, we define a random variable $O^a_{x,y}$ whose range is $\{\boxed{>}, \boxed{<}, \boxed{\simeq}\}$ with respect to a knowledge dimension $a \in \{\text{SIZE}, \text{WEIGHT}, \text{STRENGTH}, \text{RIGIDNESS}, \text{SPEED}\}$ so that:

$$\mathbb{P}(O^a_{x,y} = r), r \in \{\boxed{>}, \boxed{<}, \boxed{\simeq}\}.$$

This immediately provides two simple properties:

$$\mathbb{P}(O_{x,y} = \boxed{>}) = \mathbb{P}(O_{y,x} = \boxed{<})$$
$$\mathbb{P}(O_{x,x} = \boxed{\simeq}) = 1$$

2.1.3 *Physical Implications of Verbs*

Next we consider representing relative physical implications of actions applied over two objects. For example, consider an action frame "x threw y." In general, following implications are likely to be true:

$$\text{"}x \text{ threw } y\text{"} \implies x >^{\text{size}} y$$
$$\text{"}x \text{ threw } y\text{"} \implies x >^{\text{weight}} y$$
$$\text{"}x \text{ threw } y\text{"} \implies x <^{\text{speed}} y$$

Again, in order to cope with exceptions and uncertainties, we assume a probability distribution associated with each implication. More formally, we define a random variable F^a_v to denote the implication of the action verb v when applied over its arguments x and y with respect to a knowledge dimension a so that:

$$\mathbb{P}(F^{\text{size}}_{\text{threw}} = \boxed{>}) := \mathbb{P}(\text{"}x \text{ threw } y\text{"} \implies x >^{\text{size}} y)$$
$$\mathbb{P}(F^{\text{wgt}}_{\text{threw}} = \boxed{>}) := \mathbb{P}(\text{"}x \text{ threw } y\text{"} \implies x >^{\text{wgt}} y)$$

where the range of F^{size}_{threw} is $\{\boxed{>}, \boxed{<}, \boxed{\simeq}\}$. Intuitively, F^{size}_{threw} represents the likely first order relation implied by "throw" over ungrounded (i.e., variable) object pairs.

The above definition assumes that there is only a single implication relation for any given verb with respect to a specific knowledge dimension. This is generally not true, since a verb, especially a common action verb, can often invoke a number of different frames according to frame semantics (Fillmore, 1976). Thus, given a number of different frame relations $v_1 ... v_T$ associated with a verb v, we define random variables F with respect to a specific frame relation v_t, i.e., $F^a_{v_t}$. We use this notation going forward.

FRAME PERSPECTIVE OF VERB IMPLICATIONS Figure 2.2 illustrates the frame-centric view of physical implication knowledge we aim to learn. Importantly, the key insight of our work is inspired by Fillmore's original manuscript on frame semantics (). Fillmore has argued that "frames"—the contexts in which utterances are situated—should be considered as a third primitive of describing a language, along with a grammar and lexicon. While existing frame annotations such as FrameNet (Baker, Fillmore, and Lowe, 1998), PropBank (Palmer, Gildea, and Kingsbury, 2005), and VerbNet (Kipper, Dang, Palmer, et al., 2000) provide rich frame knowledge associated with a predicate, none of them provide the exact kind of physical implications we consider in our paper. Thus, our work can potentially contribute to these resources by investigating new approaches to automatically recover richer frame knowledge from language. In addition, our work is motivated by the formal semantics of Dowty (1991), as the task of learning verb implications is essentially that of extracting lexical entailments for verbs.

Since then, adaptations of frames semantics have captured a certain set of these properties. FrameNet (Baker, Fillmore, and Lowe, 1998) describes the informational semantics of an event with abstract frames (such as `Cause_motion`). These frames can contain a wide variety of details, from core information like the AGENT and THEME, to specifics such as the DURATION and DISTANCE. PropBank (Palmer, Gildea, and Kingsbury, 2005) and VerbNet (Kipper, Dang, Palmer, et al., 2000) are similarly concerned with such informational semantics, though with a greater focus on its relation to syntax.

But the general notion of "frames" contains more information than is currently encoded in modern frame annotations. Fillmore describes several additional components to frames, including a model of context (like the social statuses of the communicators, the time of day, who the audience is), a running model of the listener's understanding, and a set of prototypes on which listeners ground new concepts. Another component to frames, and the one we consider, is a model of the world.

2.2 DATA AND CROWDSOURCED KNOWLEDGE

ACTION VERBS We pick 50 classes of Levin verbs from both "alternation classes" and "verb classes" (Levin, 1993), which corresponds to about 1100 unique verbs. We sort this list by frequency of occurrence in our frame patterns in the Google Syntax Ngrams corpus (Goldberg and Orwant, 2013) and pick the top 100 verbs.

ACTION FRAMES Figure 2.2 illustrates examples of action frame relations. Because we consider implications over pairwise argument relations for each frame, there are sometimes multiple frame relations we consider for a single frame. To enumerate action frame relations for each verb, we use syntactic patterns based on dependency parse by extracting the core components (subject, verb, direct object, prepositional object) of an action, then map the subject to an agent, the direct object to a theme, and the prepositional object to a goal.[2] For those frames that involve an argument in a prepositional phrase, we create a separate frame for each preposition based on the statistics observed in the Google Syntax Ngram corpus.

Because the syntax ngram corpus provides only tree snippets without context, this way of enumerating potential frame patterns tend to over-generate. Thus we refine our prepositions for each frame by taking either the intersection or union with the top 5 Google Surface Ngrams (Michel et al., 2011), depending on whether the frame was under- or over-generating. We also add an additional crowdsourcing step where we ask crowd workers to judge whether a frame pattern with a particular verb and preposition could plausibly be found in a sentence. This process results in 813 frame templates, an average of 8.13 per verb.

OBJECT PAIRS To provide a source of ground truth relations between objects, we select the object pairs that occur in the 813 frame templates with positive pointwise mutual information (PMI) across the Google Syntax Ngram corpus. After replacing a small set of "human" nouns with a generic HUMAN object, filtering out nouns labeled as abstract by WordNet (Miller, 1995), and distilling all surface forms to their lemmas (also with WordNet), the result is 3656 object pairs.

2.2.1 *Crowdsourcing Knowledge*

We collect human judgements of the frame knowledge implications to use as a small set of seed knowledge (5%), a development set (45%), and a test set (50%) for Action Frame relations (2.2). Crowd workers are given with a frame template such as "**x** threw **y**," and then asked to list a few plausible objects (including people and animals) for the missing slots (e.g., **x** and **y**).[3] We then ask them to rate the general relationship that the arguments of the frame exhibit with respect to all knowledge dimensions (size, weight, etc.). For each knowledge dimension, or attribute, a, workers select an answer from (1) $\mathbf{x} >^a \mathbf{y}$, (2) $\mathbf{x} <^a \mathbf{y}$, (3) $\mathbf{x} \simeq^a \mathbf{y}$, or (4) no general relation.

We conduct a similar crowdsourcing step for the set of object pairs. We ask crowd workers to compare each of the 3656 object pairs along the five knowledge dimensions we consider, selecting an answer from the same options above as with frames. We reserve 50% of the data as a test set, and split the remainder up either 5% / 45% or 20% /

2 Future research could use an SRL parser instead. We use dependency parse to benefit from the Google Syntax Ngram dataset that provides language statistics over an extremely large corpus, which does not exist for SRL.

3 This step is to prime them for thinking about the particular template; we do not use the objects they provided.

Data collected		
	Total	**Seed / dev / test**
$\text{Verbs}_{5\%}$	100	5 / 45 / 50
$\text{Verbs}_{20\%}$	”	20 / 30 / 50
$\text{Frames}_{5\%}$	813	65 / 333 / 415
$\text{Frames}_{20\%}$	”	188 / 210 / 415
$\text{Object pairs}_{5\%}$	3656	183 / 1645 / 1828
$\text{Object pairs}_{20\%}$	”	733 / 1096 / 1828

Per attribute frame statistics				
	Agreement		*Counts (usable)*	
	2/3	**3/3**	**Verbs**	**Frames**
size	0.91	0.41	96	615
weight	0.90	0.33	97	562
strength	0.88	0.25	95	465
rigidness	0.87	0.26	89	432
speed	0.93	0.36	88	420

Per attribute object pair statistics				
	Agreement		*Counts (usable)*	
	2/3	**3/3**	**Distinct objs**	**Pairs**
size	0.95	0.59	210	2552
weight	0.95	0.56	212	2586
strength	0.92	0.43	208	2335
rigidness	0.91	0.39	212	2355
speed	0.90	0.38	209	2184

Table 2.1: Statistics of crowdsourced knowledge. Frames are partitioned by verb.

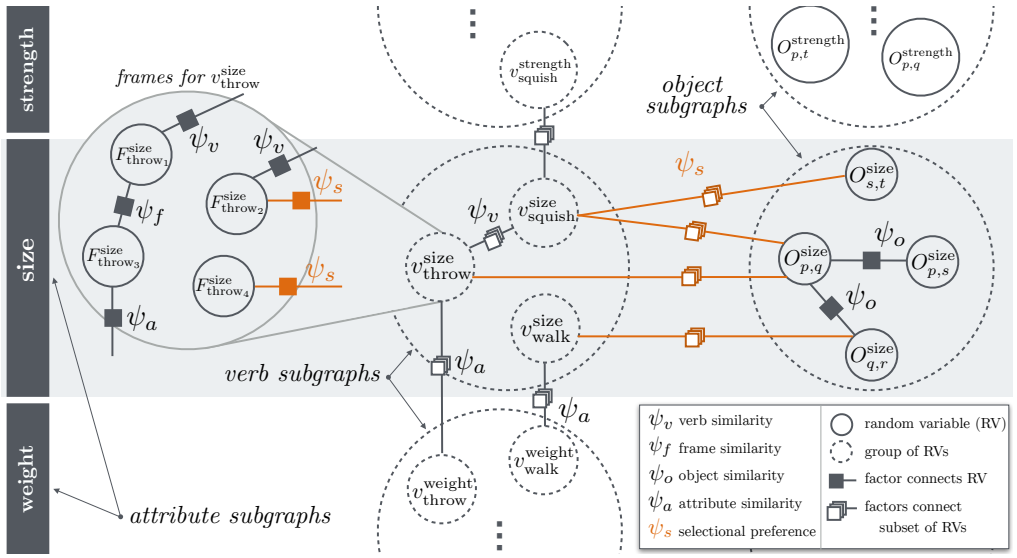


Figure 2.3: High level view of the factor graph model. Performance on both learning relative knowledge about objects (right), as well as entailed knowledge from verbs (center) via realized frames (left), is improved by modeling their interplay (orange). Unary seed (ψ_{seed}) and embedding (ψ_{emb}) factors are omitted for clarity.

30% (seed / development) to investigate the effects of different seed knowledge sizes on the model.

Statistics for the dataset are provided in Table 2.1. About 90% of the frames as well as object pairs had 2/3 agreement between workers. After removing frame/attribute combinations and object pairs that received less than 2/3 agreement, or were selected by at least 2/3 workers to have no relation, we end up with roughly 400–600 usable frames and 2100–2500 usable object pairs per attribute.

2.3 MODEL

We model knowledge acquisition as probabilistic inference over a factor graph of knowledge. As shown in Figure 2.3, the graph consists of multiple substrates (page-wide boxes) corresponding to different knowledge dimensions (shown only three of them —strength, size, weight—for brevity). Each substrate consists of two types of sub-graphs: verb subgraphs and object subgraphs, which are connected through factors that quantify action–object compatibilities. Connecting across substrates are factors that model inter-dependencies across different knowledge In what follows, we describe each graph component.

2.3.1 *Nodes*

The factor graph contains two types of nodes in order to capture two classes of knowledge. The first type of nodes are object pair nodes. Each object pair node is a random

variable $O^a_{x,y}$ which captures the relative strength of an attribute a between objects x and y.

The second type of nodes are frame nodes. Each frame node is a random variable $F^a_{v_t}$. This corresponds to the verb v used in a particular type of frame t, and captures the implied knowledge the frame v_t holds along an attribute a.

All random variables take on the values $\{\boxed{>}, \boxed{<}, \boxed{\simeq}\}$. For an object pair node $O^a_{x,y}$, the value represents the belief about the relation between x and y along the attribute a. For a frame node $F^a_{v_t}$, the value represents the belief about the relation along the attribute a between *any* two objects that might be used in the frame v_t.

We denote the sets of all object pair and frame random variables $\mathcal{O}$ and $\mathcal{F}$, respectively.

2.3.2 *Action–Object Compatibility*

The key aspect of our work is to reason about two types of knowledge simultaneously: relative knowledge of grounded object pairs, and implications of actions related to those objects. Thus we connect the verb subgraphs and object subgraphs through selectional preference factors ψ_s between two such nodes $O^a_{x,y}$ and $F^a_{v_t}$ if we find evidence from text that suggests objects x and y are used in the frame v_t. These factors encourage both random variables to agree on the same value.

As an example, consider a node $O^{size}_{p,b}$ which represents the relative size of a person and a basketball, and a node $F^{size}_{threw_{dobj}}$ which represents the relative size implied by an *"x threw y"* frame. If we find significant evidence in text that *"[person] threw [basketball]"* occurs, we would add a selectional preference factor to connect $O^{size}_{p,b}$ with $F^{size}_{threw_{dobj}}$ and encourage them towards the same value. This means that if it is discovered that people are larger than basketballs (the value $\boxed{>}$), then we would expect the frame *"x threw y"* to entail $x >^{size} y$ (also the value $\boxed{>}$).

2.3.3 *Semantic Similarities*

Some frames have relatively sparse text evidences to support their corresponding knowledge acquisition. Thus, we include several types of factors based on semantic similarities as described below.

CROSS-VERB FRAME SIMILARITY We add a group of factors ψ_v between two verbs v and u (to connect a specific frame of v with a corresponding frame of u) based on the verb-level similarities.

WITHIN-VERB FRAME SIMILARITY Within each verb v, which consists of a set of frame relations $v_1, ...v_T$, we also include frame-level similarity factors ψ_f between v_i and v_j. This gives us more evidence over a broader range of frames when textual evidence might be sparse.

	Development						Test					
Algorithm	size	weight	stren	rigid	speed	*overall*	size	weight	stren	rigid	speed	*overall*
RANDOM	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
MAJORITY	0.38	0.41	0.42	0.18	**0.83**	0.43	0.35	0.35	0.43	0.20	0.88	0.44
EMB-MAXENT	0.62	0.64	0.60	**0.83**	**0.83**	0.69	0.55	0.55	0.59	0.79	0.88	0.66
OUR MODEL (A)	0.71	0.63	0.61	0.82	**0.83**	0.71	0.55	0.55	0.55	0.79	**0.89**	0.65
OUR MODEL (B)	**0.75**	**0.68**	**0.68**	0.82	0.78	**0.74**	**0.74**	**0.71**	**0.65**	**0.80**	0.87	**0.75**

	Development						Test					
Algorithm	size	weight	stren	rigid	speed	*overall*	size	weight	stren	rigid	speed	*overall*
RANDOM	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
MAJORITY	0.50	0.54	0.51	0.50	0.53	0.51	0.51	0.55	0.52	0.49	0.50	0.51
EMB-MAXENT	0.68	0.66	0.64	0.67	0.65	0.66	0.71	0.67	0.64	0.65	**0.63**	0.66
OUR MODEL (A)	0.74	0.69	0.67	**0.68**	**0.66**	0.69	0.68	0.70	0.66	**0.66**	0.60	0.66
OUR MODEL (B)	**0.75**	**0.74**	**0.71**	**0.68**	**0.66**	**0.71**	**0.75**	**0.76**	**0.72**	0.65	0.61	**0.70**

Table 2.2: Accuracy of baselines and our model on both tasks. Top: frame prediction task; bottom: object pair prediction task. In both tasks 5% of in-domain data (frames or object pairs, respectively) are available as seed data. We compare providing the other type of data (object pairs or frames, respectively) as seed knowledge, trying 5% (OUR MODEL (A)) and 20% (OUR MODEL (B)).

OBJECT SIMILARITY As with verbs, we add factors ψ_o that encourage similar pairs of objects to take the same value. Given that each node represents a pair of objects, finding that x and y are similar yields two main cases in how to add factors (aside from the trivial case where the variable $O^a_{x,y}$ is given a unary factor to encourage the value $\boxed{\simeq}$).

1. If nodes $O_{x,z}$ and $O_{y,z}$ exist, we expect objects x and y to both have a similar relation to z. We add a factor that encourages $O_{x,z}$ and $O_{y,z}$ to take the same value. The same is true if nodes $O_{z,x}$ and $O_{z,y}$ exist.
2. On the other hand, if nodes $O_{x,z}$ and $O_{z,y}$ exist, we expect these two nodes to reach the opposite decision. In this case, we add a factor that encourages one node to take the value $\boxed{>}$ if the other prefers the value $\boxed{<}$, and vice versa. (For the case of $\boxed{\simeq}$, if one prefers that value, then both should.)

2.3.4 *Cross-Knowledge Correlation*

Some knowledge dimensions, such as size and weight, have a significant correlation in their implied relations. For two such attributes a and b, if the same frame $F^a_{v_i}$ and $F^b_{v_i}$ exists in both graphs, we add a factor ψ_a between them to push them towards taking the same value.

2.3.5 *Seed Knowledge*

In order to kick off learning, we provide a small set of seed knowledge among the random variables in $\{\mathcal{O}, \mathcal{F}\}$ with seed factors ψ_{seed}. These unary seed factors push the belief for its associated random variable strongly towards the seed label.

2.3.6 *Potential Functions*

UNARY FACTORS For all frame and object pair random variables in the training set, we train a maximum entropy classifier to predict the value of the variable. We then use the probabilities of the classifier as potentials for seed factors given to all random variables in their class (frame or object pair). Each log-linear classifier is trained separately per attribute on a featurized vector of the variable:

$$\mathbb{P}(r|X^a) \propto e^{w_a \cdot f(X^a)}$$

The feature function is defined differently according to the node type:

$$\begin{aligned} f(O^a_{p,q}) &:= \langle g(p), g(q) \rangle \\ f(F^a_{v_t}) &:= \langle h(t), g(v), g(t) \rangle \end{aligned}$$

Here $g(x)$ is the GloVe word embedding (Pennington, Socher, and Manning, 2014) for the word x (t is the frame relation's preposition, and $g(t)$ is simply set to the zero vector if there is no preposition) and $h(t)$ is a one-hot vector of the frame relation type. We use GloVe vectors of 100 dimensions for verbs and 50 dimensions for objects and prepositions (the dimensions picked based on development set).

BINARY FACTORS In the case of all other factors, we use a "soft 1" agreement matrix with strong signal down the diagonals:

$$\begin{bmatrix} & > & \simeq & < \\ > & \mathbf{0.7} & 0.1 & 0.2 \\ \simeq & 0.15 & \mathbf{0.7} & 0.15 \\ < & 0.2 & 0.1 & \mathbf{0.7} \end{bmatrix}$$

2.3.7 *Inference*

After our full graph is constructed, we use belief propagation to infer the assignments of frames and object pairs not in our training data. Each message μ is a vector where each element is the probability that a random variable takes on each value $x \in \{\boxed{>}, \boxed{<}, \boxed{\simeq}\}$. A message passed from a random variable v to a neighboring factor f about the value x is the product of the messages from its other neighboring factors about x:

$$\mu_{v \to f}(x) \propto \prod_{f' \in N(v) \setminus \{f\}} \mu_{f' \to v}(x)$$

A message passed from a factor f with potential ψ to a random variable v about its value x is a marginalized belief about v taking value x from the other neighboring random variables combined with its potential:

$$\mu_{f \to v}(x) \propto \sum_{\mathbf{x} : \mathbf{x}[v]=x} \psi(\mathbf{x}) \prod_{v' \in N(f) \setminus \{v\}} \mu_{v' \to f}(\mathbf{x}[v'])$$

After stopping belief propagation, the marginals for a node can be computed and used as a decision for that random variable. The marginal for v taking value x is the product of its surrounding factors' messages:

$$v(x) \propto \prod_{f \in N(v)} \mu_{f \to v}(x)$$

2.4 EXPERIMENTAL RESULTS

FACTOR GRAPH CONSTRUCTION We first need to pick a set of frames and objects to determine our set of random variables. The frames are simply the subset of the frames that were crowdsourced in the given configuration (e.g., seed + dev), with "soft 1" unary seed factors (the gold label indexed row of the binary factor matrix) given only to those in the seed set. The same selection criteria and seed factors are applied to the crowdsourced object pairs.

For lexical similarity factors (ψ_v, ψ_o), we pick connections based on the cosine similarity scores of GloVe vectors thresholded above a value chosen based on development set performance. Attribute similarity factors (ψ_a) are chosen based on sets of frames that reach largely the same decisions on the seed data (95%). Frame similarity factors (ψ_f) are added to pairs of frames with linguistically similar constructions. Finally, selectional preference factors (ψ_s) are picked by using a threshold (also tuned on the development set) of pointwise mutual information (PMI) between the frames and the object pairs' occurrences in the Google Syntax Ngram corpus.

For each task, we consider the set of factors to include in each model a hyperparameter, which is also tuned on the development set.

BASELINES Baselines include making a RANDOM choice, picking between $\boxed{>}$, $\boxed{<}$, and $\boxed{\approx}$), picking the MAJORITY label, and a maximum entropy classifier based on the embedding representations (EMB-MAXENT) defined in Section 2.3.6.

INFERRING KNOWLEDGE OF ACTIONS Our first experiment is to predict knowledge implied by new frames. In this task, 5% of the frames are available as seed knowledge. We experiment with two different sets of seed knowledge for the object pair data: OUR MODEL (A) uses only 5% of the object pair data as seed, and OUR MODEL (B) uses 20%.

The full results for the baseline methods and our model are given in the upper half of Table 2.2. Our model outperforms the baselines on all attributes except for the speed, which has a highly skewed label distribution to allow the majority baseline to perform

Ex	Frame gloss	Attr	Score
1	____ opened ____	*size*	
2	PERSON set ____ upon ____	*wgt*	
3	____ stood on ____	*str*	
4	PERSON arrived on ____	*rgd*	
5	____ put up ____	*spd*	
6	PERSON drove ____ for ____	*size*	
7	PERSON stopped ____ with ____	*wgt*	
8	____ lived at ____	*str*	
9	____ snipped off ____	*rgd*	
10	____ caught ____	*spd*	

Figure 2.4: Example model predictions on dev set frames. The model's confidence is shown by the bars on the right. The correct relation is highlighted in orange (6–10 are failure cases for the model). If there are two blanks, the relation is between them. If there is only one blank, the relation is between PERSON and the blank. Note that = receives minuscule weight because it is never the correct value for frames in the seed set.

well. Sample correct predictions from the development set are shown in examples 1–5 of Figure 2.4.

INFERRING KNOWLEDGE OF OBJECTS Our second experiment is to predict the correct relations of new object pairs. The data for this task is the inverse of before: 5% of the object pairs are available as seed knowledge, and we experiment with both 5% (OUR MODEL (A)) and 20% (OUR MODEL (B)) frames given as seed data. Again, both are independently tuned on the development data. Results for this task are presented in the lower half of Table 2.2. While OUR MODEL (A) is competitive with the strongest baseline, introducing the additional frame data allows OUR MODEL (B) to reach the highest accuracy.

2.5 DISCUSSION

METAPHORICAL LANGUAGE While our frame patterns are intended to capture action verbs, our templates also match senses of those verbs that can be used with abstract or metaphorical arguments, rather than directly physical ones. One example from the development set is "x contained y." While x and y can be real objects, more abstract senses of "contained" could involve y as a "forest fire" or even a "revolution." In these instances, $x >^{\text{size}} y$ is plausible as an abstract notion: if some entity can contain a revolution, we might think that entity as "larger" or "stronger" than the revolution.

ERROR ANALYSIS Examples 6–10 in Figure 2.4 highlight failure cases for the model. Example 6 shows a case where the comparison is nonsensical because *"for"* would

naturally be followed by a purpose (*"He drove the car for work."*) or a duration (*"She drove the car for hours."*) rather than a concrete object whose size is measurable. Example 7 highlights an underspecified frame. One crowd worker provided the example, *"PERSON stopped the fly with {the jar / a swatter},"* where fly $<^{\text{weight}}$ {jar, swatter}. However, two crowd workers provided examples like *"PERSON stopped their car with the brake,"* where clearly car $>^{\text{weight}}$ brake. This example illustrates complex underlying physics we do not model: a brake—the pedal itself—is used to stop a car, but it does so by applying significant force through a separate system.

The next two examples are cases where the model nearly predicts correctly (8, e.g., *"She lived at the office."*) and is just clearly wrong (9, e.g., *"He snipped off a locket of hair"*). Example 10 demonstrates a case of polysemy where the model picks the wrong side. In the phrase, *"She caught the runner in first,"*, it is correct that she $>^{\text{speed}}$ runner. However, the sense chosen by the crowd workers is that of, *"She caught the baseball,"* where indeed she $<^{\text{speed}}$ baseball.

2.6 RELATED WORK

Several works straddle the gap between IE, knowledge base completion, and learning commonsense knowledge from text. Earlier works in these areas use large amounts of text to try to extract general statements like "A THING CAN BE READABLE" (Gordon, Van Durme, and Schubert, 2010) and frequencies of events (Gordon and Schubert, 2012). Our work focuses on specific domains of knowledge rather than general statements or occurrence statistics, and develops a frame-centric approach to circumvent reporting bias. Other work uses a knowledge base and scores unseen tuples based on similarity to existing ones (Angeli and Manning, 2013; Li et al., 2016), or extends it by inferring new facts from unstructured text using natural language inference (Angeli and Manning, 2014). Zhang et al. (2017) predict the likelihood of entailed commonsense statements extracted from a large text corpus. In contrast to the above, our work seeks to induce several novel types of graded physical knowledge which lack existing databases.

A number of recent works combine multimodal input to learn visual attributes (Bruni et al., 2012; Silberer, Ferrari, and Lapata, 2013), extract commonsense knowledge from web images (Tandon et al., 2016), and overcome reporting bias (Misra et al., 2016). In contrast, we focus on natural language evidence to reason about attributes that are both in (size) and out (weight, rigidness, etc.) of the scope of computer vision. Yet other works mine numerical attributes of objects (Davidov and Rappoport, 2010; Narisawa et al., 2013; Takamura and Tsujii, 2015) and comparative knowledge from the web (Tandon, De Melo, and Weikum, 2014). Our work uniquely learns verb-centric lexical entailment knowledge.

A handful of works have attempted to learn the types of knowledge we address in this work. One recent work tried to directly predict several binary attributes (such "is large" and "is yellow") from on off-the-shelf word embeddings, noting that accuracy was very low (Rubinstein et al., 2015). Another line of work addressed grounding verbs in the context of robotic tasks. One paper in this line acquires verb meanings by observing state

changes in the environment (She and Chai, 2016). Another work in this line does a deep investigation of eleven verbs, modeling their physical effect via annotated images along eighteen attributes (Gao et al., 2016). These works are encouraging investigations into multimodal groundings of a small set of verbs. Our work instead grounds into a fixed set of attributes but leverages language on a broader scale to learn about more verbs in more diverse set of frames. In this, our work can be seen as exploring predicate lexical semantics in the vein of semantic proto-roles (Dowty, 1991; Kako, 2006; Reisinger et al., 2015), but instead affording pairwise, physical relations between roles.

2.7 CONCLUSION

VERB PHYSICS presents a novel take on verb-centric frame semantics to learn implied physical knowledge latent in verbs. Empirical results confirm that by modeling changes in physical attributes entailed by verbs together with objects that exhibit these properties, we are able to better infer new knowledge in both domains.

3

SOCIAL CHEMISTRY

Understanding and reasoning about social situations relies on unspoken commonsense rules about *social norms*, i.e., acceptable social behavior (Haidt, 2012). For example, when faced with situations like "*wanting to call the cops on my neighbors*," (Figure 3.1), we perform a rich variety of reasoning about about legality, cultural pressure, and even morality of the situation (here, "*reporting a crime*" and "*being friends with your neighbor*" are conflicting norms). Failure to account for social norms could significantly hinder AI systems' ability to interact with humans (Pereira, Prada, and Santos, 2016).

In this chapter, we introduce Social Chemistry as a new formalism to study people's social and moral norms over everyday real life situations. Our approach based on crowdsourced descriptions of norms is inspired in part by studies in *descriptive* or *applied* ethics (Hare et al., 1981; Kohlberg, 1976), which takes a *bottom-up* approach by asking people's judgements on various ethical situations. This is in contrast to the *top-down* approach taken by *normative* or *prescriptive* ethics to prescribe the key elements of ethical judgements. The underlying motivation of our study is that we, the NLP field, might have a real chance to contribute to the studies of computational social norms and descriptive ethics through large-scale crowdsourced annotation efforts combined with state-of-the-art neural language models.

To that end, we organize *descriptive* norms via free-text *rules-of-thumb (RoTs)* as the basic conceptual units.

> **Rule-of-Thumb (RoT)** — A descriptive cultural norm structured as the judgment of an action. For example, *"It's rude to run the blender at 5am."*

Each RoT is further broken down with 12 theoretically-motivated dimensions of people's judgments such as social judgments of good and bad, theoretical categories of moral foundations, expected cultural pressure, and assumed legality. All together, these annotations comprise Social-Chem-101, a new type of NLP resource that catalogs 292k RoTs over 104k real life situations, along with 365k sets of structural annotations, which break each RoT into 12 dimensions of norm attributes. Together, this amounts to over 4.5M categorical and free-text annotations.

We investigate how state-of-the-art neural language models can learn and generalize out of Social-Chem-101 to accurately reason about social norms with respect to a previously unseen situation. We term this modeling framework Neural Norm Transformer, and find it is able to generate relevant (and potentially novel) rules-of-thumb conditioned on all attribute dimensions. Even so, this breadth of this task proves challenging to current neural models, with humans rating model's adherence to different attributes from 0.28 to 0.91 micro-F1.

In addition, we showcase a potential practical use case of computational social norms by analyzing political news headlines through the lens of our framework. We find

Figure 3.1: This figure illustrates an intuitive subset of our formalism to reason about social norms in language. Our approach centers around Rules-of-Thumb (RoTs; text in colored tubes), which describe social expectations given a situation (text in the center hexagon). Rather than prescribing what is right or wrong, RoTs reveal ethical judgments about social propriety from varying perspectives. Note that the social identities of the participants of situations would further inform which social norms are most relevant.

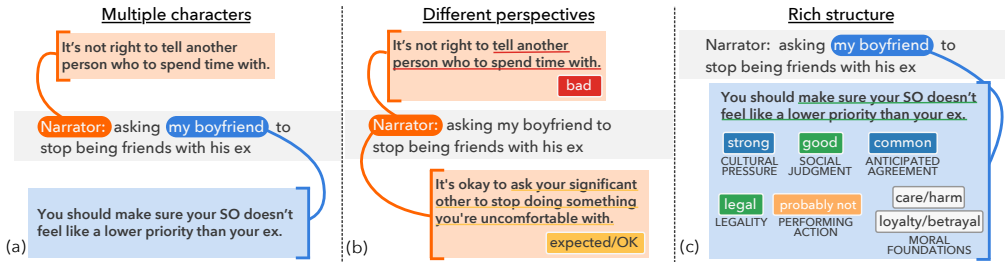


Figure 3.2: Three different slices of a complete annotation for a single situation, meant to illustrate our approach. Each **RoT** (text in colored boxes, e.g., "It's not right to tell...") is written for a particular real life **situation** (text in pale grey boxes, e.g., "asking my boyfriend to stop being ...") and a specific **person** in that situation ("narrator" vs "my boyfriend"). *(a)* A situation often includes multiple people with distinct perspectives, evoking different (and possibly conflicting) RoTs. *(b)* Even a single person may have multiple, conflicting RoTs—key ingredients for moral dilemmas. *(c)* Each RoT is further broken down with categorical and free text annotations (shown in tiny colored buttons. e.g., "strong" for *cultural pressure*). The full definition of the low-level RoT attributes are in Figure 3.4.

that our empirical results align with the *Moral Foundation Theory* of Graham, Haidt, and Nosek (2009); Haidt (2012) on how the moral norms of different communities vary depending on their political leanings and news reliability. Our empirical studies demonstrate that computational modeling of social norms is a feasible and promising research direction that warrants further investigation. SOCIAL-CHEM-101 provides a new resource to teach AI models to learn people's norms, as well as to support novel interdisciplinary research across NLP, computational norms, and descriptive ethics.

3.1 APPROACH

The study of social norms have roots in descriptive ethics and moral psychology. They tell us that social norms are culturally-sensitive standards of appropriate conduct. Alongside explicit laws and regulations that govern our society, social norms perform the role of providing guidelines on socially appropriate behaviors (Bowdery, 1941; Elster, 2006; Kohlberg, 1976) and are responsible for setting implicit expectations of what is socially right or wrong (Haidt, 2012; Hare et al., 1981; Malle, Guglielmo, and Monroe, 2014). They influence a wide-range of social functions such as preserving biological needs to survival (e.g., refraining from harming or killing), maintaining social civility and order (e.g., maintaining politeness, recognizing personal space), and providing identity and belonging to a community (e.g., respecting the elderly). In turn, these social norms influence how we judge, communicate, and interact with each other.

ROTS Our aim is then to forefront these implicit expectations about social norms via RoTs. We formalize the definition of RoTs as situationally-relevant evaluative judgments of social norm, and posit that for any given **situation**, one or more RoTs will be evoked in the minds of the interpreter. Consider the following situation and its RoT.

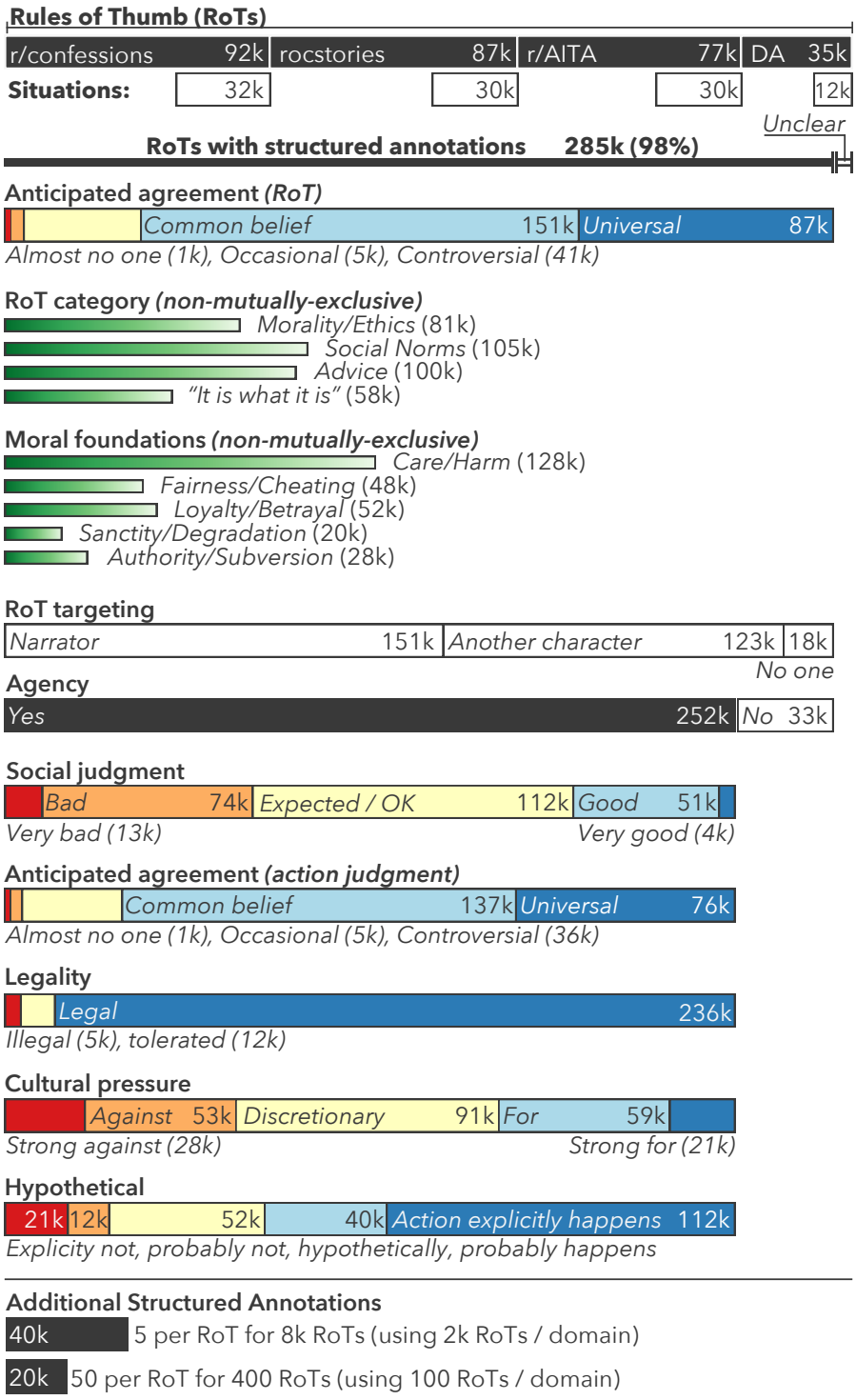


Figure 3.3: SOCIAL-CHEM-101 Dataset statistics. Bars are drawn to scale. Individual values for all of the different attributes are also given in Figure 3.4.

Punching someone.
RoT 1: It is unacceptable to injure a person.
RoT 2: People should not steal from others.
RoT 3: It is bad to betray a friend.
RoT 4: It is OK to want to take revenge.

Most readers can instantly recognize the situation is in violation of an unspoken social rule: *"Do not injure others."* This rule is responsible for the series of natural questions that probe at the morality of the action, like "why did the narrator punch someone?" "was the action justified?" and "do I want to sympathize with the narrator?" The role of the RoT is to identify the unspoken rule in the situation by specifying the behavior or **action** ("injuring a person") and its **acceptability judgment** ("it is unacceptable"). More complex situations can be associated with multiple RoTs, as seen in the example below:

Punching a friend who stole from me.
RoT 1: It is unacceptable to injure a person.
RoT 2: People should not steal from others.
RoT 3: It is bad to betray a friend.
RoT 4: It is OK to want to take revenge.

The RoTs represent a variety of social norms that elaborate on various perspectives available in the situation: RoTs about stealing (RoT 1) vs. punching (RoT 2), RoTs targeting the different characters in the situation (RoTs 1, 4 target the narrator; RoTs 2, 3 target narrator's friend), and RoTs that elaborate on additional social interpretation implicit in the situation (RoT 3: theft from a friend is cast as an act of betrayal). Effectively, RoTs represent evaluative judgments about a social situation in light of unspoken but accepted social norms.[1] Figure 3.2 shows three subsets of a situation's annotation to illustrate the perspectives RoTs capture.

CULTURAL SCOPE OF THIS STUDY We recognize that social norms are often culturally sensitive (Haidt, Koller, and Dias, 1993; Kagan, 1984) and judgments of morality and ethics concerning individuality, community and society do not always hold universally (Shweder, 1990). While some situations (e.g., "punching someone") might have similar levels of acceptability across a number of cultures, others might have drastically varied levels depending on the culture of its participants (e.g., "kissing someone on the cheek as a greeting"). As a starting point, our study focuses on the socio-normative judgments of English-speaking cultures represented within North America. While we find some variation of judgments in our annotations (e.g., with respect to certain worker characteristics, see §a.1.15), extending this formalism to other countries and non-English speaking cultures remains a compelling area of future research.

1 Our definition of RoTs corresponds to the first of the two evaluative moral judgments defined in Malle, Guglielmo, and Monroe (2014).

3.2 SOCIAL-CHEM-101 DATASET

We obtained 104k source situations from 4 text domains (§3.2.1), for which we elicited 292k RoTs from crowd workers (§3.2.2). We then define a structured annotation task where workers isolate the central action described by the RoT and provide a series of judgments about the RoT and the action (§3.2.3). In total, we collect 365k structured annotations, performing multiple annotations per RoT for a subset of the RoTs to study the variance in annotations. Figure 3.3 illustrates our dataset statistics.

3.2.1 *Situations*

We use a *situation* to denote the one-sentence prompt given to a worker as the basis for writing RoTs. We gather a total of 104k real life situations from four domains: scraped titles of posts in the subreddits `r/confessions` (32k) and `r/amitheasshole` (`r/AITA`, 30k), which largely focus on moral quandaries and interpersonal conflicts; 30k sentences from the ROCStories corpus (`rocstories`, Mostafazadeh et al., 2016); and scraped titles from the Dear Abby advice column archives[2] (`dearabby`, 12k). [3]

3.2.2 *Rules-of-Thumb (RoTs)*

To collect RoTs, we provide workers with a situation as a prompt and them to write 1 – 5 RoTs inspired by that situation. From the 104k situations, we elicit a total of 292k RoTs. Despite RoTs averaging just 10 words, we observe that 260k/292k RoTs are unique across the dataset.

For the development of RoTs, we instruct the workers to produce RoTs that *explain the basics of social norms*, just as one would instruct a five-year-old child on the ABCs of acceptable conduct. RoTs are to be:

1. **inspired by the situation**, to maintain a lower bound on relevance;
2. **self-contained**, to be understandable without additional explanation; and
3. structured as **judgment** of acceptability (e.g., good/bad, (un)acceptable, okay) and an **action** that is assessed.

In order to encourage RoT diversity, we also ask that an RoT should counterbalance *vagueness* against *specificity* so that RoTs generalize across multiple situations (e.g., *"It is rude be selfish."*) without being too specific (e.g., *"It is rude not to share your mac'n'cheese with your younger brother."*). We also ask workers to write RoTs illustrating *distinct ideas* and *avoid trivial inversions* to prevent low-information RoTs that rephrase the same idea or simply invert the judgement and action.

2 https://www.uexpress.com/dearabby/archives
3 See Appendix a.1.1 for further data preprocessing details.

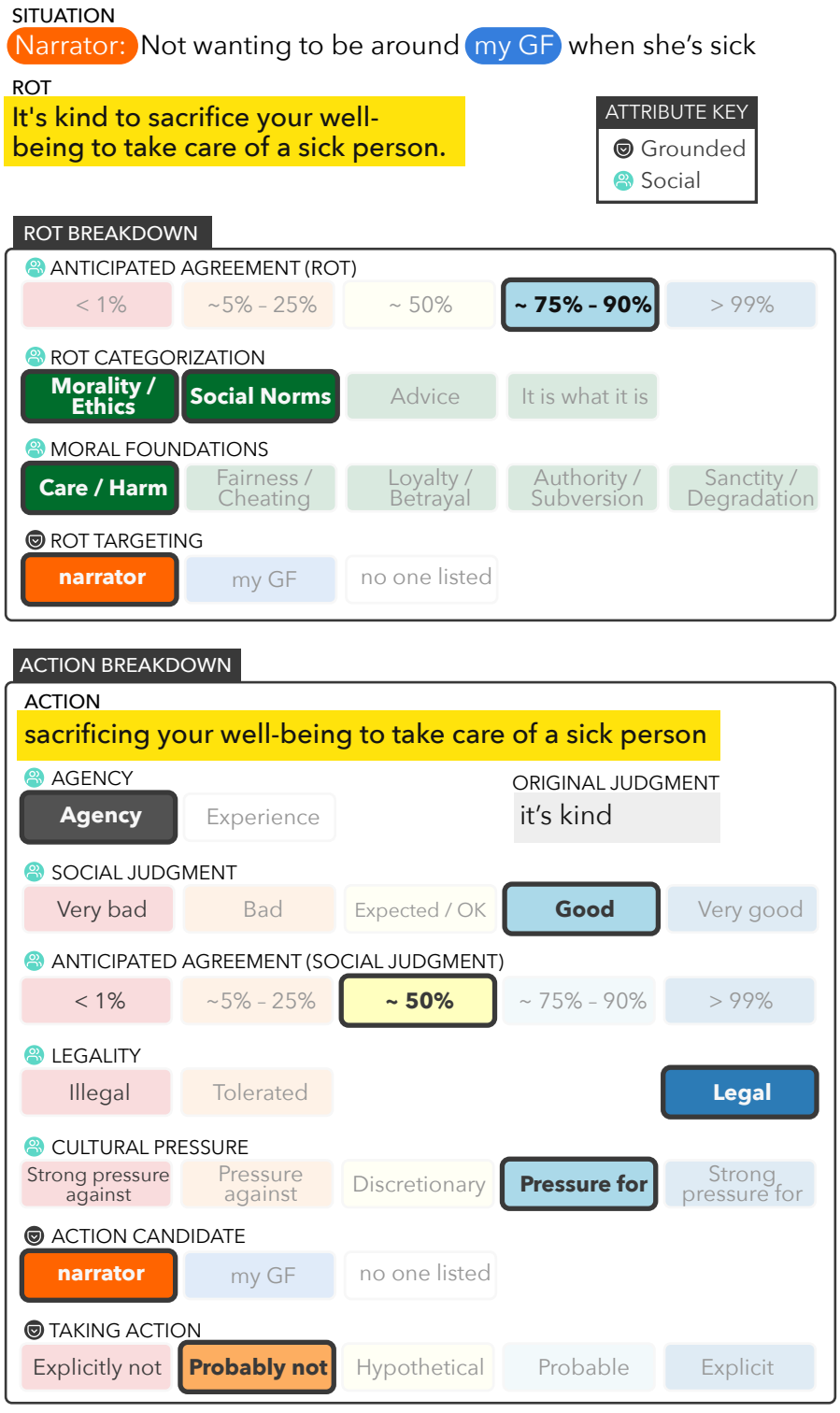


Figure 3.4: All attribute values for structured RoT annotations, with one complete example annotation filled in.

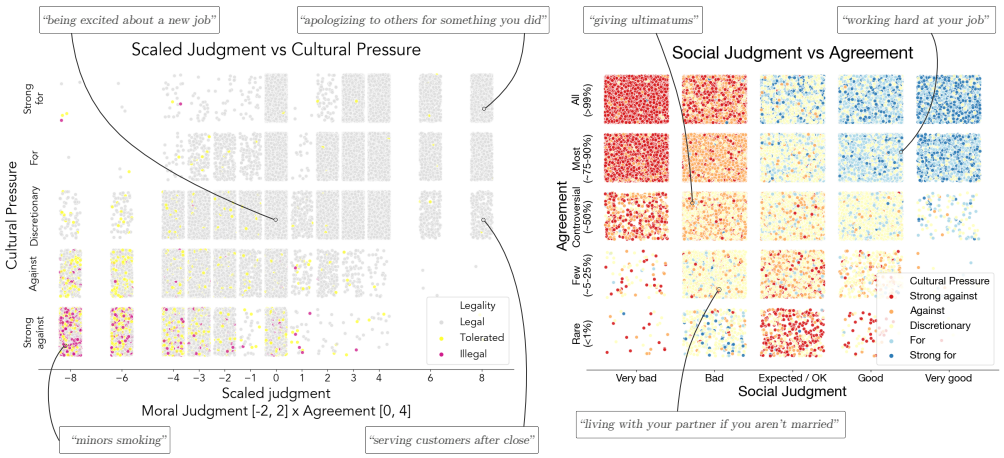


Figure 3.5: Plotting the distribution of RoTs in SOCIAL-CHEM-101 along axes of *moral judgment, agreement, cultural pressure,* and *legality.* **Left:** Moral judgment is scaled with agreement (how commonly held the belief is) and plotted against cultural pressure. Illegal activities fall in the bottom left: actions that are universally understood to be wrong and people feel negative cultural pressure for. **Right:** Moral judgment is plotted against agreement. Discretionary actions span a range of moral values (yellow ranging horizontally) and fringe beliefs often evoke strong negative cultural pressure even when morally neutral (bottom of plot).

CHARACTER IDENTIFICATION. We ask workers to identify phrases in each situation that refer to people. For example, in a situation, like *"My brother chased after the Uber driver,"* workers mark the underlined spans. We collect three workers' spans, calling each span a *character*. All characters identified become candidates for grounding RoTs and actions in the structured annotation. As such, we optimize for recall instead of precision by using the largest set of characters identified by any worker. We also include a *narrator* character by default.

3.2.3 *RoT Breakdowns*

We perform a structured annotation, which we term a *breakdown*, on each RoT. In an RoT breakdown, a worker isolates the underlying action contained in the RoT. Then, they assign a series of categorical attributes to both the RoT and the action. These categorical annotations allow for additional analyses and experiments relative to the text-only RoTs.

The attributes fall into two categories corresponding to the central annotation goals. The first goal is to tightly *ground* RoTs to their respective situations. The second goal is to partition *social* expectations using theoretically motivated categories.

A subset of the attributes are labeled on the RoT (e.g., *"It is expected that you report a crime"*), while others are on the action (e.g., *"reporting a crime"*). Figure 3.4 provides the complete set of labels available for an RoT breakdown.[4]

4 Workers are given the choice to mark the RoT as confusing, vague, or low quality, and move on (2% of RoTs).

GROUNDING ATTRIBUTES We call three attributes *grounding attributes*. Their goal is to ground the RoT and action to the situation and characters. At the RoT-level, workers mark which character should heed the RoT with the **RoT Targeting** attribute. At the action level, workers first pick the **action's best candidate** character, for whom the action is most relevant. However, since RoTs can identify actions that are both explicit and hypothetical in the situation, we additionally annotate whether the candidate character is explicitly **taking the action** in the situation.

SOCIAL ATTRIBUTES The second set of attributes characterize social expectations in an RoT. The first two social attributes both label **anticipated agreement**. For an RoT, this attribute asks how many people probably *agree* with the RoT as stated. At the action level, it asks what portion of people probably agree with the *judgment* given the *action*.

Four social attributes relate to the theoretical underpinnings of this work in §3.1. An RoT-level attribute is the set of **Moral Foundations**, based on a well-known social psychology theory that outlines culturally innate moral reasoning (Haidt, 2012). The action-level attributes **legality** and **cultural pressure** are designed to reflect the two-coarse-grained categories proposed by the Social Norms Theory (Kitts and Chiang, 2008; Perkins and Berkowitz, 1986). Legality corresponds to prescriptive norms: what one ought to do. Cultural pressure corresponds to descriptive norms: what one is socially influenced to do. Finally, the **social judgment** aims to capture subjective moral judgment. A base judgment of what is good or bad is thought to intrinsically motivate social norms (Haidt, Koller, and Dias, 1993; Malle, Guglielmo, and Monroe, 2014).

The final two attributes provide a coarse categorization over RoTs and actions. The **RoT Category** attribute estimate distinctions between morality, social norms, and other kinds of advice. This aims to separate moral directives from tips or general world knowledge (e.g., "It is good to eat when you are hungry"). The attribute **agency** is designed to let workers distinguish RoTs that involve agentive action from those that indicate an an experience (e.g., "It is sad to lose a family member").

3.2.4 *Analysis*

We briefly highlight three key aspects of our formalism: social judgment, anticipated agreement, and cultural pressure. Figure 3.5 shows two plots partitioning RoTs based on these three attributes (with legality also highlighted in the left plot (a)).

In the left plot (Figure 3.5 (a)), the x-axis contains a new quantity, where social judgment ($\in [-2, 2]$) is multiplied by agreement ($\in [0, 4]$) to scale it.[5] The result is that x values range from universally-agreed bad actions (-8) to universally-agreed good actions (+8). Intuitively, the bottom-left group shows illegal actions, which are both "bad" (left x) and people feel strong pressure not to do (bottom y). The data are generally

5 Strict statisticians will note that plotting ordinal values numerically is an abuse of notation, much less scaling two values together. We present these graphs for illustrative purposes to observe the stratification of our dataset, not to make quantitative claims.

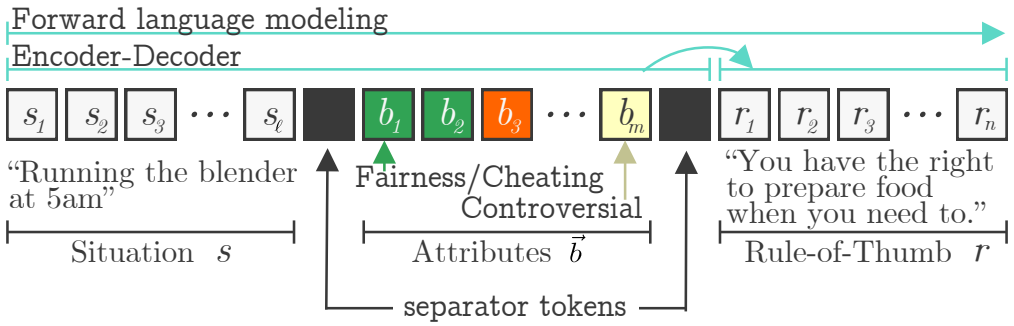


Figure 3.6: Illustration of modeling setup for the objective $p(r|s, \vec{b}_r)$.

distributed in a line towards the top right, which are "good" (right x) actions that people feel strong pressure to do (top y).

However, the spread of the data in Figure 3.5 (a) illustrates the difference between morality and cultural pressure. There are a range of morally charged actions, but for which people don't feel cultural pressure (the horizontal range in x values across the central y = *Discretionary*). Conversely, we observe actions that are morally neutral, but for which people do feel cultural pressure (the vertical range in y values along the middle $x = 0$).

The right plot, Figure 3.5 (b), shows social judgment against agreement, colored by cultural pressure. At high levels of agreement (top of graph), cultural pressure (color) follows social judgment (horizontal changes in x values). However, for controversially-held judgments (lower y values), we see a range of cultural pressure. This includes morally good or bad actions that are still discretionary (middle y values), as well as morally neutral actions for which people feel strong cultural pressure (lower y values).

These plots illustrate two ways of stratifying actions along socially relevant dimensions. We anticipate considerable further dataset exploration remains.

3.3 MODEL

We investigate neural models based on pre-trained language models for learning various sub-tasks derived from SOCIAL-CHEM-101.

3.3.1 *Training Objectives*

Our main modeling formulation is straightforward. Given a situation (s), we wish to model the conditional distribution of RoTs (r), actions (a), and set of attributes from the breakdown ($\vec{b}$). We can partition the attributes $\vec{b} = \{\vec{b}_r, \vec{b}_a\}$ into disjoint sets relevant to the RoT and action, and write

$$p(r, a, \vec{b}|s) = \underbrace{p(a, \vec{b}_a|r, \vec{b}_r, s)}_{\textit{action transcription}} \times \underbrace{p(r, \vec{b}_r|s)}_{\textit{RoT prediction}} . \tag{3.1}$$

Objective		
RoT	**Action**	**Interpretation**
$p(r\|s)$	$p(a\|s)$	Text-only generation
$p(\vec{b}_r\|s)$	$p(\vec{b}_a\|s)$	Attribute prediction
$p(r\|s, \vec{b}_r)$	$p(a\|s, \vec{b}_a)$	Controlled generation
$p(\vec{b}_r\|s, r)$	$p(\vec{b}_a\|s, a)$	Attribute labeling
$p(r, \vec{b}_r\|s)$	$p(a, \vec{b}_a\|s)$	Model choice generation

Table 3.1: Generative model objectives corresponding to the training setups we consider. Each model (RoT or action) is trained on all objectives simultaneously.

Equation 3.1 allows us to model all components of interest given a situation s. However, the *action transcription* term is quite strongly conditioned, because actions are so closely related to their RoTs. In this chapter, we instead focus our study of actions on a more difficult distribution that conditions only on the situation:

$$\underbrace{p(a, \vec{b}_a|r, \vec{b}_r, s)}_{\textit{action transcription}} \overset{\textit{omit RoT}}{\longrightarrow} \underbrace{p(a, \vec{b}_a|s)}_{\textit{action prediction}} . \tag{3.2}$$

We model both the *RoT prediction* (Eq. 3.1) and *action prediction* (Eq. 3.2) distributions with conditional forward language modeling. We tokenize all quantities $(s, r, a, \vec{b})$, creating unique tokens for each attribute value b_i, and concatenate them together in a canonical order to form strings $p(x_{\text{out}}|x_{\text{in}})$. We then train to maximize the standard language modeling objective:

$$x = [x_{\text{in}}; x_{\text{out}}], \quad p(x) = \prod_{i=1}^{n} p(x_i|x_{<i}). \tag{3.3}$$

Both the *RoT prediction* (Eq. 3.1) and *action prediction* (Eq. 3.2) distributions have similar forms $p(y, \vec{b}_y|s)$ for $y \in \{r, a\}$. We take advantage of this symmetry to study variations of both distributions. Inspired by recent work (Zellers et al., 2019), we construct permutations of our data that omit different fields while maintaining the canonical order. Table 3.1 shows the setups that we consider, and Figure 3.6 illustrates an example objective.

We train each model (either RoT or action) on all relevant objectives in Table 3.1 (i.e., one of the columns). Intuitively, this allows the model to condition on and generate a range of fields.[6] We can do this by simply treating each objective as defining a subset of the fields, as well as their ordering, for each data point. Then, we combine and shuffle all objectives' views of the data.

6 It is possible to remove the assumption that the situation is provided, which would allow the model to generate s as well. We leave such experiments for future work.

	→ *RoT*				→ *Action*							
	Category	**Moral F.**	**Agree**	***Relev.***	**Agency**	**Judgment**	**Agree**	**Pressure**	**Legal**	**Taking**	***Relev.***	
Random RoT	0.73	0.84	0.48	1.25	0.90	0.57	**0.55**	0.53	0.80	0.04	1.22	Model choice $p(y, \vec{b}_y \mid s)$
BERT-Score (Z et al., 2020)	**0.76**	0.83	0.48	2.00	0.90	**0.64**	0.46	**0.61**	0.81	0.20	2.00	
GPT (R et al., 2018)	0.71	0.77	0.39	2.23	0.82	0.40	0.36	0.32	0.76	0.15	2.25	
BART (L et al., 2019)	0.69	0.79	**0.49**	2.60	**0.91**	0.55	0.54	0.46	0.80	0.18	2.52	
T5 (R et al., 2019)	0.62	**0.85**	0.42	**2.78**	0.78	0.36	0.36	0.23	0.56	0.23	**2.73**	
GPT-2 Small (R et al., 2019)	0.62	0.79	0.34	2.03	0.82	0.34	0.34	0.27	0.79	0.09	1.99	
GPT-2 XL - No pre-train	0.68	0.78	0.20	1.37	0.81	0.37	0.30	0.33	0.79	0.06	1.29	
GPT-2 XL	0.75	0.84	0.42	2.53	**0.91**	0.51	0.36	0.45	**0.82**	**0.32**	2.60	
Random RoT	0.59	0.75	**0.41**	1.20	0.84	0.27	0.28	0.21	0.74	0.01	1.19	Controlled $p(y \mid s, \vec{b}_y)$
BERT-Score (Z et al., 2020)	0.66	0.78	**0.41**	2.00	0.87	0.40	**0.45**	0.34	**0.76**	0.16	1.97	
GPT (R et al., 2018)	0.64	0.79	0.36	2.21	0.83	0.46	0.36	0.38	0.74	0.17	2.26	
BART (L et al, 2019)	0.70	**0.81**	0.38	2.60	0.84	0.47	0.42	0.41	0.73	0.20	2.44	
T5 (R et al., 2019)	0.66	0.80	0.40	**2.77**	0.83	0.41	0.34	0.38	0.73	0.24	**2.79**	
GPT-2 Small (R et al., 2019)	0.64	0.78	0.30	2.10	0.78	0.38	0.30	0.27	0.71	0.10	1.97	
GPT-2 XL - No pre-train	0.67	0.79	0.23	1.35	0.83	0.36	0.32	0.26	0.73	0.04	1.33	
GPT-2 XL	**0.71**	0.79	0.38	2.65	**0.90**	**0.51**	0.38	**0.42**	0.74	**0.28**	2.54	

Table 3.2: Human evaluation results for conditionally generating RoTs and actions, either letting the models choose the attributes (top half), or providing the attributes as input constraints (bottom half). All columns are micro-F1 scores (0–1), except *Relevance* (1–3). **Takeaway:** While state-of-the-art models are able to generate relevant RoTs and actions that generally follow constraints (moderately high scores in some columns), correctly conditioning on a complete set of attributes remains challenging (several columns show poor model performance in bottom half).

3.4 ARCHITECTURES

We present results for the GPT and GPT-2 architectures (Radford et al., 2018b; 2019a), as well as two encoder-decoder language models (BART and T5, Lewis et al., 2019; Raffel et al., 2019). We train forward language models with loss over the entire sequence x, whereas encoder-decoder models only compute loss for the output sequence x_{out}. Collectively, we term these architectures trained on our objectives the NEURAL NORM TRANSFORMER.

3.5 EXPERIMENTS AND RESULTS

3.5.1 *Tasks*

While we train each model on all (RoT or action) objectives at once, we pick two particular objectives to asses the models. The first is $p(y, \vec{b}_y|s)$ — *"model choice."* In this setting, each model is allowed to pick the most likely attributes $\vec{b}_y$ given a situation s, and generate an RoT (or action) y that adheres to those attributes. This setup should be easier because a model is allowed to pick the conditions of its own generation ($\vec{b}_y$).

The second setting is $p(y|s, \vec{b}_y)$ — *"conditional."* We provide models with a set of attributes $\vec{b}_y$ that they must follow when generating an RoT (or action) y. This presents a more challenging setup, because models cannot simply condition on the set of attributes that they find most likely. We select sets of attributes $\vec{b}_y$ provided by the human

annotators for the situation s to ensure models are not tasked with generating from impossible constraints.

SETUP We split our dataset into 80/10/10% train/dev/test partitions by situation, such that each domain's situations are proportionally distributed. This guarantees previously unobserved dev and test situations. For all models we use top-p decoding with $p = 0.9$ (Holtzman et al., 2020a).

BASELINES We use a *Random RoT* baseline to verify the dataset diversity (selections should have low relevance to test situations) and evaluation setup (RoTs and actions should still be internally consistent). We also use a *BERT-Score* (Zhang et al., 2020) retrieval baseline that finds the most similar training situation. If attributes $\vec{b}_y$ are provided, the retriever picks the RoT (or action) from the retrieved situation with the most similar attributes.

ABLATIONS We report two model ablations. For *-Small*, we finetune GPT-2 Small with the same general architecture. For *-No pretrain*, we randomly initialize the model's weights.[7]

3.5.2 *Results*

HUMAN EVALUATION Table 3.2 presents a human evaluation measuring how effective models are at generating RoTs and actions for both task settings. While most columns measure attribute adherence, the *Relevance* score is critical for distinguishing whether RoTs actually apply to the provided situation (e.g., see low scores for the *Random RoT* baseline). In both setups, T5's generations rank as most tightly relevant to the situation. But in terms of correctly following attributes, GPT-2 is more consistent, especially in the *controlled* task setup (lower; top scores on 5/9 attributes). However, no model is able to achieve a high score on all columns in the bottom half of the table. This indicates that fully constrained conditional generation may still present a significant challenge for current models.

AUTOMATIC EVALUATION We also provide automatic metrics of the generated outputs. We train attributes classifiers using RoBERTa (Liu et al., 2019b), and use them to classify the model outputs.[8]

Table 3.3 presents test set model performance on perplexity, BLEU (Papineni et al., 2002), and attribute micro-F1 classifier score. The automatic metrics are consistent with human evaluation. T5 is a strong generator overall, achieving the highest BLEU score and the highest *relevance* score in §3.5.2. However, GPT-2 more consistently adheres to attributes, outperforming T5 in attribute F_1 with nearly 20 points gap for RoTs, and over 10 points for actions.

7 We omit the evaluation of an "out-of-the-box GPT2-XL" baseline (i.e. no fine-tuning) whose outputs predictably do not resemble RoTs or actions.

8 BERT and BART performed worse across attributes.

Model	Ppl.	BLEU-4	Attr. μF1
→ RoT			
GPT	1.81	5.41	0.42
BART-large	1.76	6.65	0.47
T5-large	1.94	**10.79**	0.34
GPT-2 Small	1.97	4.97	0.38
GPT-2 XL - No fine-tune	-	0.46	0.20
GPT-2 XL - No pre-train	2.54	4.39	0.42
GPT-2 XL	1.75	6.53	**0.53**
→ Action			
GPT	1.80	6.75	0.60
BART-Large	1.72	8.34	0.66
T5-Large	2.00	**8.93**	0.58
GPT-2 Small	1.94	6.62	0.56
GPT-2 XL - No fine-tune	-	0.25	0.52
GPT-2 XL - No pre-train	2.51	5.43	0.55
GPT-2 XL	1.73	7.98	**0.68**

Table 3.3: Test set performance by automatic metrics, including an attribute classifier. Perplexities are not comparable between encoder-decoder models (BART and T5, loss on x_{out} only) and other models (loss on full sequence x). **Takeaway:** Automatic metrics corroborate human evaluation results: while T5 is most adept at BLEU, GPT-2 XL more consistently adheres to attributes (Attr. μF1).

		Left (-) or Right (+)	Reliability
	Agreement	-0.015**	-0.008*
ROT Cat.	Morality / Ethics	-0.069***	-0.022***
	Social Norms	0.019***	-0.006*
	It is what it is	0.039***	-0.007**
	Advice	0.031***	0.033***
Moral F.	Care / Harm	-0.033***	-0.016***
	Authority / Subversion	*n.s.*	*n.s.*
	Fairness / Cheating	-0.050***	*n.s.*
	Loyalty / Betrayal	0.026***	-0.007**
	Sanctity / Degradation	0.014**	-0.017***

Table 3.4: Correlations between generated RoT attributes for headlines and the news source's political leaning (left: neg., right: pos.) and reliability (controlled for political leaning). Results shown are significant after Holm-correction for multiple comparisons ($p <$ 0.001: ***, $p <$ 0.01: **, $p <$ 0.05: *, $p >$ 0.05: *n.s.*). **Takeaway:** We see evidence that a model trained on the SOCIAL-CHEM-101 Dataset can naturally uncover moral and topical leanings in news sources, mirroring results found in previous news studies.

3.6 MORALITY & POLITICAL BIAS

To demonstrate a use case of our proposed formalism, we analyze the social norms and expectations evoked in news headlines from news sources of various political leanings and trustworthiness, using the NEURAL NORM TRANSFORMER (GPT-2 XL). Specifically, we generate ROTs and attributes for 50,000 news headlines randomly selected from Nørregaard, Horne, and Adalı (2019), a large corpus of political headlines from 2018 paired with news source ratings of political leaning (5-point scale from left- to right-leaning) and factual reliability (5-point scale from least reliable to most reliable).[9]

Table 3.4 shows the correlations between RoT attributes and the political leaning and reliability of sources. Our results strongly corroborate findings by Graham, Haidt, and Nosek (2009), showing that liberal headlines evoke more "fairness" and "care," while right-leaning headlines evoke more "sanctity" and "loyalty." Furthermore, in line with findings by Volkova et al. (2017), more reliable news source tend to evoke more advice and less morality.

3.7 RELATED WORK

Our formalism heavily draws from works in descriptive ethics and social psychology, but is also inspired by studies in social implicatures and cooperative principles in

9 We use the MediaBias/FactCheck ratings: https://mediabiasfactcheck.com.

pragmatics (Grice, 1975; Kallia, 2004) and the theories of situationally-rooted evocation of frames (Fillmore and Baker, 2001).

Our work adds to the growing literature concerned with distilling reactions to situations (Ding and Riloff, 2016; Vu et al., 2014) as well as social and moral dynamics in language (Van Hee et al., 2015). Commonly used for coarse-grained analyses of morality in text (Fulgoni et al., 2016; Volkova et al., 2017; Weber et al., 2018), Graham, Haidt, and Nosek (2009) introduce the Moral Foundations lexicon, a dictionary of morality-evoking words (later extended by Rezapour, Shah, and Diesner, 2019).

A recent line of work focused on representing social implications of everyday situations in free-form text in a knowledge graph (Rashkin et al., 2018; Sap et al., 2019b). Relatedly, Sap et al. (2020) introduce Social Bias Frames, a hybrid free-text and categorical formalism to reason about biased implications in language. In contrast, our work formalizes a new type of reasoning around expectations of social norms evoked by situations.

Finally, concurrent works have developed rich and exciting resources studying similar phenomena. Tay et al. (2020) study *Would you rather?* questions, and Acharya, Talamadupula, and Finlayson (2020) investigate ritual understanding across cultures. Hendrycks et al. (2020) study ethical questions, attempting to assign a real-valued utility to scenarios across a range of ethical cateogires. And Lourie, Bras, and Choi (2020) define the challenge of predicting the *r/AITA* task using the full posts. In contrast to these studies, our work addresses norms by distilling cultural knowledge to a new conceptual level of Rules-of-Thumb and corresponding structural annotations.

3.8 CONCLUSION

We present SOCIAL-CHEM-101, an attempt at providing a formalism and resource around the study of grounded social, moral, and ethical norms. Our experiments demonstrate preliminary success in generative modeling of structured RoTs, and corroborate findings of moral leaning in an extrinsic task. Comprehensive modeling of social norms presents a promising challenge for NLP work in the future.

Part II

IMPLICIT COMMONSENSE

4

NEURAL NATURALIST

Humans are adept at making fine-grained comparisons, but sometimes require aid in distinguishing visually similar classes. Take, for example, a citizen science effort like iNaturalist,[1] where everyday people photograph wildlife, and the community reaches a consensus on the taxonomic label for each instance. Many species are visually similar (e.g., Figure 4.1, top), making them difficult for a casual observer to label correctly. This puts an undue strain on lieutenants of the citizen science community to curate and justify labels for a large number of instances. While everyone may be capable of making such distinctions visually, non-experts require training to know what to look for.

Field guides exist for the purpose helping people learn how to distinguish between species. Unfortunately, field guides are costly to create because writing such a guide requires expert knowledge of class-level distinctions.

In this chapter, we study the problem of explaining the differences between two images using natural language. We introduce a new dataset called *Birds-to-Words* of paragraph-length descriptions of the differences between pairs of bird photographs. We find several benefits from eliciting comparisons: (a) without a guide, annotators naturally break down the subject of the image (e.g., a bird) into pieces understood by the everyday observer (e.g., head, wings, legs); (b) by sampling comparisons from varying visual and taxonomic distances, the language exhibits naturally adaptive granularity of detail based on the distinctions required (e.g., "red body" vs "tiny stripe above its eye"); (c) in contrast to requiring comparisons between categories (e.g., comparing one species vs. another), non-experts can provide high-quality annotations without needing domain expertise.

We also propose the *Neural Naturalist* model architecture for generating comparisons given two images as input. After embedding images into a latent space with a CNN, the model combines the two image representations with a joint encoding and comparative module before passing them to a Transformer decoder. We find that introducing a comparative module—an additional Transformer encoder—over the combined latent image representations yields better generations.

Our results suggest that these classes of neural models can assist in fine-grained visual domains when humans require aid to distinguish closely related instances. Non-experts—such as amateur naturalists trying to tell apart two species—stand to benefit from comparative explanations. Our work approaches this sweet-spot of visual expertise, where any two in-domain images can be compared, and the language is detailed, adaptive to the types of differences observed, and still understandable by laypeople.

Recent work has made impressive progress on context sensitive image captioning. One direction of work uses class labels as context, with the objective of generating captions that distinguish why the image belongs to one class over others (Hendricks et

1 https://www.inaturalist.org

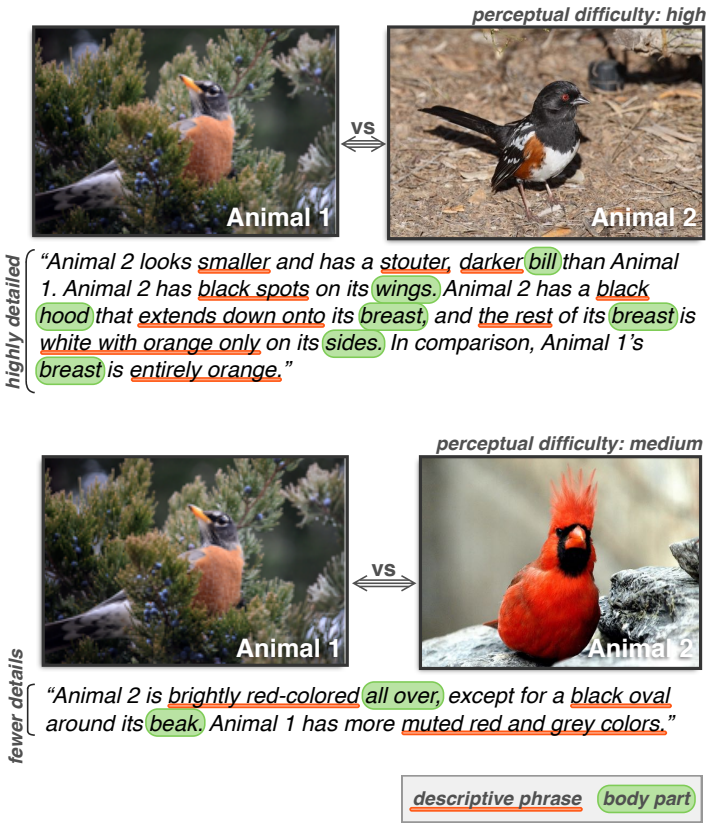


Figure 4.1: The Birds-to-Words dataset: comparative descriptions adapt naturally to the appropriate level of detail (orange underlines). A difficult distinction (TOP) is given a longer and more fined-grained comparison than an easier one (BOTTOM). Annotators organically use everyday language to refer to parts (green highlights).

al., 2016). Another choice is to use a second image as context, and generate a caption that distinguishes one image from another. Previous work has studied ways to generalize single-image captions into comparative language (Vedantam et al., 2017), as well as comparing two images with high pixel overlap (e.g., surveillance footage) (Jhamtani and Berg-Kirkpatrick, 2018). Our work complements these efforts by studying directly comparative, everyday language on image pairs with no pixel overlap.

Our approach outlines a new way for models to aid humans in making visual distinctions. The Neural Naturalist model requires two instances as input; these could be, for example, a query image and an image from a candidate class. By differentiating between these two inputs, a model may help point out subtle distinctions (e.g., one animal has spots on its side), or features that indicate a good match (e.g., only a slight difference in size). These explanations can aid in understanding both differences between species, as well as variance within instances of a single species.

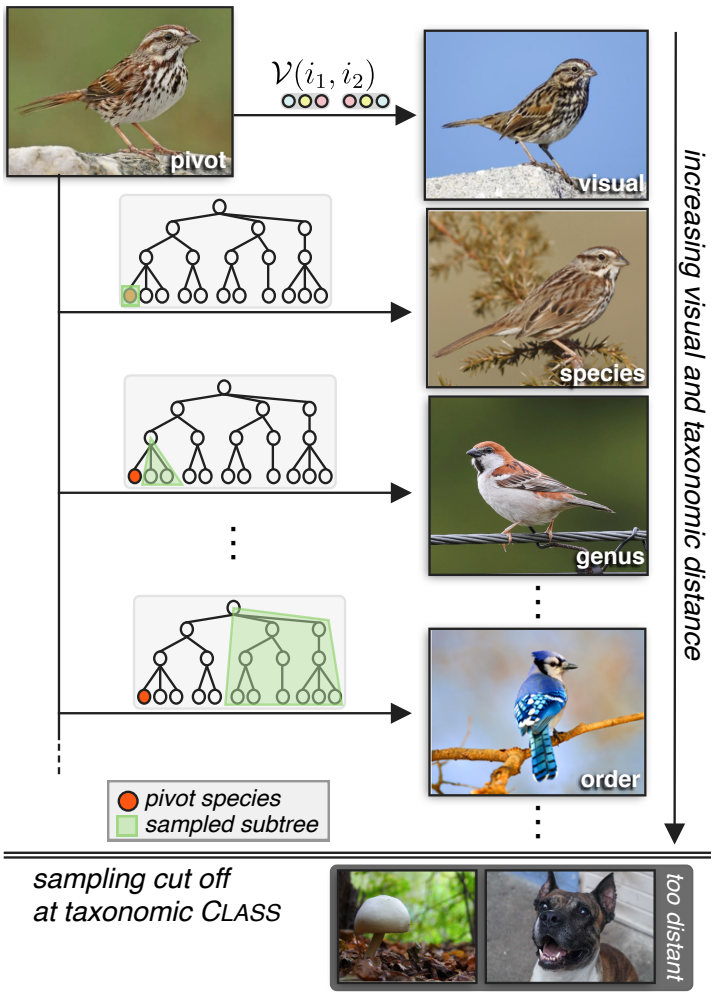


Figure 4.2: Illustration of pivot-branch stratified sampling algorithm used to construct the Birds-to-Words dataset. The algorithm harnesses visual and taxonomic distances (increasing vertically) to create a challenging task with board coverage.

4.1 BIRDS-TO-WORDS DATASET

Our goal is to collect a dataset of tuples (i_1, i_2, t), where i_1 and i_2 are images, and t is a natural language comparison between the two. Given a domain $\mathcal{D}$, this collection depends critically on the criteria we use to select image pairs.

If we sample image pairs uniformly at random, we will end up with comparisons encompassing a broad range of phenomena. For example, two images that are quite different will yield categorical comparisons (*"One is a bird, one is a mushroom."*). Alternatively, if the two images are very similar, such as two angles of the same creature, comparisons between them will focus on highly detailed nuances, such as variations in pose. These phenomena support rich lines of research, such as object classification (Deng et al., 2009) and pose estimation (Murphy-Chutorian and Trivedi, 2009).

			Images		
Dataset	**Domain**	**Lang**	**Ctx**	**Cap**	**Example**
CUB Captions *(R, 2016)*	Birds	M	1	1	*"An all black bird with a very long rectrices and relatively dull bill."*
CUB-Justify *(V, 2017)*	Birds	S	7	1	*"The bird has white orbital feathers, a black crown, and yellow tertials."*
Spot-the-Diff *(J&B, 2018)*	Surveilance	E	2	1–2	*"Silver car is gone. Person in a white t shirt appears. 3rd person in the group is gone."*
Birds-to-Words *(this work)*	Birds	E	2	2	*"Animal1 is gray, while animal2 is white. Animal2 has a long, yellow beak, while animal1's beak is shorter and gray. Animal2 appears to be larger than animal1."*

Table 4.1: Comparison with recent fine-grained language-and-vision datasets. *Lang* values: S = scientific, E = everyday, M = mixed. *Images Ctx* = number of images shown, *Images Cap* = number of images described in caption. Dataset citations: *R = Reed et al.*, *V = Vedantam et al.*, *J&B = Jhamtani and Berg-Kirkpatrick.*

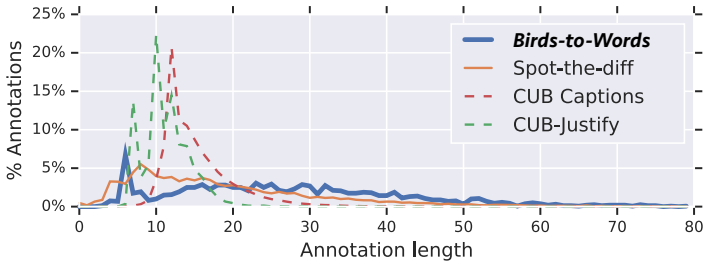
We aim to land somewhere in the middle. We wish to consider sets of distinguishable but intimately related pairs. This sweet spot of visual similarity is akin to the genre of differences studied in fine-grained visual classification (Krause et al., 2013a; Wah et al., 2011). We approach this collection with a two-phase data sampling procedure. We first select *pivot* images by sampling from our full domain uniformly at random. We then *branch* from these images into a set of secondary images that emphases fine-grained comparisons, but yields broad coverage over the set of sensible relations. Figure 4.2 provides an illustration of our sampling procedure.

4.1.1 *Domain*

We sample images from iNaturalist, a citizen science effort to collect research-grade[2] observations of plants and animals in the wild. We restrict our domain $\mathcal{D}$ to instances labeled under the taxonomic CLASS[3] *Aves* (i.e., birds). While a broader domain would yield some comparable instances (e.g., *bird* and *dragonfly* share some common body parts), choosing only *Aves* ensures that all instances will be similar enough structurally to be comparable, and avoids the gut reaction comparison pointing out the differences in animal type. This choice yields 1.7M research-grade images and corresponding taxonomic labels from iNaturalist. We then perform pivot-branch sampling on this set to choose pairs for annotation.

2 Research-grade observations have met or exceeded iNaturalist's guidelines for community consensus of the taxonomic label for a photograph.

3 To disambiguate *class*, we use CLASS to denote the taxonomic rank in scientific classification, and simply "class" to refer to the machine learning usage of the term as a label in classification.



Birds-to-Words Dataset	
Image pairs	3,347
Paragraphs / pair	4.8
Paragraphs	16,067
Tokens / paragraph	32.1 MEAN
Sentences	40,969
Sentences / paragraph	2.6 MEAN
Clarity rating	≥ 4/5
Train / dev / test	80% / 10% / 10%

Figure 4.3: Annotation lengths for compared datasets (TOP), and statistics for the proposed Birds-to-Words dataset (BOTTOM). The Birds-to-Words dataset has a large mass of long descriptions in comparison to related datasets.

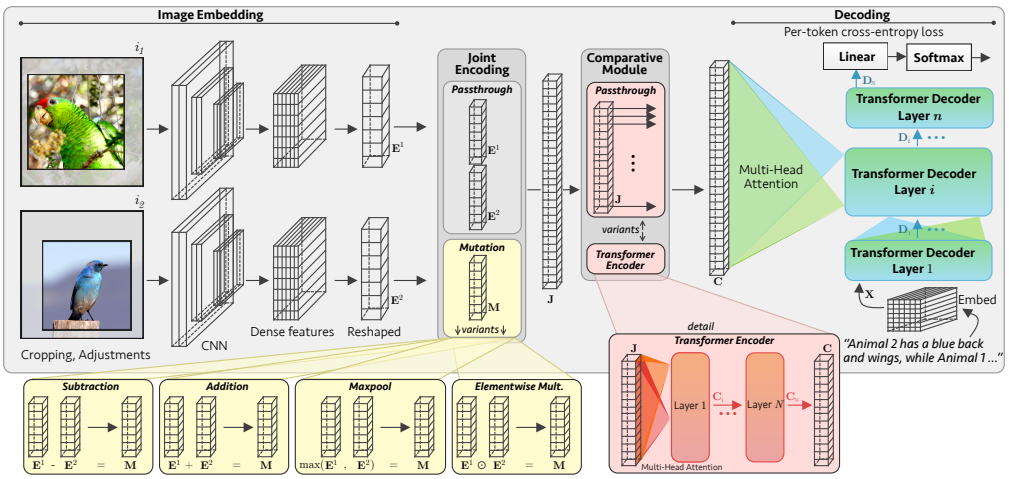


Figure 4.4: The proposed Neural Naturalist model architecture. The multiplicative joint encoding and Transformer-based comparative module yield the best comparisons between images.

4.1.2 *Pivot Images*

The *Aves* domain in iNaturalist contains instances of 9k distinct species, with heavy observation bias to more common species (such as the mallard duck). We uniformly sample species from the set of 9k to help overcome this bias. In total, we select 405 species and corresponding photographs to use as i_1 images.

4.1.3 *Branching Images*

We use both a visual similarity measure and taxonomy to sample a set of comparison images i_2 branching off from each pivot image i_1. We use a branching factor of $k = 12$ from each pivot image.

To capture visually similar images to i_1, we employ a similarity function $\mathcal{V}(i_1, i_2)$. We use an Inception-v4 (Szegedy et al., 2017) network pretrained on ImageNet (Deng et al., 2009) and then fine-tuned to perform species classification on all research-grade observations in iNaturalist. We take the embedding for each image from the last layer of the network before the final softmax. We perform a k-nearest neighbor search by quantizing each embedding and using L2 distance (Guo et al., 2016; Wu et al., 2017), selecting the $k_v = 2$ closest images in embedding space.

We also use the iNaturalist scientific taxonomy $\mathcal{T}(\mathcal{D})$ to sample images at varying levels of taxonomic distance from i_1. We select $k_t = 10$ taxonomically branched images by sampling two images each from the same SPECIES ($\ell = 1$), GENUS, FAMILY, ORDER, and CLASS ($\ell = 5$) as c. This yields 4,860 raw image pairs (i_1, i_2).

4.1.4 *Language Collection*

For each image pair (i_1, i_2), we elicit five natural language paragraphs describing the differences between them.

An annotator is instructed to write a paragraph (usually 2–5 sentences) comparing and contrasting the animal appearing in each image. We instruct annotators not to explicitly mention the species (e.g., *"Animal 1 is a penguin"*), and to instead focus on visual details (e.g., *"Animal 1 has a black body and a white belly"*). They are additionally instructed to avoid mentioning aspects of the background, scenery, or pose captured in the photograph (e.g., *"Animal 2 is perched on a coconut"*).

We discard all annotations for an image pair where either image did not have at least $\frac{4}{5}$ positive ratings of image clarity. This yields a total of 3,347 image pairs, annotated with 16,067 paragraphs. Detailed statistics of the Birds-to-Words dataset are shown in Figure 4.3, and examples are provided in Figure 4.5. Further details of our both our algorithmic approach and dataset construction are given in Appendices b.1 and b.2.

4.2 NEURAL NATURALIST MODEL

TASK Given two images (i_1, i_2) as input, our task is to generate a natural language paragraph $t = x_1 \ldots x_n$ that compares the two images.

ARCHITECTURE Recent image captioning approaches (Sharma et al., 2018; Xu et al., 2015) extract image features using a convolutional neural network (CNN) which serve as input to a language decoder, typically a recurrent neural network (RNN) (Mikolov et al., 2010) or Transformer (Vaswani et al., 2017). We extend this paradigm with a joint encoding step and comparative module to study how best to encode and transform multiple latent image embeddings. A schematic of the model is outlined in Figure 4.4, and its key components are described in the upcoming sections.

4.2.1 *Image Embedding*

Both input images are first processed using CNNs with shared weights. In this work, we consider ResNet (He et al., 2016) and Inception (Szegedy et al., 2017) architectures. In both cases, we extract the representation from the deepest layer immediately before the classification layer. This yields a dense 2D grid of local image feature vectors, shaped (d, d, f). We then flatten each feature grid into a (d^2, f) shaped matrix:

$$\mathbf{E}^1 = \langle \mathbf{e}^1_{1,1}, \ldots, \mathbf{e}^1_{d,d} \rangle = \text{CNN}(i_1)$$
$$\mathbf{E}^2 = \langle \mathbf{e}^2_{1,1}, \ldots, \mathbf{e}^2_{d,d} \rangle = \text{CNN}(i_2)$$

Figure 4.5: Samples from the dev split of the proposed Birds-to-Words dataset, along with Neural Naturalist model output (M) and one of five ground truth paragraphs (G). The second row highlights failure cases in red. The model produces *coherent* descriptions of *variable granularity*, though *emphasis* and *assignment* can be improved.

4.2.2 *Joint Encoding*

We define a joint encoding $\mathbf{J}$ of the images which contains both embedded images $(\mathbf{E}^1, \mathbf{E}^2)$, a mutated combination $(\mathbf{M})$, or both. We consider as possible mutations $\mathbf{M} \in \{\mathbf{E}^1 + \mathbf{E}^2, \mathbf{E}^1 - \mathbf{E}^2, \max(\mathbf{E}^1, \mathbf{E}^2), \mathbf{E}^1 \odot \mathbf{E}^2\}$. We try these encoding variants to explore whether simple mutations can effectively combine the image representations.

4.2.3 *Comparative Module*

Given the joint encoding of the images $(\mathbf{J})$, we would like to represent the differences in feature space $(\mathbf{C})$ in order to generate comparative descriptions. We explore two variants at this stage. The first is a direct passthrough of the joint encoding $(\mathbf{C} = \mathbf{J})$. This is analogous to "standard" CNN+LSTM architectures, which embed images and

pass them directly to an LSTM for decoding. Because we try different joint encodings, a passthrough here also allows us to study their effects in isolation.

Our second variant is an N-layer Transformer encoder. This provides an additional self-attentive mutations over the latent representations $\mathbf{J}$. Each layer contains a multi-headed attention mechanism (ATTN_{MH}). The intent is that self-attention in Transformer encoder layers will guide comparisons across the joint image encoding.

Denoting LN as *Layer Norm* and FF as *Feed Forward*, with $\mathbf{C}_i$ as the output of the ith layer of the Transformer encoder, $\mathbf{C}_0 = \mathbf{J}$, and $\mathbf{C} = \mathbf{C}_N$:

$$\begin{aligned}\mathbf{C}_i^H &= \text{LN}(\mathbf{C}_{i-1} + \text{ATTN}_{\text{MH}}(\mathbf{C}_{i-1})) \\ \mathbf{C}_i &= \text{LN}(\mathbf{C}_i^H + \text{FF}(\mathbf{C}_i^H))\end{aligned}$$

4.2.4 *Decoder*

We use an N-layer Transformer decoder architecture to produce distributions over output tokens. The Transformer decoder is similar to an encoder, but it contains an intermediary multi-headed attention which has access to the encoder's output $\mathbf{C}$ at every time step.

$$\begin{aligned}\mathbf{D}_i^{H_1} &= \text{LN}(\mathbf{X} + \text{ATTN}_{\text{MASK,MH}}(\mathbf{X})) \\ \mathbf{D}_i^{H_2} &= \text{LN}(\mathbf{D}_i^{H_1} + \text{ATTN}_{\text{MH}}(\mathbf{D}_i^{H_1}, \mathbf{C})) \\ \mathbf{D}_i &= \text{LN}(\mathbf{D}_i^{H_2} + \text{FF}(\mathbf{D}_i^{H_2}))\end{aligned}$$

Here we denote the text observed during training as $\mathbf{X}$, which is modulated with a position-based encoding and masked in the first multi-headed attention.

4.3 EXPERIMENTS

We train the Neural Naturalist model to produce descriptions of the differences between images in the Birds-to-Words dataset. We partition the dataset into train (80%), val (10%), and test (10%) sections by splitting based on the pivot images i_1. This ensures i_1 species are unique across the different splits.

We provide model hyperparameters and optimization details in Appendix b.3.

4.3.1 *Baselines and Variants*

The *most frequent* paragraph baseline produces only the most observed description in the training data, which is that the two animals appear to be exactly the same. *Text-Only* samples captions from the training data according to their empirical distribution. *Nearest Neighbor* embeds both images and computes the lowest total L_2 distance to a

	Dev			Test		
	BLEU-4	ROUGE-L	CIDEr-D	BLEU-4	ROUGE-L	CIDEr-D
Most Frequent	0.20	0.31	**0.42**	0.20	0.30	**0.43**
Text-Only	0.14	0.36	0.05	0.14	0.36	0.07
Nearest Neighbor	0.18	0.40	0.15	0.14	0.36	0.06
CNN + LSTM (Vinyals et al., 2015)	0.22	0.40	0.13	0.20	0.37	0.07
CNN + Attn. + LSTM (Xu et al., 2015)	0.21	0.40	0.14	0.19	0.38	0.11
Neural Naturalist – Simple Joint Encoding	0.23	0.44	0.23	-	-	-
Neural Naturalist – No Comparative Module	0.09	0.27	0.09	-	-	-
Neural Naturalist – Small Decoder	0.22	0.42	0.25	-	-	-
Neural Naturalist – Full	**0.24**	**0.46**	0.28	**0.22**	**0.43**	0.25
Human	0.26 +/- 0.02	0.47 +/- 0.01	0.39 +/- 0.04	0.27 +/- 0.01	0.47 +/- 0.01	0.42 +/- 0.03

Table 4.2: Experimental results for comparative paragraph generation on the proposed dataset. For human captions, mean and standard deviation are given for a one-vs-rest scheme across twenty-five runs. We observed that CIDEr-D scores had little correlation with description quality. The Neural Naturalist model benefits from a strong joint encoding and Transformer-based comparative module, achieving the highest BLEU-4 and ROUGE-L scores.

training set pair, sampling a caption from it. We include two standard neural baselines, CNN (+ Attention) + LSTM, which concatenate the images embeddings, optionally perform attention, and decode with an LSTM. The main model variants we consider are a simple joint encoding ($\mathbf{J} = \langle \mathbf{E}^1, \mathbf{E}^2 \rangle$), no comparative module ($\mathbf{C} = \mathbf{J}$), a small (1-layer) decoder, and our full Neural Naturalist model. We also try several other ablations and model variants, which we describe later.

4.3.2 *Quantitative Results*

AUTOMATIC METRICS We evaluate our model using three machine-graded text metrics: BLEU-4 (Papineni et al., 2002), ROUGE-L (Lin, 2004), and CIDEr-D (Vedantam, Lawrence Zitnick, and Parikh, 2015). Each generated paragraph is compared to all five reference paragraphs.

For human performance, we use a one-vs-rest scheme to hold one reference paragraph out and compute its metric using the other four. We average this score across twenty-five runs over the entire split in question.

Results using these metrics are given in Table 4.2 for the main baselines and model variants. We observe improvement across BLEU-4 and ROUGE-L scores compared to baselines. Curiously, we observe that the CIDEr-D metric is susceptible to common patterns in the data; our model, when stopped at its highest CIDEr-D score, outputs a variant of, *"these animals appear exactly the same"* for 95% of paragraphs, nearly mimicking the behavior of the most frequent paragraph (*Freq.*) baseline. The corpus-level behavior of CIDEr-D gives these outputs a higher score. We observed anecdotally higher quality outputs correlated with ROUGE-L score, which we verify using a human evaluation (paragraph after next).

ABLATIONS AND MODEL VARIANTS We ablate and vary each of the main model components, running the automatic metrics to study coarse changes in the model's behavior. Results for these experiments are given in Table 4.3. For the *joint encoding*, we try combinations of four element-wise operations with and without both encoded images. To study the *comparative module* in greater detail, we examine its effect on the top three joint encodings: $(i_1, i_2, -)$, $-$, and $\odot$. After fixing the best joint encoding and comparative module, we also try variations of the *decoder* (Transformer depth), as well as *decoding algorithms* (greedy decoding, multinomial sampling, and beamsearch).

Overall, we we see that the choice of joint encoding requires a balance with the choice of comparative module. More disruptive joint encodings (like element-wise multiplication $\odot$) appear too destructive when passed directly to a decoder, but yield the best performance when paired with a deep comparative module. Others (like subtraction) function moderately well on their own, and are further improved when a comparative module is introduced.

HUMAN EVALUATION To verify our observations about model quality and automatic metrics, we also perform a human evaluation of the generated paragraphs. We sample 120 instances from the test set, taking twenty each from the six categories for choosing comparative images (visual similarity in embedding space, plus five taxonomic distances). We provide annotators with the two images in a random order, along with the output from the model at hand. Annotators must decide which image contains *Animal 1*, and which contains *Animal 2*, or they may say that there is no way to tell (e.g., for a description like *"both look exactly the same"*).

We collect three annotations per datum, and score a decision only if $\geq 2/3$ annotators made that choice. A model receives +1 point if annotators decide correctly, 0 if they cannot decide or agree there is no way to tell, and -1 point if they decide incorrectly (label the images backwards). This scheme penalizes a model for confidently writing incorrect descriptions. The total score is then normalized to the range $[-1, 1]$. Note that *Human* uses one of the five gold paragraphs sampled at random.

Results for this experiment are shown in Table 4.4. In this measure, we see the frequency and text-only baselines now fall flat, as expected. The frequency baseline never receives any points, and the text-only baseline is often penalized for incorrectly guessing. Our model is successful at making distinctions between visually distinct species (GENUS column and ones further right), which is near the challenge level of current fine-grained visual classification tasks. However, it struggles on the two data subsets with highest visual similarity (VISUAL, SPECIES). The significant gap between all methods and human performance in these columns indicates ultra fine-grained distinctions are still possible for humans to describe, but pose a challenge for current models to capture.

While this evaluation setup possesses the desirable quality of being pragmatic (i.e., whether the generations can be *used* to distinguish images), it does not measure the correctness of each component of the description. Because of this, a paragraph that always helps humans choose between image pairs, but still contains incorrect information, may nevertheless receive a perfect score. To achieve a fine-grained understanding

of the generated text quality, one may consider a more systematic human evaluation to assess the precision and recall of individual facts, such as the one proposed by Kasai et al. (2021).

4.3.3 *Qualitative Analysis*

In Figure 4.5, we present several examples of the model output for pairs of images in the dev set, along with one of the five reference paragraphs. In the following section, we split an analysis of the model into two parts: largely positive findings, as well as common error cases.

Positive Findings

We find that the model exhibits **dynamic granularity**, by which we mean that it adjusts the magnitude of the descriptions based on the scale of differences between the two animals. If two animals are quite similar, it generates fine-grained descriptions such as, *"Animal 2 has a slightly more curved beak than Animal 1,"* or *"Animal 1 is more iridescent than Animal 2."* If instead the two animals are very different, it will generate text describing larger-scale differences, like, *"Animal 1 has a much longer neck than Animal 2,"* or *"Animal 1 is mostly white with a black head. Animal 2 is almost completely yellow."*

We also observe that the model is able to produce coherent paragraphs of **varying linguistic structure.** These include a range of comparisons set up across both single and multiple sentences. For example, one it generates straightforward comparisons of the form, *Animal 1 has X, while Animal 2 has Y.* But it also generates contrastive expressions with longer dependencies, such as *Animal 1 is X, Y, and Z. Animal 2 is very similar, except W.* Furthermore, the model will mix and match different comparative structures within a single paragraph.

Finally, in addition to varying linguistic structure, we find the model is able to produce **coherent semantics** through a series of statements. For example, consider the following full output: *"Animal 1 has a very long neck compared to Animal 2. Animal 1 has shorter legs than Animal 2. Animal 1 has a black beak, Animal 2 has a brown beak. Animal 1 has a yellow belly. Animal 2 has darker wings than Animal 1."* The range of concepts in the output covers *neck, legs, beak, belly, wings* without repeating any topic or getting sidetracked.

Error Analysis

We also observe several patterns in the model's shortcomings. The most prominent error case is that the model will sometimes **hallucinate differences** (Figure 4.5, bottom row). These range from pointing out significant changes that are missing (e.g., *"a black head"* where there is none (Fig. 4.5, bottom left)), to clawing at subtle distinctions where there are none (e.g., *"[its] colors are brighter . . . and [it] is a bit bigger"* (Fig. 4.5, bottom

right)). We suspect that the model has learned some associations between common features in animals, and will sometimes favor these associations over visual evidence.

The second common error case is **missing obvious distinctions**. This is observed in Fig. 4.5 (bottom middle), where the prominent beak of Animal 1 is ignored by the model in favor of mundane details. While outlying features make for lively descriptions, we hypothesize that the model may sometimes avoid taking them into account given its per-token cross entropy learning objective.

Finally, we also observe the model sometimes **swaps which features are attributed to which animal**. This is partially observed in Fig. 4.5 (bottom left), where the *"black head"* actually belongs to Animal 1, not Animal 2. We suspect that mixing up references may be a trade-off for the representational power of attending over both images; there is no explicit bookkeeping mechanism to enforce which phrases refer to which feature comparisons in each image.

4.4 RELATED WORK

Employing visual comparisons to elicit focused natural language observations was proposed by Maji (2012). Zou, Chaudhuri, and Kalai (2015) studied this tactic in the context of crowdsourcing, and Su et al. (2017) performed a large scale investigation in the aircraft domain, using reference games to evoke attribute phrases. We take inspiration from these works.

Previous work has collected natural language captions of bird photographs: CUB Captions (Reed et al., 2016) and CUB-Justify (Vedantam et al., 2017) are both language annotations on top of the CUB-2011 dataset of bird photographs (Wah et al., 2011). In addition to describing two photos instead of one, the language in our dataset is more complex by comparison, containing a diversity of comparative structures and implied semantics. We also collect our data without an anatomical guide for annotators, yielding everyday language in place of scientific terminology.

Conceptually, our work offers a complementary approach to works that generate single-image, class or image-discriminative captions (Hendricks et al., 2016; Vedantam et al., 2017). Rather than discriminative captioning, we focus on comparative language as a means for bridging the gap between varying granularities of visual diversity.

Methodologically, our work is most closely related to the Spot-the-diff dataset (Jhamtani and Berg-Kirkpatrick, 2018) and other recent work on change captioning (Park, Darrell, and Rohrbach, 2019; Tan and Bansal, 2019). While change captioning compares two images with few changing pixels (e.g., surveillance footage), we consider image pairs with no pixel overlap, motivating our stratified sampling approach to select comparisons.

Finally, the NLVR^2 dataset (Suhr et al., 2018) introduces a challenging natural language reasoning task using two images as context. Our work instead focuses on generating comparative language rather than reasoning.

4.5 CONCLUSION

We present the Birds-to-Words dataset and Neural Naturalist model for generating comparative explanations of fine-grained visual differences. This dataset features paragraph-length, adaptively detailed descriptions written in everyday language. We hope that continued study of this area will produce models that can aid humans in critical domains like citizen science.

Joint Encoding						Comparative Module	Decoder	Decoding Algorithm	Dev		
i_1	i_2	−	+	max	⊙				BLEU-4	ROUGE-L	CIDEr-D
✓	✓					6-Layer Transformer	6-Layer Transformer	Beamsearch	0.23	0.44	0.23
		✓							0.23	0.45	0.27
			✓						**0.24**	0.43	**0.28**
				✓					0.23	0.43	0.24
					✓				**0.24**	**0.46**	**0.28**
✓	✓	✓							0.22	0.44	0.22
✓	✓		✓						0.22	0.42	0.25
✓	✓			✓					0.21	0.42	0.22
✓	✓				✓				0.22	0.43	0.23
✓	✓	✓	✓	✓	✓				0.21	0.43	0.20
					✓	Passthrough	6-Layer Transformer	Beamsearch	0.00	0.02	0.00
					✓	1-L Transformer			**0.24**	0.44	0.27
					✓	3-L Transformer			**0.24**	0.44	0.27
					✓	6-L Transformer			**0.24**	**0.46**	**0.28**
		✓				Passthrough	6-Layer Transformer	Beamsearch	0.22	0.40	0.22
		✓				1-L Transformer			0.21	0.41	0.26
		✓				3-L Transformer			0.22	0.41	0.22
		✓				6-L Transformer			**0.23**	**0.45**	**0.27**
✓	✓	✓				Passthrough	6-Layer Transformer	Beamsearch	0.09	0.27	0.09
✓	✓	✓				1-L Transformer			**0.24**	0.43	0.24
✓	✓	✓				3-L Transformer			0.22	0.42	**0.26**
✓	✓	✓				6-L Transformer			0.22	**0.44**	0.22
					✓	6-Layer Transformer	1-L Transformer	Beamsearch	0.22	0.42	0.25
					✓		3-L Transformer		0.23	0.42	0.25
					✓		6-L Transformer		**0.24**	**0.46**	**0.28**
					✓	6-Layer Transformer	6-Layer Transformer	Greedy	0.21	0.44	0.18
					✓			Multinomial	0.20	0.42	0.16
					✓			Beamsearch	**0.24**	**0.46**	**0.28**

Table 4.3: Variants and ablations for the Neural Naturalist model. We find the best performing combination is an elementwise multiplication (⊙) for the joint encoding, a 6-layer Transformer comparative module, a 6-layer Transformer decoder, and using beam-search to perform inference.

	Visual	Species	Genus	Family	Order	Class
Freq.	0.00	0.00	0.00	0.00	0.00	0.00
Text-Only	0.00	-0.10	-0.05	0.00	0.15	-0.15
CNN + LSTM	-0.15	**0.20**	0.15	**0.50**	0.40	0.15
CNN + Attn. + LSTM	**0.15**	0.15	0.15	-0.05	0.05	0.20
Neural Naturalist	0.10	-0.10	**0.35**	0.40	**0.45**	**0.55**
Human	0.55	0.55	0.85	1.00	1.00	1.00

Table 4.4: Human evaluation results on 120 test set samples, twenty per column. Scale: -1 (perfectly wrong) to 1 (perfectly correct). Columns are ordered left-to-right by increasing distance. Our model outperforms baselines for several distances, though highly similar comparisons still prove difficult.

5

SCARECROW

It is commonly believed that GPT-3 is a much better underlying language model for text generation than GPT-2. Supporting evidence for this position—though strong—is largely anecdotal, or with reference to related tasks such as reading comprehension or narrative cloze (Brown et al., 2020). But just how good is the text generated by GPT-3? What kind of errors does this model make? And how does its error distribution compare with earlier language models or human authored text?

To answer these questions, we develop SCARECROW, a methodology for eliciting categorical judgements of errors in machine generated text from crowd workers. Since the goal of natural language generation (NLG) is to produce fluent outputs which can be read by laypeople, we propose that the most important errors to address are those which are recognized by readers without NLP expertise. Our framework allows crowd workers to annotate problems in model outputs at the span level. A single such annotation is shown in Figure 5.1.

To make this possible, we establish a categorization of shortcomings commonly found in machine generated text (Table 5.1). This error schema covers a broad scope of problems as identified by experts, but has been honed according to what is salient to non-expert readers through several pilot rounds of ontology-free crowd annotation. The result is a framework that is usable by everyday people with minimal training, but covers the error phenomena found in real machine generated text. Labeling spans of text using specific error types creates a picture of contemporary model generations with an unprecedented level of detail. In contrast to judging text holistically (Celikyilmaz, Clark, and Gao, 2021), insights from this method are specific and actionable, as it measures exactly how and where problems arise.

We conduct a large-scale analysis of human and machine generated text using SCARECROW, collecting 13k annotations of 1.3k paragraphs, amassing 41k spans labeled with error type, severity, and an explanation. Through this, we characterize in which ways GPT-3's generations are better than those of previous models, and which aspects do not improve with increased data and parameters. We also provide a rigorous error analysis of text generated by several other contemporary language models, examining the impact of model size, training data, and decoding strategy.

We provide our detailed annotator training system and task interface so that future researchers may employ and refine them for error analyses of machine generated text. We hope this will contribute to the standardization of NLG human evaluation (Howcroft et al., 2020).

We begin with key findings about popular language models (§5.1), then proceed to describe our setting and motivation (§5.2), annotation (§5.3, §5.4), and detailed results (§5.5, §5.6). The Appendices provide more comprehensive coverage of the error schema

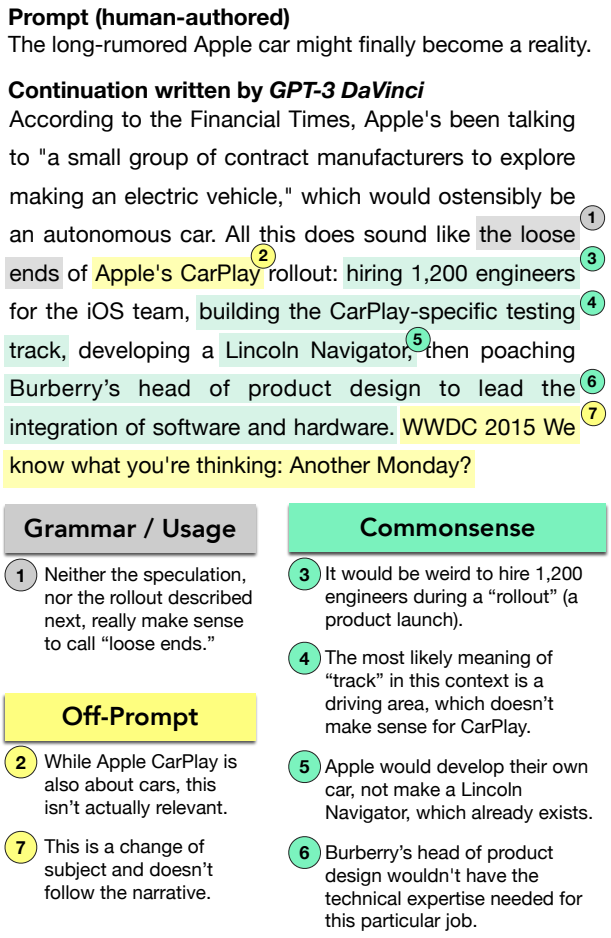


Figure 5.1: After a model (here, GPT-3 DaVinci) has read the prompt (top sentence) and generated a continuation (next paragraph), the SCARECROW annotation framework provides a systematic way for humans to mark issues throughout the text and explain what is wrong. Our own annotations are pictured here.

(c.1), crowdsourcing (c.2), annotator agreement and data quality (c.3), further analysis (c.4), and potential future directions (c.5).

5.1 KEY FINDINGS

We perform a large-scale annotation of errors in English news text generated by five sources (four models and ground truth articles). We present Figures 5.2, 5.3, and 5.4 as summaries of our main results. As a reminder to readers, Grover (Zellers et al., 2019) is the same model size and architecture as GPT-2 XL (Radford et al., 2019b), but trained in-domain (on news text). As such, our results cover three increasing model sizes (GPT-2 Small, XL, and GPT-3 (Brown et al., 2020)), one change in domain (Grover), and ground-truth text (Human). For GPT-3, we also study a variety of decoding configurations (Figure 5.4).

ERROR TYPE	DEFINITION	EXAMPLE
Language Errors		
Grammar and Usage	Missing, extra, incorrect, or out of order words	…explaining how cats feel **emoticons** …
Off-Prompt	Generation is unrelated to or contradicts prompt	**PROMPT:** Dogs are the new kids. **GENERATION:** Visiting **the dentist can be scary**
Redundant	Lexical, semantic, or excessive topical repetition	Merchants worry about **poor service** **or service that is bad** …
Self-Contradiction	Generation contradicts itself	Amtrak plans to **lay off many employees,** though **it has no plans cut employee hours.**
Incoherent	Confusing, but not any error type above	Mary gave her kids cheese toast but **drew a map of it on her toast.**
Factual Errors		
Bad Math	Math or conversion mistakes	*…it costs over £1,000* ***($18,868)*** *…*
Encyclopedic	Facts that annotator knows are wrong	***Japanese Prime Minister Justin Trudeau*** *said Monday…*
Commonsense	Violates basic understanding of the world	The dress was made at the **spa.**
Reader Issues		
Needs Google	Search needed to verify claim	***Jose Celana, an artist based in Pensacola, FL*** *,…*
Technical Jargon	Text requires expertise to understand	*…an 800-megawatt* ***photovoltaic*** *plant was built…*

Table 5.1: Error types in the SCARECROW framework, grouped into three categories. The categories are explained further in §5.3.4, and detailed definitions and examples for each error type is provided in Appendix c.1.

The main quantity we measure (on y-axes) is *span coverage*, which is the average portion of tokens that ends up covered by annotations of a particular span type. (Spans can fully nest and overlap, so there is no upper bound.) Figure 5.2 measures span coverage for each type of span separately, Figure 5.3 stacks them, and Figure 5.4 removes non-error spans (reader issues) before adding them (as in Figure 5.3, but without showing the individual types).

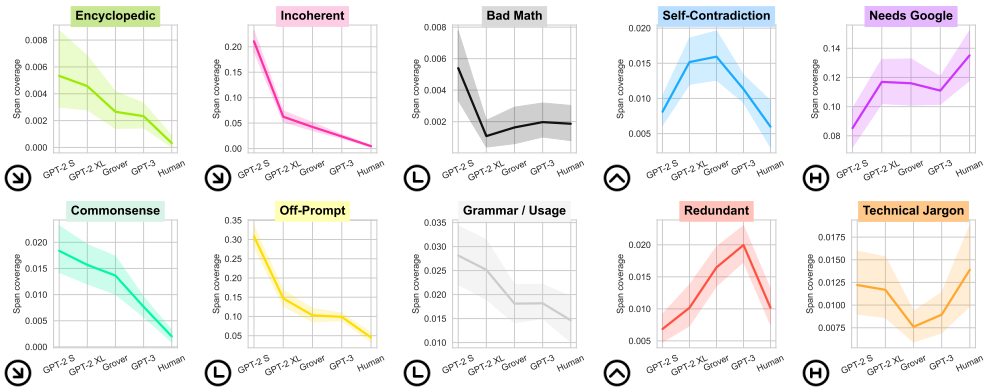


Figure 5.2: Average portion of tokens annotated with each span type (y-axis) across models (x-axis), with 95% confidence intervals.[1] We group the trends into several broad categories. **Decreasing:** fine-tuning and increasing model size improves performance. **Model plateau:** increasing model size to GPT-3 does not correlate with further improvements. **Rising and falling:** errors become more prevalent with some models, then improve. **Humans highest:** these spans are labeled most on human-authored text; both are *reader issues* (distinct from *errors*; see Table 5.1). Several of these trends are affected by decoding settings (e.g., Figure 5.4) and choice of measurement (e.g., Figure 5.8), which we discuss in §5.5. Details: all models, including GPT-3, use the same "apples-to-apples" decoding hyperparameters: top-p=0.96, temperature=1, and no frequency penalty.

The following are our key findings.

1. Scaling pays off to improve Encyclopedic, Commonsense, and Incoherent errors (Fig. 5.2). These error categories **decrease** with in-domain training (Grover) and larger model size (GPT-3). Human text still shows the fewest of these kinds of errors.

2. Scaling benefits plateau for Off-Prompt, Bad Math, and Grammar and Usage errors (Fig. 5.2). These three error categories see a **model plateau** in error reduction when scaling to GPT-3. Of these error types, humans still commit fewer **Off-Prompt** (more: §5.5.1) and **Grammar and Usage** errors, but **Bad Math** appears saturated for our domain.

3. Self-Contradiction and Redundant errors exhibit more complex scaling behavior (Fig. 5.2). We roughly categorize these trends as **rising and falling**: increasing for medium or large-scale models, but dropping for human-authored text. Further analysis (§5.5.2, §5.5.3) reveals these more complex patterns are affected both by interactions with other error types, as well how errors are counted.

1 We acknowledge trendlines imply a spurious connection between models (and humans); we use them here instead of bar plots as a visual aid.

4. Human-authored text produces the most reader issues (Figs. 5.2 and 5.3). The **Needs Google** and **Technical Jargon** span categories both have a **humans highest** trend, and both fall under *reader issues*: problems that are not necessarily *errors*, but that still prevent full comprehension or factual verification of the text (more: §5.5.4).

Furthermore, human-authored text is not free from error annotations (Figure 5.3). This can serve either as a control for baseline error rates (more: §5.5.6), or as a mechanism for critiquing human writing.

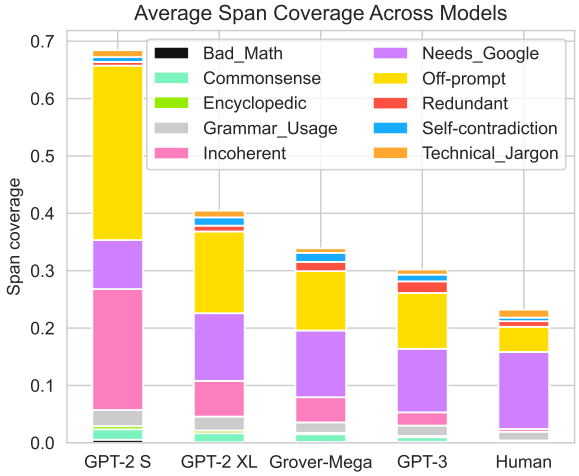


Figure 5.3: Average portion of tokens covered by span annotations, broken down by span type. All models, including GPT-3, use the same apples-to-apples decoding hyperparameters: top-p=0.96, temperature=1, and no frequency penalty. We scale each span by its token length, normalize by generation token lengths, and remove severity-1 **Grammar and Usage** errors (see §c.3).

5. Decoding hyperparameters have a huge impact (Figure 5.4). For the previous findings, we fix the sampling configuration for all models to an apples-to-apples setup for fair comparison: top-p = 0.96, (softmax) temperature = 1, and no frequency penalty (i.e., word repetition penalty; defined precisely in §5.4.2, Equation 5.1). To study the effects of these decoding settings, we annotate text generated by GPT-3 using a variety of values for top-p and temperature, both with and without a frequency penalty.

To our surprise, the decoding hyperparameters considerably affected error rates (more: §5.5.5). As seen in Figure 5.4, the worst sampling procedure for GPT-3 (argmax sampling with no frequency penalty) performed even worse than GPT-2 XL. But the best sampling procedure (surprisingly, also argmax sampling, but with a frequency penalty) produced text with as few apparent SCARECROW error spans as those authored by humans (more: §5.5.6).

All of these findings are discussed in more detail in §5.5. In the intervening sections, we describe the scope and details of our annotation framework, as well as the data we collected.

5.2 EVALUATION OF NATURAL LANGUAGE GENERATION

We make our study in the area of open-ended natural language generation, a loose term for generating longer texts with an increased level of creative freedom. Story, blog, and dialog generation are examples of open-ended generation tasks. The common factor in all open-ended generation tasks is the wide and diverse nature of target outputs. Lexically and even semantically dissimilar responses to the same prompt could be equally valid. For example, a model prompted with the blog title "Recipes for success this Holiday season" could describe how to roast a turkey or strategies for dealing with the stresses of holiday travel.

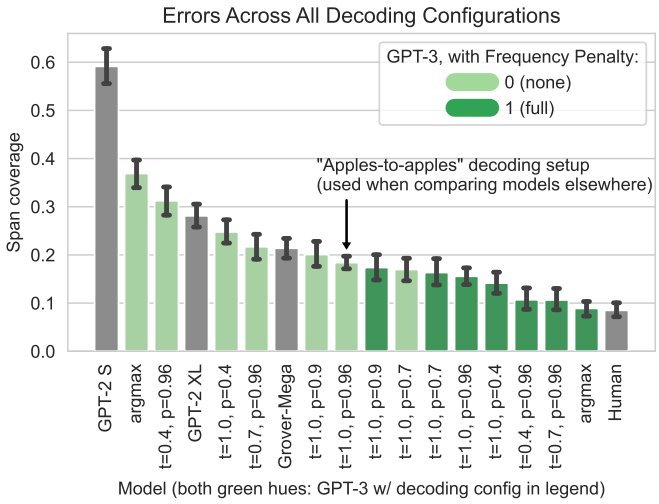


Figure 5.4: Taking the average span coverage (Figure 5.3) and removing reader issues (**Technical Jargon** and **Needs Google**), we plot values and 95% confidence intervals for all models, including all decoding hyperparameters we tested for GPT-3. We find a surprisingly large change in annotated errors depending on the decoding setting used.

This allowable variation poses a particular difficulty for the evaluation of generation systems. Traditionally, text generation quality for tasks like machine translation or graph-to-text generation has been measured by word overlap with human-authored references (Lin, 2004; Papineni et al., 2002). Though measures like BLEU allow for multiple references, they break down when the space of allowable outputs is large, as in open-ended generation. Recently introduced metrics seek to remedy this problem (Hashimoto, Zhang, and Liang, 2019; Pillutla et al., 2021), but the gold standard for evaluating generated text is still human judgment.

However, current approaches to eliciting human judgement of generated text often do not provide detailed insight into where models are making progress, where they are failing, and the scope of these failures. A/B-style testing allows for directly comparing one system against others (Clark and Smith, 2021), but can only express relative improvements. Simple Likert scale judgements can assess text quality, but do not explain why a generated text receives a given rating, or which segment of the text is problematic. Insights into model failures often come instead from a small scale expert analysis of outputs. However, these "error analyses," once a staple of NLP research, have become less common in recent years, perhaps due to their small size and high variance.

A hypothesis of the current work is that a well designed error analysis annotation framework could be used by crowdworkers to annotate large amounts of text, thereby providing detailed information about model progress and failures as well as actionable directions for future research. Such a framework would be easy to learn, reusable, and independent of particular models or experimental conditions. In what follows, we outline the details of such a method.

5.3 SCARECROW ANNOTATION METHODOLOGY

This section describes the high-level annotation methodology for SCARECROW.

5.3.1 *Prompt and Generation*

Our annotations consider two segments of text: a one-sentence prompt, and a one-paragraph generation. The *prompt* is human-written. It provides both starting tokens for model generation, as well as context for humans to evaluate whether a model is able to stay on-prompt—both topically and factually. Annotators know that the prompt is written by a human.

The *generation* is either text sampled from a language model, or the human-authored continuation to the prompt. Annotators, who do not know whether the generation came from a model or humans, assess this text. A paragraph length is chosen to balance expressiveness with scope. For expressiveness, models must be given a sufficient number of tokens to express their capabilities lexically, syntactically, and semantically. One paragraph allows for significantly more variation than a single sentence. On the other hand, assessing multiple paragraphs is challenging, both as a crowdsourcing task itself, and because it broadens the kinds of errors to include larger narrative scope. We leave extensions of SCARECROW to longer narrative lengths for future work.

5.3.2 *Span Labeling*

Annotators select spans that contain problems in the generation. The spans are automatically snapped to word boundaries. We choose spans to balance specificity (i.e.,

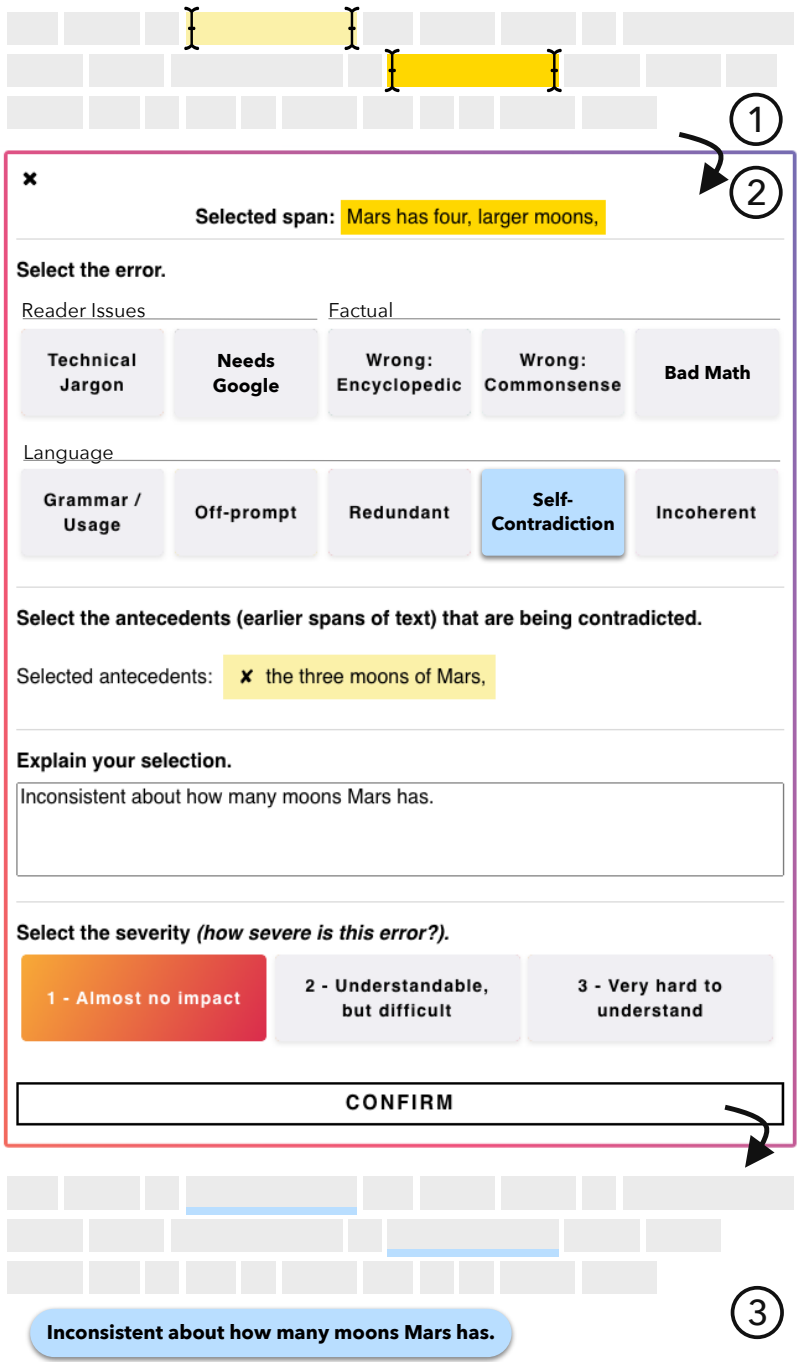


Figure 5.5: SCARECROW interface for annotating a single span: (1) highlighting a span (and later, an antecedent); (2) completing the annotation, with the span type, explanation, and severity; (3) the error annotation is saved—interactive controls allow detailed viewing and editing of spans (not shown).

vs. simply commenting on the text as a whole) with ease of use (vs. imposing a more structured annotation schema).

5.3.3 *Span Selection*

We instruct workers to select the smallest span—minimally a single word—that contains an issue. Sometimes this involves an entire phrase, sentence, or multiple sentences. We aim for specificity because during aggregation, it is possible to "back off" annotations to larger spans, but not the inverse.

Once they select a span, workers (1) label the error type, (2) choose a severity level, and (3) explain their reasoning behind the error. Workers use the annotation interface shown in Figure 5.5 to mark a span with these three steps. We describe each step in greater detail in the next three sections.

5.3.4 *Error Types*

Each selected span is labeled with exactly one error type. Multiple errors may be marked with partially or fully overlapping spans in the case that one text segment contains multiple problems.

We chose ten error types to balance three criteria: linguistic analysis, observed errors in generated text, and capabilities of everyday people with one to two hours of training.[2] We developed the schema by starting with the first two criteria (linguistic analysis and observed errors), and refining it over several pilot annotation studies, with 30 crowd workers performing 750 total annotations of 60 paragraphs before beginning data collection.

We broadly group the errors into three categories: *language* errors, *factual* errors, and *reader issues*. Language errors are issues with internal and external structure of of text: which ideas are expressed, and whether they are expressed coherently and consistently. Factual errors denote that the information presented is known to be incorrect. Reader issues, on the other hand, are cases where the text is too technical or obscure to assess its factuality. Hence, reader issues are not errors, per se, but regions where a reader would need assistance outside of the text itself for comprehension.

We present the ten error types in Table 5.1 (several pages back). Appendix c.1 provides more details, examples, and explanations for all error types.

5.3.5 *Severity*

Errors naturally vary in how jarring they are to a reader. We define three error severity levels, and ask annotators to pick one for each error.

The severity levels are as follows. **(1)** Almost no impact on quality; just a small problem. **(2)** Understandable, but difficult; what's written is still comprehensible, but there's clearly an issue. **(3)** Very difficult to understand; the error almost completely ruins the text.

We provide examples of each severity in Appendix c.2.1.

2 The complete training material is available for download.

5.3.6 *Explanation*

Finally, we ask annotators to explain their reasoning behind each error in natural language. We provide example explanations during training, but do not impose strict guidelines. This chapter primarily focuses on quantitative error analysis, but we anticipate the error explanations may warrant future investigation.

5.3.7 *Annotation Process*

Because the SCARECROW annotation relies entirely on human workers, the training, annotator selection and feedback, interface, number of annotators per instance, and any aggregation may be customized to serve a specific research analysis. We defer a discussion of our particular choices for this chapter's data collection to Section c.2.2.

5.4 DATA COLLECTION

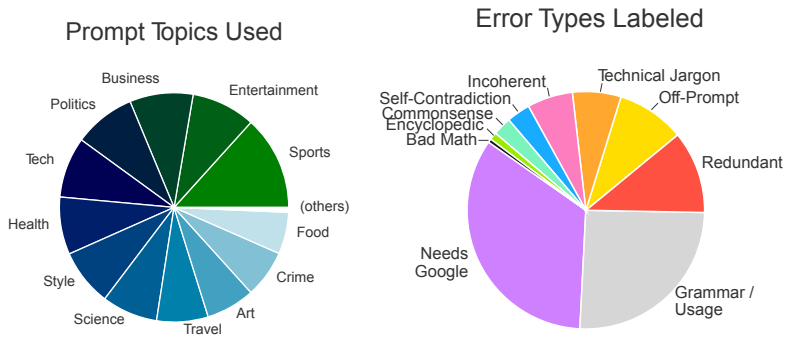


Figure 5.6: Visual overviews of the distribution of prompt topics used for generating the 1.3k paragraphs used in the annotation (left), and the types of the 41k spans labeled during the annotation (right).

We collect 13k human annotations of 1.3k paragraphs using SCARECROW, resulting in over 41k spans.

5.4.1 *Models*

We consider four model configurations to test recent state-of-the-art transformer-based (Vaswani et al., 2017) models.

GPT-2 SMALL (Radford et al., 2019b) The 117M parameter variant of GPT-2, which is pretrained on WebText, without additional fine-tuning.

GPT-2 XL (ibid.) The 1.5B parameter variant of GPT-2, (WebText, no fine-tuning).

MODEL	top-p	t	F.P.	GENS	ANNS	SPANS
GPT-2 S	0.96	1.00	0	81	809	3694
GPT-2 XL	0.96	1.00	0	81	806	3087
GROVER-MEGA	0.96	1.00	0	80	796	3006
GPT-3	0.40	1.00	0	66	660	2064
	0.70	1.00	0	65	648	1841
	0.90	1.00	0	63	629	1794
	n/a	argmax	0	66	659	2153
	0.96	0.40	0	65	650	2249
	0.96	0.70	0	61	610	1865
	0.96	1.00	0	206	2055	6234
	0.40	1.00	1	50	500	1280
	0.70	1.00	1	53	530	1481
	0.90	1.00	1	54	540	1717
	n/a	argmax	1	51	509	1384
	0.96	0.40	1	53	530	1401
	0.96	0.70	1	50	498	1369
	0.96	1.00	1	84	838	2947
HUMAN				79	789	2296
TOTAL				1308	13056	41862

Table 5.2: Statistics of data annotated with SCARECROW. t is the (softmax) temperature, and F.P. is a frequency penalty for already-generated words (explained in §5.4.2). GENS, ANNS, and SPANS are then number of generations, annotations over those generations, and error spans marked during the annotations, respectively. We perform the most annotations on the strongest available generative model (GPT-3).

GROVER-MEGA (Zellers et al., 2019) The 1.5B parameter variant of Grover, a model with the same architecture and parameter count of GPT-2, trained on news articles and their metadata.

GPT-3 DAVINCI (Brown et al., 2020) The 175B parameter variant of GPT-3, which is trained on a version of the Common Crawl web scrape with additional filtering and deduplicating.

These model choices allow us to study several factors in isolation, such as model size (GPT-2 S vs. XL) and training data (GPT-2 XL vs. Grover).

In addition, we also use the actual human-written text from the data sources we draw from, which we denote as **Human**.

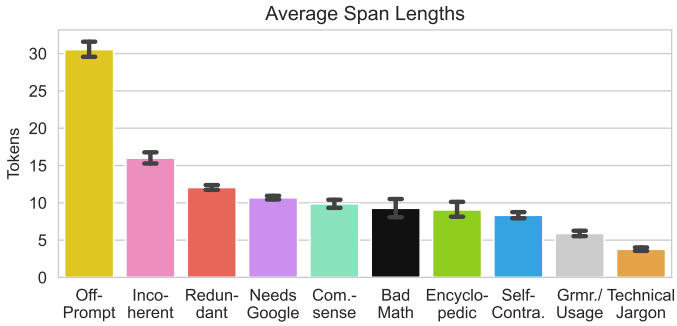


Figure 5.7: Average number of tokens covered by each annotated span. We observe span length correlates with how abstract the error category is, from word-level issues (**Technical Jargon**), through phrase-level semantics (e.g., **Commonsense**), and into problems of pragmatics (**Off-Prompt**).

5.4.2 *Decoding strategies*

We consider three main hyperparameters when sampling from models: p for *top-p* or *nucleus sampling* (Holtzman et al., 2020b), an alternative to *top-k*;[3] t for the *softmax temperature*; and *f.p.* for *frequency penalty*. The frequency penalty scales a token's likelihood based on how many times it was already generated by applying the following modification to the model's output:

$$\ell_i(t) \longleftarrow \ell_i(t) - c_{<i}(t) \cdot \alpha_f \tag{5.1}$$

where $\ell_i(t)$ is the model's output for token t at the i-th position,[4] $c_{<i}(t)$ is the count of token t's sampled occurrences prior to the i-th position, and α_f is the frequency penalty. We omit studying *presence penalty*, another hyperparameter offered for GPT-3, simply due to annotation budget constraints.

To compare models as consistently as possible, we set identical decoding strategies for our primary data collection. We refer to this as the "apples-to-apples" decoding setup throughout the chapter:

$$p = 0.96 \qquad t = 1.0 \qquad f.p. = 0$$

However, we also wish to study the effects of these decoding strategies. We annotate generations from the strongest available model (currently, GPT-3) varying the following parameters:

3 We omit separate studies of *top-k*, due to results presented by Holtzman et al. (2020), and OpenAI's removal of *top-k* from the GPT-3 API.

4 While $\ell_i(t)$ is defined to be *"logits (un-normalized log-probabilities),"* because it is un-normalized, we anticipate that it is simply the model's output before the log(softmax(·)) is applied. See OpenAI's description of frequency and presence penalties: `https://beta.openai.com/docs/api-reference/parameter-details`

$$p \in \{0.4, 0.7, 0.9, 0.96\}$$
$$t \in \{0.0 \text{ } (argmax), 0.4, 0.7, 1.0\}$$
$$\text{f.p.} \in \{0 \text{ (none)}, 1 \text{ (full)}\}$$

For budget reasons, we only vary p and t independently—i.e., we set $p = 0.96$ when varying t, and $t = 1.0$ when varying p.

5.4.3 *Prompt Selection*

We use news articles as the sources of prompts for models to condition on for generation. Specifically, we use news articles found in the Common Crawl. We select the first sentence as the prompt.

Our use of news text is constrained by two factors. First GPT-3 is trained on the Common Crawl, from 2016 through 2019. We wish to avoid testing GPT-3 by generating from articles it saw during training, due to the possibility of copying (Carlini et al., 2020). Second, news articles began heavily covering the COVID-19 pandemic beginning around February 2020. Though testing models' capabilities to generate text about unseen events is a valuable line of study, the distribution shift caused by COVID-19 in news writing about all aspects of life is difficult to overstate.

As such, to make the comparison more amenable to models' training data, we consider news articles from January 2020. We select articles where there is a known topic—such as *Food* or *Sports*—from the Common Crawl metadata, to allow for studying any effect of coarse-grained subject.

5.4.4 *Generation*

We generate between 80 and 145 tokens[5] from each model as a continuation to the first sentence of the news article. We stop generating when we heuristically detect the first sentence boundary after 80 tokens. If the model does not end a sentence between 80 and 145 tokens, we sample again. For the *Human* setting, we use the remainder of the article, similarly stopping after the first sentence boundary after 80 tokens.

5.4.5 *Annotation*

CROWDSOURCING Workers first complete training and qualification tasks. We provide more details in Appendix c.2.2. From pilot studies, we discovered that each error, depending on its severity and clarity, has a < 100% chance of being identified by each worker. In other words, a human annotator can be seen as a high-precision, moderate-recall stochastic process for labeling text problems. To account for this, we have **10 workers** annotate each paragraph.

5 Counted by Stanza tokenization (Qi et al., 2020), not byte-pair encoding (BPE) or whitespace-separated tokens.

DATASET STATISTICS We list the data collection quantities in Table 5.2, and plot visualizations of three aspects: prompt topic and annotated span proportions are shown in Figure 5.6, and average span lengths are shown in Figure 5.7.

5.5 DETAILED ANALYSIS

In this section we perform a more detailed analysis of the trends of individual error types and decoding configurations.

To begin, we consider apples-to-apples model decoding configurations. To expand on these results, originally presented in Figure 5.2, we also present two additional ways of counting error spans, which we show in Figure 5.8. While our method for counting errors throughout the chapter takes into account the number of tokens covered in each span (*span coverage*), we also show plots for scaling each span by its severity level (*span coverage* × *severity*), and by ignoring both severity and token length (simply *span counts*). These changes in measurement further illuminate model error characters, which we discuss in the upcoming sections (refer to Figure 5.8).

5.5.1 *Off-Prompt*

Under initial analysis of *span coverage*, **Off-Prompt** errors show a *model plateau* at GPT-3. Measuring *span counts* offers barely perceptible improvement, indicating that scaling language models over more in-domain training does not guarantee topicality.

This observation is consistent with growing work on *prompt programming* as a new technique for attempting to steer large pretrained models to complete the desired task (Branwen, 2020; Gao, Fisch, and Chen, 2020; Reynolds and McDonell, 2021). In practice, we observe that while GPT-3 will sometimes continue a prompt by writing an article, other times, it may elaborate on the prompt itself:

> **PROMPT**
> Do you prefer the idea of being outdoors in the fresh air to being stuck inside with phones ringing and messages pinging?
>
> **GPT-3**
> Can you leave work at work? Are you flexible enough to **cover holidays or take on additional responsibilities? Can you prioritize tasks? If your boss comes to you on Tuesday to confirm the new social media strategy, are you able to pick up the ball and get the messaging hammered out by Thursday?** ...

Of course, this generation is not *literally* **Off-Prompt**, but it is out of place when other generations are continuations of the prompt, rather than further elaborations of it.

While avoiding **Off-Prompt** issues for language models is worth exploring with prompt programming and other avenues, an investigation of these techniques is outside the scope of this work.

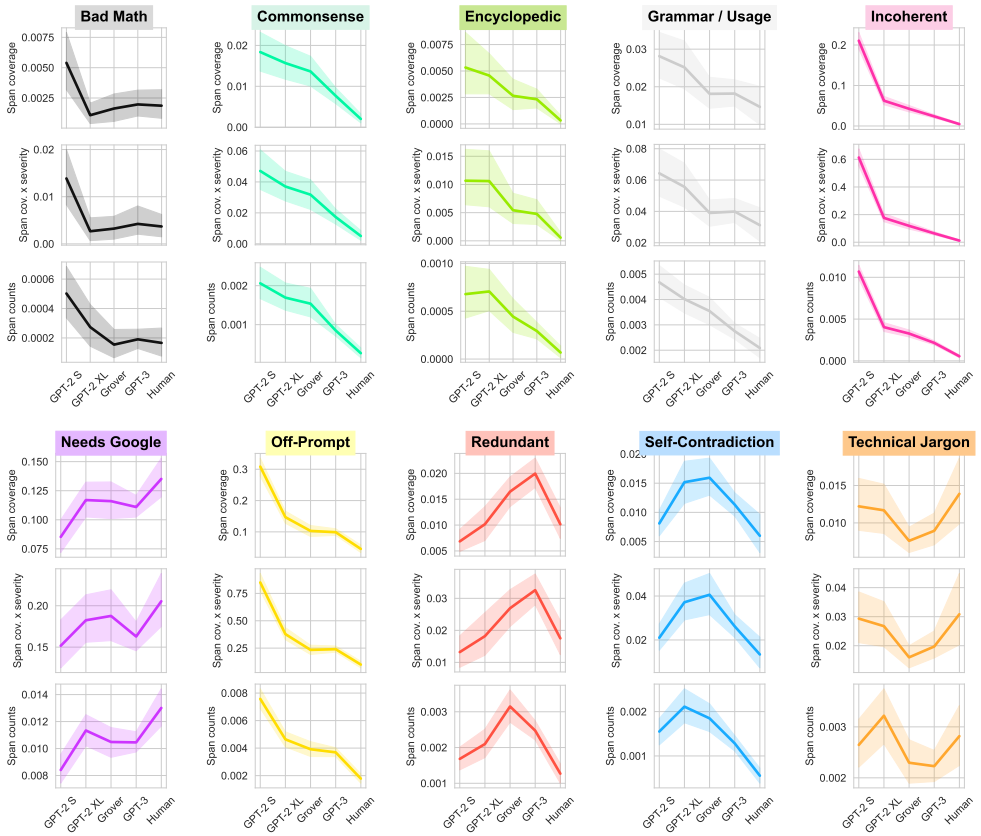


Figure 5.8: Comparison of three different ways of measuring quantities of error span annotations, shown per label. (The top plot for each span type is identicial to the one shown in Figure 5.2.) The top method (*span coverage*) is used in the rest of the chapter; we provide the comparisons here to illustrate how this decision affects analysis. **Top subplots:** *span coverage*, where the number of tokens annotated as the error span are divided by the length of each annotation. (Annotations with no spans count as 0.) Intuitively, this measures the expected portion of tokens that will be covered by an error span. **Middle subplots:** *span coverage* × *severity*, like the top measure, but each span's token count is multiplied by its severity, more harshly penalizing errors intuitively marked as worse. **Bottom subplots:** *span counts*, where each error span simply counts as 1, regardless of the span length. In all cases, model configurations are set as closely as possible (top-p = 0.96, t = 1.0, no frequency penalty), severity-1 grammar errors are removed (see §c.3), and 95% confidence intervals are shown as bands. **Takeaways:** Compared to the approach used in the rest of the chapter (*span coverage;* top), scaling by severity (middle) does not affect the relative model ordering, primarily widening confidence intervals. However, ignoring span lengths (bottom) does affect the results in several cases. **Grammar and Usage** and **Encyclopedic** develop clearer *decreasing* shapes, previously suffering from various levels of *model plateau* at GPT-3. Furthermore, the relative model ordering is changed for **Redundant**, **Self-Contradiction**, and **Technical Jargon** spans.

Finally, we note that **Off-Prompt** spans are the most prevalent *error* (not reader issue) marked for human-authored text. We suggest that a higher rate of false positives

for this error type, coupled with its prevalence in model-generated text, makes further refinement of this error a compelling avenue for further study.

5.5.2 *Self-Contradiction*

While changing from *span coverage* to *span counts* alters the relative order of GPT-2 XL and Grover (though still within confidence bounds), the puzzling question is why GPT-2 Small performs better than most (or all) other models. Why would the smallest model produce the fewest **Self-Contradiction** errors?

We posit the reason is that GPT-2 generations are so **Incoherent** and **Off-Prompt** that there is little opportunity for relevant, comprehensible points to be made and then reversed. For example, see the GPT-2 Small annotated generation in the top left of Figure c.2. The entire text is covered by **Off-Prompt** and **Incoherent** errors.[6] If we look at GPT-2 Small's error distribution in Figure 5.3, we see most of its added density comes from significantly more **Off-Prompt** and **Incoherent** tokens.

5.5.3 *Redundant*

The different counting methods shown in Figure 5.8 reveal a change in the results for **Redundant** errors. Rather than repetition simply increasing as models grow larger, we observe that GPT-3 repeats in a similar number of cases (lower span *counts*), but for more tokens (higher span coverage). This matches the qualitative observation that GPT-3 produces larger *topically* repetitive blocks, rather than simple word or phrase repetitions generated by GPT-2-sized models:

> **GPT-2 Small**
> ... owners have started growing their own breeds and dogs are **starting to start** so there's really ...
>
> **GPT-3**
> The focus of your thoughts should be on the task at hand, **not on your productivity. You shouldn't be thinking about how you can be more productive. You should be thinking about how you can be productive right now.** ...

Such repetitions can be more difficult to clearly isolate, because even slight wording changes produce variations in tone and connotation. Rather than being identical *semantically*, we observe GPT-3 will seem stuck on a particular *topic*, elaborating on and rephrasing similar ideas more times than a human writer (hopefully) would.

5.5.4 *Reader Issues*

As noted in §5.1, we observe the highest number of **Needs Google** and **Technical Jargon** issues in human-authored text.

6 The high double-error coverage reveals another consideration: to what *depth* (i.e., number of overlapping spans) will annotators mark? By the design of our framework, **Incoherent** errors serve as a fall-back, but without it, we might imagine poor generations splatter-painted by other error types.

Needs Google issues broadly represent any specific claim that could be fact-checked. In our domain (news articles), these are primarily whether an event happened on a particular day, whether a person holds a role, or whether a mechanism works as described (e.g., chemical or technical). As seen in Figure c.6 (which shows GPT-3's span distribution), **Needs Google** issues happen roughly equally for all topics. We believe this trend is due to the news article domain, which is prone to a high density of specific information. As such, for other domains, this trend may be less prevalent, more difficult to label (e.g., subtle claims assumed to be true in long running text), or both.

We observe that **Technical Jargon** issues are influenced by topic (Figure c.6, bottom), occurring significantly more frequently in *Business, Health, Science,* and *Technology* topics than in others. This trend displays a clear topic-dependence even within a single broader domain (news). These results indicate that both reader issues are characteristics of natural text. Of course, one might wish to measure or minimize potential reader issues for a particular application—for example, claim verification, or controlling for reading level.

5.5.5 *Decoding Hyperparameters*

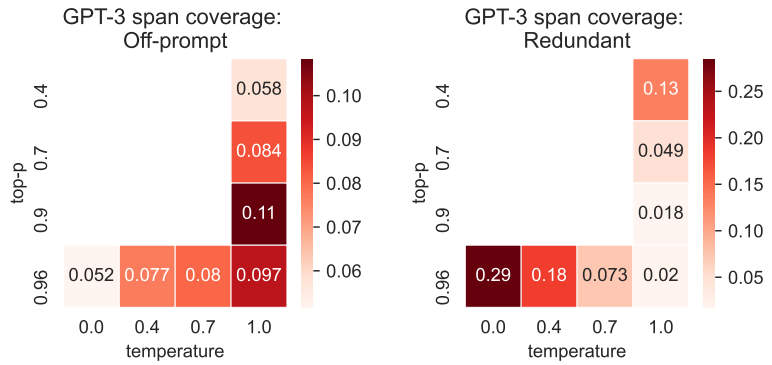


Figure 5.9: GPT-3 span coverage for **Off-Prompt** (left) and **Redundant** (right) for values of top-p and temperature (t = 0 is argmax; both plots with no frequency penalty; argmax sampling is agnostic to the top-p value, so we simply plot it in the p = 0.96 cell). **Takeaway:** Our annotation confirms intuitive expectations of the effect of sampling on two error categories. When sampling from a larger pool of words (higher p and t), a model is more likely to veer **Off-Prompt**, but less likely to produce **Redundant** text.

We discuss the effects of the decoding hyperparameters we consider—top-p, temperature, and frequency penalty—on generation quality. For the sake of annotation cost, we only vary these parameters for the strongest model available, GPT-3.

First, we show the effect of varying top-p and temperature alone (i.e., with no frequency penalty) on different span types. Figure 5.9 shows the effect on two salient spans: **Off-Prompt** and **Redundant**. (We omit others for space.) We observe that

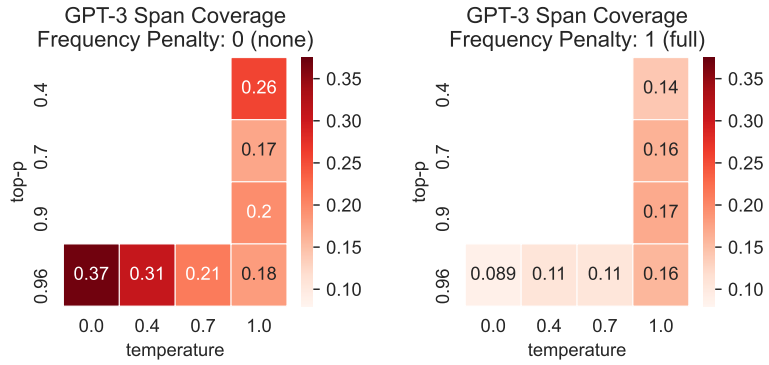


Figure 5.10: Comparison of *frequency penalty* off (left) and full (right) for GPT-3 (removing reader issues and severity-1 **Grammar and Usage** errors; argmax sampling is agnostic to the top-p value, so we simply plot it in the $p = 0.96$ cell). We observe the frequency penalty improves average span coverage for all values of top-p and temperature. Furthermore its trend is reversed: with a frequency penalty, the least diverse sampling mechanisms (low temperature and low top-p) now produce text with the fewest error spans, rather than the most. (See Figure 5.4 for confidence intervals on each value.)

annotators naturally label errors the way we would intuitively expect the model to produce them, given the hyperparamter changes. The bottom-right corner of each subplot, where $t = 1$ and $p = 0.96$, is the configuration with the highest amount of randomness from sampling. As we move away from that corner—either left by lowering temperature, or up by lowering top-p—we lower the amount of randomness. We observe a positive correlation with randomness and **Off-Prompt** errors, and an inverse correlation with **Redundant** errors. In other words, sampling from a larger set of words makes the model more prone to changing topics, but less likely to repeat itself, and vice versa.

After confirming these intuitive measures, we turn our attention to Figure 5.10, which investigates the overall error spans for GPT-3 both without (left) and with (right) the frequency penalty. (Note that unlike Figure 5.9, both heatmaps in Figure 5.10 have the same color scale.) We observe that introducing the frequency penalty lowers error rates for every value of temperature and top-p that we try. Furthermore, it appears to reverse the trend seen without a frequency penalty: that sampling from a larger set of words produces fewer errors.

The overall results for all decoding configurations were shown previously in Figure 5.4. In the next section, we focus on the GPT-3 decoding configuration that produced the fewest number of errors, and compare it to human authored text.

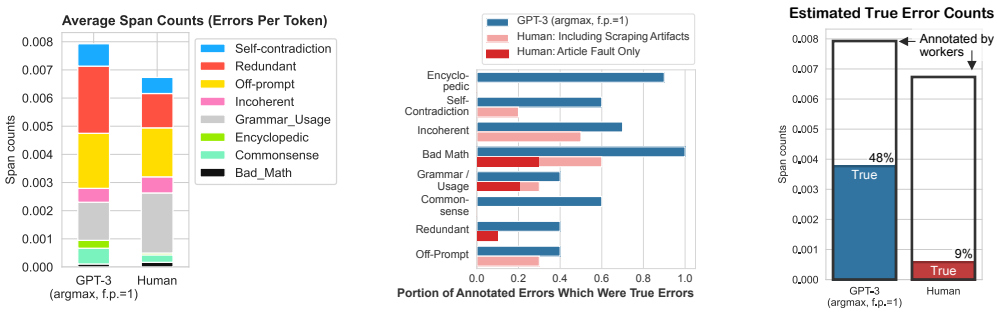


Figure 5.11: Analysis of the best GPT-3 configuration (*argmax, freq. penalty = 1*) vs. human-authored text. **Left:** A breakdown of errors by type. **Center:** Results of manually annotating 10 random spans from each type with whether the error was legitimate. For human-authored text, we also show errors marked on scraping artifacts that were present in the Common Crawl data. **Right:** Scaling each error type (*left plot*, now shown in black outline) by the portion of errors found to be legitimate (*center plot*), we estimate the true errors counts for each model (color-filled portions). **Takeaway:** Humans have more difficulty spotting errors in higher quality text; accounting for this difference dramatically increases the gap between model-authored and human-authored text. For simplicity, all plots use error *counts* rather than error *coverage*—i.e., they count the number of error spans, rather than scaling by the number of tokens covered.

5.5.6 *Best GPT-3 vs. Humans*

The best GPT-3 configuration shown in Figure 5.4—argmax sampling with frequency penalty = 1—appears to match error rates seen in human text. Is the text generated by this model truly as error-free as news articles?

We first look at the error composition of both sets of annotations. To get a clear picture of the potential problems, we plot only error spans (ignoring reader issues), and we omit length scaling, instead plotting span counts. This breakdown is shown in the left plot of Figure 5.11. The error compositions are similar, the largest differences being more **Redundant** errors for GPT-3, and more **Grammar and Usage** errors for human-authored text.

Next, we perform a manual analysis of 160 errors, sampling 10 at random from each of the 8 error types for each model (GPT-3 and human-authored text). We show the results in the center plot of Figure 5.11. We notice that a greater portion of errors in human-authored text were due to artifacts present in the text-only format of the Common Crawl. For example, links to other articles or advertisements sometimes appear in the middle of an article's text. While annotators were quick to mark these spans, they reflect errors in formatting, not in writing. We partition these errors separately and exclude them from the subsequent calculations.[7]

7 GPT-3's generations also sometimes exhibited what appeared to be formatting errors due to training on web-scraped text, though more rarely. For example, some generations contained *Which?* after vague noun phrases, which appear to be learned from Wikipedia, where under-specified information is tagged by an editor with this word. For fairness, we removed these errors from GPT-3's tally as well, though they were few enough we do not plot them separately.

Finally, we scale each error type's prevalence for each model (i.e., the left plot of Figure 5.11) by the portion of errors that we estimate to be legitimate based on our manual annotation (i.e., Figure 5.11, center) to produce the right plot of Figure 5.11. After taking into account each error type's frequency, we estimate that 48% of GPT-3's worker-annotated errors overall are legitimate, compared to 9% for human-written articles.

This analysis suggests two findings. First, human-authored news paragraphs contain many times fewer issues than text authored by GPT-3 using the best decoding configuration we tested. Second, the noise of error annotations may be as high as 90% when assessing high-quality text. Though it would require further manual annotation to verify, we conjecture that the trend of GPT-3's error spans being more reliable (only 50% noise) would continue, and that text generated by GPT-2 would contain even fewer false positives. We note that such rates are not fixed—after all, the manual annotations were done by one of the authors simply by reading carefully—but that more realistic text may require correspondingly more effort by human annotators.

5.6 ERROR PREDICTION

A natural question is: using this data, can machines learn to detect and classify errors in machine generated text?

TASK We frame this problem as a span classification task. Given a span from a generated text, the goal is to classify its error type or output "No Error" if there is none. Positive examples for each error class are taken from our data. We sample random spans that were not labeled with any error type as negative examples. To ensure a breadth of span lengths, we sample 3 negative spans for every length of error span in the generated text. We split the generated texts into train, development, and test sets using 1063 texts (28029 error spans), 100 texts (2538 spans) and 100 texts (2677 spans) respectively.

MODEL We use a standard span classification model inspired by Wadden et al. (2019). This model encodes every generated text using a pretrained language model (RoBERTa-large). Spans are represented with the final layer of this encoding. Following previous work, we concatenate the start and end tokens with a task-specific learned length embedding. The resulting vector is passed through a feedforward network which reduces its dimensionally to the number of error categories plus a "No Error" option. The resulting model has 357M trainable parameters. The model is trained to minimize the cross entropy of the correct span category. We train for 15 epochs using AdamW with a learning rate of 10^{-6}. We validate after each epoch and use the checkpoint with the lowest validation loss (epoch 8).

EVALUATION To evaluate the error prediction model, we use per-token precision, recall, and F_1 score per error category. We classify every span up to length 30 in a

generated text. We take as gold labels the aggregated human error spans collected in our data. For comparison, we also report the average metrics of one annotator versus the others.

RESULTS Table 5.3 shows the error prediction capability of this model in terms of precision and recall. As we noted earlier, a single human annotator can be thought of as a high precision, low recall judge. These results bear out this claim. For all but one category, humans have higher precision annotations. However, the models trained on the aggregation of human labels can achieve considerably higher recall. For half of the error categories, this leads to higher model F_1 scores than the human annotators.

We see that the model is quite successful at identifying information that human's would have to manually verify (**Needs Google**), achieving nearly perfect recall with precision close to 0.6. The model can also identify **Grammar and Usage**, **Incoherent**, and **Redundant** errors with higher recall than an individual human annotator, though at the cost of precision (sometimes in the .20s).

These results show some promise for training error detection models on our data. Notably, a thorough search of hyperparameters such as learning and negative sampling rates has not been conducted and could possibly improve even the basic model offered here. Architecture choices such as the underlying pretrained language model, the span representation, or the structure of the final classification module should also be explored. A different framing of the error prediction task (i.e., rather than exhaustive span classification) may also yield better performance.

Error	Model			Human		
	P	R	F_1	P	R	F_1
Bad Math	–	0	–	0.72	0.14	**0.24**
Commonsense	0.77	0.06	**0.10**	0.17	0.02	0.04
Encyclopedic	–	0	–	0.22	0.03	**0.05**
Grammar and Usage	0.29	0.23	**0.26**	0.30	0.04	0.08
Incoherent	0.59	0.34	**0.43**	0.69	0.15	0.24
Off-Prompt	0.67	0.29	0.41	0.88	0.31	**0.46**
Redundant	0.23	0.82	0.36	0.88	0.35	**0.50**
Self-Contradiction	0.08	0.23	0.12	0.51	0.09	**0.16**
Technical Jargon	0.18	0.74	**0.29**	0.61	0.12	0.20
Needs Google	0.59	0.96	**0.73**	0.78	0.20	0.32

Table 5.3: Model prediction results. Bold F_1 scores denote the higher average; values marked "–" cannot be computed due to division by zero. **Takeaway:** Humans have higher precision in every error type except **Commonsense**, but relatively sparse annotations lead to lower computed recall. This allows the model to achieve higher F_1 scores for half of the span categories.

5.7 RELATED WORK

Automated evaluation metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), and BERTScore (Zhang et al., 2019) compute a generation's score based on a true reference, or a set of references. Their use is well-established in tasks like machine translation and summarization, but they are less helpful in open-ended text generation, where there is a vast diversity of possible high-quality continuations.

Recent studies propose automated metrics for open-ended text generation evaluation such as: Perception Score (Gu, Wu, and Yu, 2020), which diffuses evaluation onto a multidimensional space and assigns a single holistic score; UNION (Guan and Huang, 2020), which learns to distinguish human-written stories from negative samples by generating perturbations of human written stories; and MAUVE (Pillutla et al., 2021), which compares the distribution of machine-generated text to that of human language.

Method	**GC**	**SET**	**DE**	**RR**	**EE**	**RS**	**SA**
Likert-Scale	✓		✓			✓	
RankME	✓			✓		✓	
RoFT	✓		✓		✓		
SCARECROW		✓	✓		✓		✓

Table 5.4: Comparison of different natural language generation human evaluations. Here, **GC** : General Criteria, **SET** : Specific Error Type, **DE** : Direct Evaluation, **RR** : Relative Ranking, **EE** : Error Explanation, **RS** : Rating Scale, **SA** : Span Annotation.

An alternate recent approach to assessing open-ended text generation was presented in TuringAdvice (Zellers et al., 2021a), where crowd workers assess machine-generated advice in response to Reddit posts. In their error analysis, Zellers et al. connect problems in generated text to core NLP tasks, such as **Self-Contradiction** errors as instances of failed natural language inference (Monz and Rijke, 2001), or **Off-Prompt** errors as cases of failed reading comprehension (Richardson, Burges, and Renshaw, 2013). While past work has attempted to guide text generation using discriminative models trained for such tasks (Holtzman et al., 2018), it remains an open challenge.

Comparative human evaluations of natural language generations ask annotators to rank system outputs relative to each other. Text is typically evaluated using a few global criteria, such as fluency and relevance, using discrete (e.g., 5-point) (Sai, Mohankumar, and Khapra, 2020) or continuous scales (Novikova, Dušek, and Rieser, 2018). Recent work even automates this approach, running a human evaluation alongside automatic metrics on leaderboard submissions (Khashabi et al., 2021). In the RoFT system (Dugan et al., 2020), annotators attempt to detect the boundary between human- and machine-written text as a proxy for assessing quality. Table 5.4 summarizes the differences between these schemes and SCARECROW. See Celikyilmaz, Clark, and Gao (2021) for a recent survey of text generation evaluation techniques across both human and automatic metrics.

While these approaches may be helpful—sometimes (Card et al., 2020)—at ranking systems, they do not give us insight into exactly *which* parts of a generation fall short, and *why*. One approach related to or annotation method is pursued by Wood et al. (2018), who develop a collaborative mobile app where users draw "graffiti" commentary on news articles. SCARECROW aims to assess model generations the way we would critique human-written text: by locating, coarsely categorizing, and explaining problems.

5.8 CONCLUSION

We present SCARECROW, a method for identifying and explaining issues in generated text. Along with the annotation framework, we present an analysis of the SCARECROW method applied to several large neural language models in an open-ended news generation task. We release our data and methodology to the community.

6

CONCLUSION

In this dissertation, we described four projects bridging natural language and the natural world with an eye towards commonsense knowledge. Verb Physics and Social Chemistry explored explicit representations of physical and social commonsense. Commonsense was implicit in Neural Naturalist and Scarecrow, where language was used or evaluated in comparative settings. All of these projects seek to move NLP's perspective of natural language beyond textual corpora.

Considerable work remains. Here, we humbly offer a suggestion for busting language out of the textual prison we, researchers in NLP, have put it in.

The field should be renamed. Natural language processing should become **communication processing**. With this change, modes of communication beyond written text must become first-class objects of study: speech[1] (Hinton et al., 2012), gesture (Cheok, Omar, and Jaward, 2019), facial expression (Shan, Gong, and McOwan, 2009), body language (Marmpena et al., 2019), and eye gaze (Admoni and Scassellati, 2017). Whether language originally evolved as a tool for thought (Reboul, 2015), all language studied in NLP today is used for communication. Even still, just a small fraction of NLP research incorporates these modalities (Baltrusaitis, Ahuja, and Morency, 2019; Muller et al., 2013). By ignoring channels of human communication that contain considerable information (Mehrabian and Ferris, 1967; Mehrabian and Wiener, 1967; Pease and Pease, 2008), we are left with predictable failure modes due to the lack of context in written and transcribed text. In one way of looking, we are simply working with lost data. We have accepted such limitations for years. It is time to move forward.

Focusing on communication will radically raise our standards of **grounding**. Even in this dissertation, *grounding* is used to refer to other parts of the text (e.g., §3.2.3). Instead, *grounding* should link language to other representations of the world. We have begun this with image and video (Das et al., 2013; Lin et al., 2014), instructions (Artzi and Zettlemoyer, 2013; Blukis et al., 2018), and interaction (Kojima, Suhr, and Artzi, 2021; Zellers et al., 2021b). But rather than grounding into isolated environments, breaking NLP out of its cage will require observing real world social interactions (Bisk et al., 2020; Hovy and Yang, 2021).

As a field, we do not yet know what problems we will face when broadening our study to these domains. What a perfect time to find out.

1 Here by *speech* we mean not simply transcription, but the inclusion of additional linguistic phenomena like prosody.

Part III

APPENDIX

a

APPENDIX MATERIAL FOR SOCIAL CHEMISTRY

A.1 ADDITIONAL DATASET DETAILS

A.1.1 *Situations*

DOMAINS We provide here a more thorough description how we collected situations from the four domains we consider. Figure a.1 gives more example situations from each domain.

1. **`r/amitheasshole`** (30k) — The *Am I the Asshole? (AITA)* subreddit. This posts of this subreddit pose moral quandries, such as *"AITA for wanting to uninvite an (ex?)-friend from my wedding for shit-talking our marriage?"* We use the data from Lourie, Bras, and Choi (2020). They scrape the titles of posts, omitting the preamble (e.g., *"AITA for"*), normalizing to present tense, and filtering out administrative posts. We do not use any annotations provided by that community (where other posters vote who had the moral high ground).

2. **`r/confessions`** (32k) — The *Confessions* subreddit. This posts of this subreddit discuss personal stories, often with interpersonal conflicts, such as *"I feel threatened by women prettier than me."* As with `r/AITA`, we scrape only the titles of these posts. This subreddit contains a high volume of hateful or disturbing content; we attempt to filter the worst of this using keywords, and also allow annotators to mark dark or disturbing items.

3. **`rocstories`** (30k) — The ROCStories corpus from (Mostafazadeh et al., 2016). ROCStories involve stories about everyday situations, and are generally less controversial than the other sources, e.g., *"They weren't sure either so he started asking friends."*. We select a subset of the sentences from ROCStories which are likely to involve two character references based on POS tagging (Toutanova et al., 2003), personal pronouns, and WordNet (Miller, 1995). We then randomly sample to pick 30k sentences.

4. **`dearabby`** (12k) — Titles of the Dear Abby advice column. These titles are usually information dense summaries of interpersonal situations written in the style of news headlines, e.g., *"Pushy Party Guests Make Themselves Too Much at Home."* We scrape all of the titles found in the archives, and use heuristics to attempt to filter out all posts that do not match this style, such as announcements and holiday greetings.

We attempt to balance the number of situations collected for each domain. However, we are limited by the complete set of examples from `dearabby` (12k).

[r/amitheasshole]
- telling my friend and her family to move out
- choosing to spend time with my friends or boyfriend rather than my family
- not wanting to hangout with sick girlfriend
- not wanting to do household chores
- banning my ex from my Spotify account

[r/confessions]
- My SO thinks I hate pickles, I like pickles but he LOVES pickles so I always pretend to hate them so he can have them.
- Best friend just got engaged.
- My girlfriend cheated and im cheating back on her
- I hate myself because I couldn't save my mother
- I'm scared of being a dad

[rocstories]
- Clark Ryder was proud of his job as a photojournalist.
- They had so many questions that I couldn't answer.
- Her husband surprised her on her birthday with plane tickets!
- She decided to wear slippers to protect her feet from Jason's toys.
- When he got to the assembled class he became very nervous.

[dearabby]
- Family of Six Tries Not to Be a Burden on Weekend Hosts
- Breakup Letter to Soldier Could Jeopardize Comrades in Arms
- Gentle Nudge Has Not Worked to Dislodge Mom From House
- Planning Helps Students Get Good Letter of Recommendation
- Man With Breast Cancer Experiences Extra Stress

Figure a.1: Five randomly sampled situations from each of the four domains we consider.

ADDITIONAL LABELS We allow annotators to mark each situation with any of the following labels that apply.

- **Unclear** The situation was too simple, vague, or confusing to understand what happened.
- **NSFW** The situation contains suggestive or adult content.
- **Dark / disturbing / controversial.** The situation contained content that may make folks uncomfortable, like suicide, torture, or abuse.

Annotators may pass on writing RoTs for a situation marked with any of those boxes, or they may still choose to do so. We keep all of the labels collected. They are included in the dataset as additional fields. For example, they could be used to omit certain training data to keep a model biased away from potentially controversial subjects.

A.1.2 *Character Identification*

Our goal during character identification is to find the most descriptive phrase referring to each unique non-narrator person in the passage exactly once.

The reason for this goal is that always having a single, best reference to each person in the situation enables more consistent grounding.

While this goal is relatively straightforward, we find many edge cases arise. In cases where it is unclear if a person should be marked, our central criteria is **whether someone might write RoTs involving that person**. If so, that person should be

included so they are a candidate for grounding. We found handling all of these edge cases complex enough to require human annotation instead of heuristics. We provide here the character identification guidelines that we give to the crowd worker annotators, along with an example illustrating each one.

CHARACTER IDENTIFICATION GUIDELINES

- **Don't include the (first person) narrator.** For example, *"I ate pizza"* would have no people highlighted.
- **Only include people.** For example, *"My horse George provides good conversation"* would have no people highlighted.
- **Only highlight each person once.** For example, *"I gave my brother a hug, I like him, he's so nice"*, we would only include my brother, not *"him"* or *"he."*
- **Highlight the most descriptive mention of a person.** For example, *"I can't stand him, my brother is so mean."*, we would pick my brother even though it comes after *"him."*
- **Include the full phrase referring to the person.** Include words like *"a"*, *"the"*, *"my"*, and longer phrases. For example, *"The strange guy talked to my brother and my oldest uncle,"* we would pick The strange guy, my brother, and my oldest uncle, instead of just *"guy"*, *"brother"*, and *"uncle."*
- **Don't include phrases where a generic person-looking word is used without referring to a particular person.** This often happens when describing a place or thing. For example, *"I walked into the men's room,"* we would not pick anything, because *"mens' room"* is a generic phrase. Similarly, we would not pick anything for, *"I am a child,"* because *"child"* is just used as a description. But for, *"I walked into my brother's room"*, we would pick my brother.
- **Include people used to refer to someone.** For example, *"**My brother's girlfriend** is so cool,"* we would pick both my brother and **my brother's girlfriend**.
- **Include pronouns (she, her, hers, etc.) *if* they're the most specific word available.** For example, in a sentence like *"I love him,"* we would pick him. However, for a sentence like, *"I love my brother, I can always talk to him."* we would instead pick my brother because it's more specific.
- **Include pronouns like *"they"* and *"them"*, also *if* they're the most specific word available.** For example, if we had the sentence *"They went to the party."* we would pick they. However, if we had the sentence *"My friends went to the party and they had a good time."* we would instead pick My friends since it is more specific.
- **Include plural first person pronouns (us, we, etc.) once.** For example, in a sentence like *"We went to the park."* we would pick we. Or for a sentence like *"They spent hours talking to us and we had a good time."* we would pick they and us.
- **Include other groups of people like *"her siblings," "their class,"* and *"his team."*** For example, in a sentence like *"I talked to all of **his uncles** for a while."* we would pick both his and **his uncles**.

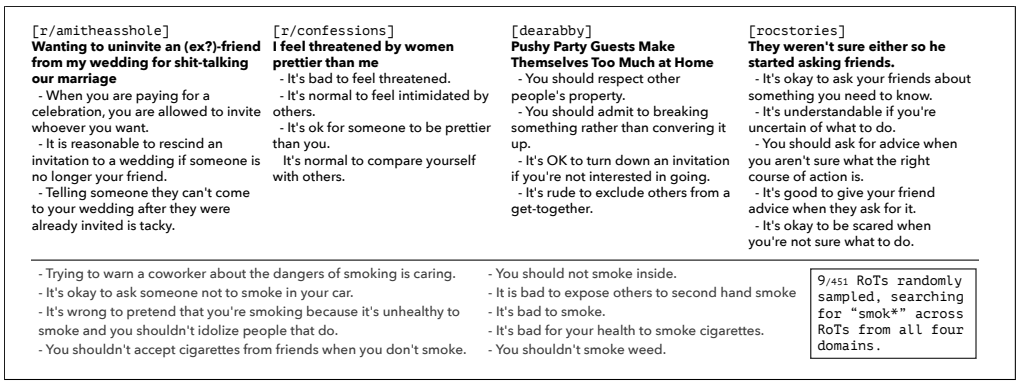


Figure a.2: **Top:** An example situation (bold) and corresponding RoTs (bullets) from each of the four domains we consider. **Bottom:** Random sample of RoTs about smoking, found by searching for *smok** across the dataset.

- **Include proper names of people that aren't the narrator.** For example, in a sentence like *"Mary chased John at the park."* we assume they are people (unless otherwise specified), and we would pick both Mary and John.
- **Include people with titles like *"the policeman"* and *"the mailman."*** For example, in the sentence *"I chased the store clerk."* we would select the store clerk.
- **Include words like *"someone"* and *"everyone."*** For example, in the sentence *"I am going to dinner with someone."* we would select someone.

A.1.3 *Rules-of-Thumb (RoTs)*

This section provides more information on how RoTs are written. Figure a.2 shows a sample of RoTs organized both by situation domain and topic.

As mentioned briefly in Section 3.2.2 of the paper body, we present workers with a series of guidelines for how to write RoTs. All RoT writing guidelines are in service of the goal that RoTs capture social, ethical, moral, and cultural norms. Unlike the guidelines for character identification, which are largely syntactic, the guidelines for writing RoTs are semantic. This makes them more challenging both to define and check.

To motivate these guidelines, and to help readers intuitively characterize what RoTs are, we present the RoT writing guidelines here at greater length, annotated with examples and explanations. For each guideline (**in bold**), we provide an example situation (*in italics*) along with candidate RoTs that **violate** or **follow** the guideline.

ROT WRITING GUIDELINES

- **Explain the basics of good and bad behavior.** RoTs should describe cultural expectations, as if to a file-year-old child who doesn't yet know how the world works.
  - Example situation: *Not wanting to take tests to apply for college*

  - **Violates:** "Studies have shown people perform best on tests after sleeping at least seven hours"
  - **Follows:** "It's normal to be stressed out by exams"
  - **Why:** This broad guideline attempts to distinguish RoTs from encyclopedic knowledge. Instead, RoTs should contain everyday, commonsense knowledge about social norms and expectations.

- **Judgment and action.** An RoT must comtain a judgment and an action.
  - Example situation: *Telling my husband he shouldn't buy his dream boat*
  - **Violates:** "Boats are expensive"
  - **Follows:** "It's mean to squash someone's dreams"
  - **Follows:** "People should be open to discussing big purchases with their spouses"
  - **Why:** Requiring an *action* helps ensure RoTs are about things peoeple do. Requiring a *judgment* pushes statements to contain some information about norms and expectations.

- **Self-contained.** An RoT must be fully understandable on its own, without the situation it came from.
  - Example situation: *Being angry at my sister for not attending our fathers funeral because of his criminal history.*
  - **Violates:** "It makes them feel bad"
  - **Violates:** "The father caused emotional distress to his daughter and the narrator should not judge her actions too harshly."
  - **Follows:** "If someone commits serious crimes, it's OK for family to cut off contact with them."
  - **Why:** Without this requirement, RoTs would not naturally generalize to new situations, and would trend too specific. The would also could contain much less information, as much of the semantic content could be left in the situation and only referred to by the RoT.

- **Inspired by situation.** An RoT should be inspired by the situation it came from.
  - Example situation: *Wanting to uninvite a friend from my wedding.*
  - **Violates:** "It's rude to point at people you don't know"
  - **Follows:** "It's devastating to be excluded from a wedding you were invited to"
  - **Why:** Maintaining a link between RoT and situation allows for grounding RoTs during the structured annotation. Furthermore, since a different worker will likely provides the structural annotation for an RoT, relevance to the source situation helps ensure the worker understands the RoT's context and implications.

- **Balance Specificity and Vagueness.** An RoT should be inspired by, and relevant to, the provided situation. However, a rule-of-thumb should also give a general

rule for how people behave in society, so should apply to more than just the given situation.

  - Example situation: *Not tipping my cashier last Tuesday*
  - **Violates:** "Not tipping a cashier last Tuesday is rude"
  - **Violates:** "It's rude to be cheap"
  - **Follows:** "It's usually OK not to tip cashiers in retail or grocery stores"
  - **Why:** This requirement can be the hardest to assess because of its subjectivity. RoTs that are too specific are usually slight modifications of the situation that include a judgment, and don't describe underlying expectations. RoTs that are too vague often do describe norms, but the link to the situation can be so distant as to be misleading. Good RoTs may be somewhat specific, but explain both the underlying norms at play, and apply to other situations.

- **Distinct ideas.** When multiple RoTs are provided for a situation, each should contain a distinct idea. This includes inversions of the same idea.
  - Example situation: *Never taking out the trash*
  - **Violates:** "It's irresponsible to avoid the chores you are assigned" *with* "It's bad to not do chores you're supposed to do"
  - **Violates:** "It's irresponsible to avoid the chores you are assigned" *with* "It's responsible to do the chores you are assigned"
  - **Follows:** "It's irresponsible to avoid the chores you are assigned" *with* "You're generally expected to have a clean home"
  - **Why:** This requirement is to prevent merely collecting paraphrases of the same RoT. Furthermore, we assume that inversions are usually trivial semantic mutations, so they are also not worth collecting at scale.

A.1.4 *RoT Breakdowns*

In this section, we provide more information about the structural RoT annotations, which we call "RoT Breakdowns." In particular, we illustrate the potential values for each attribute with an example.

A.1.5 *RoT Categorization*

RoT categories are originally designed to distinguish more desired annotation topics (morality/ethics, social norms) from less desired ones (advice and "it is what it is" statements). RoT categories are not mutually-exclusive, and the lines are not always clear. While we use all data regardless of RoT category in this paper's experiments, future work using this dataset may consider filtering based on RoT category. Annotators are allowed to select from none through all, but are encouraged to select the $1-2$ best.

Information

Type	Social
Annotated	RoT
Prompt	*What category best fits this RoT?*

Values

Label	Description	Example
Morality/ Ethics	Rules or guiding principles of right or wrong conduct	*It's wrong to turn your back on your friends who need your help.*
Social Norms	Standards of appropriate behavior in a society; emphasizes social conventions	*It's good to shake hands with your opponent even if you lost.*
Advice	Prudent behaviors that are aimed at improving one's life	*It's good to take medicine your doctor prescribes.*
It is what it is	Describes how things are; avoids ethics, morality or social norms	*It is nice to be tall.*

A.1.6 *Moral Foundations*

To simplify the annotation, we label *axes* of moral foundations. For example, *Care/Harm* are annotated together, rather than as *Care* and *Harm* separately. Other social attributes, such as *social judgment*, attempt to explicitly capture the goodness or badness of the act. Additionally, we omit the sixth moral foundation, *Liberty/Oppression*, which was later added to the formalism after the first five. Some examples are drawn from literature on Moral Foundations (Haidt, 2012). Note that these labels are not mutually exclusive, and annotators may choose from none through all.

Information

Type	Social
Annotated	RoT
Prompt	*Which moral foundations best fit this RoT?*

Values

Label	Description	Example
Care/Harm	Preventing or inflicting pain or suffering	*It's mean to tell someone they aren't attractive as someone else.*
Fairness/ Cheating	Notions of equity, justice, and rights	*It's wrong to copy someone else's answers in an exam.*
Loyalty/ Betrayal	Obligations or concerns for group, family, and nation.	*It's rude for a cheerleader to boo their own team.*
Authority/ Subversion	Submission and deference to traditions or legitimate authority.	*It's rude to walk away from your boss while they are talking to you.*
Sanctity/ Degradation	Abhorrence for disgusting things or actions; purity	*It's disgusting to pee into a public pool full of people.*

A.1.7 *Action and Judgment*

Each RoT should contain and action and a judgment. An essential step in the RoT brekadown is to identify the action so that it may be annotated with action attributes. We have annotators write both the action and the judgment, conjugating to the action to gerund tense, and rewriting when needed.

If the annotator is unable to clearly separate and action and a judgment from the RoT, they can simply mark the RoT as unclear and move on.

We provide here a few examples of RoTs along with corresponding actions and judgments. Some are straightforward, while others require more substantial rewriting.

- **RoT:** It's good to care for the elderly.
  - **Action:** caring for the elderly
  - **Judgment:** it's good
- **RoT:** It's okay to be angry if your friend talks to someone you used to date
  - **Action:** being angry when your friend talks to someone you used to date
  - **Judgment:** it's okay
- **RoT:** The holidays are expected to be especially difficult for those that are grieving.
  - **Action:** finding the holidays difficult while grieving
  - **Judgment:** expected

A.1.8 *Agency*

It can be challenging to distinguish *agency* from *experience* in cases where the action involves thinking thoughts or feeling emotions. We provide the following additional examples to workers for these cases, and allow their discretion during the annotation:

- **Experience:** Feeling upset when someone calls you a name
- **Agency:** Being mad for two days when someone calls you a name
- **Agency:** Taking revenge on someone for calling you a name

Information

Type	Social
Annotated	Action
Prompt	*Is the action $action something you do or control, or is it something you experience?*

Values

Label	Description	Example
Agency	You can do this	*Doing the dishes*
Experience	This happens to you; you have no control	*Being stuck in an earthquake*

A.1.9 *Social Judgment*

For social judgment, we ask workers to transcribe the intent of RoT's original judgment, rather than pick their own. Then, workers can mark their disagreement through their annotation of the *anticipated agreement* attribute. This allows us to faithfully capture controversial opinions, while also marking them as controversial.

Information

Type	Social
Annotated	Action
Prompt	*Which best matches the RoT's original judgment ($judgmentSpan) of $action ?*

Values

Label	Description	Example
Very bad	Spiteful, malicious, or insulting	*Slashing tires; Destroying someone's phone*
Bad	Frowned upon, rude, unkind, bad	*Littering; saying something mean to a friend*
Expected / OK	Normal; generally expected by society	*Wearing clothes; showing up to work on time*
Good	Kind of considerate	*Driving a friend to the airport*
Very good	Generous, self-sacrificing, perhaps even heroic	*Buying groceries for a financially struggling neighbor*

A.1.10 *Anticipated Agreement*

Anticipated agreement is theoretically continuously valued, compared to other attributes that are more naturally ordinal. Inspired by Wang, Durrett, and Erk (2018),

we bin this attribute into intuitive ranges, primarily to reduce cognitive load during annotation.

Information

Type	Social
Annotated	RoT, Action
Prompts	**RoT:** *What portion of people probably agree that $RoT ?*
	Action: *What portion of people probably agree that $action is $socialJudgment ?*

Values

	Label	Description	Example
	< 1%	Almost no one	*Murdering people is great*
	~ 5–25%	People occasionally think this.	*Ghosts exist*
	~ 50%	Controversial (people naturally disagree)	*It's okay to have an abortion*
	~ 75–90%	Common belief	*It's okay to try again when you fail*
	> 90%	Universally true	*It's bad to deliberately hurt your pets*

A.1.11 *Legality*

Annotations of legality are usually straightforward. Whether an action is illegal but tolerated can vary depending on location and knowledge of the law.

Information

Type	Social
Annotated	Action
Prompt	*Where you live, how legal is the action $action ?*

Values

	Label	Description	Example
	Illegal	Legal consequences if caught	*Theft; murder*
	Depends/ Tolerated	Generally "illegal", but often unenforced depending on circumstances	*Using a cellphone while driving*
	Legal	Not illegal	*Coughing without covering one's mouth*

A.1.12 *Cultural Pressure*

We provide instructions that cultural pressure could come from one's family, friends, community, culture, or society at large. We ask annotators to evaluate cultural pressure according to their own feelings.

Information

Type	Social
Annotated	Action
Prompt	*How much cultural pressure do you (or those you know) feel about $action ?*

Values

	Label	Description	Example
	Strong pressure against	Culture frowns upon this action	*Intentionally harming an animal*
	Pressure against	Culture generally discourages this action	*Spending money on jewelry if you can't afford it*
	Discretionary	Culture has little or nothing to say about this action	*Choosing to read before bed*
	Pressure for	Culture generally encourages this action	*Being honest with people*
	Strong pressure for	Culture strongly promotes this action	*Wearing clothes in public*

A.1.13 *Taking Action*

RoTs are written for a range of both hypothetical and actual actions related to the provided situation. Furthermore, sometimes the action is one that is explicitly not happening. This attribute labels how likely it is that the action is being taken by the relevant character. *Note: a subset of the* `r/AITA` *annotations were performed before the "probably not" label was introduced; for those, "hypothetical" is marked instead.*

Information

Type	Grounded
Annotated	Action
Prompt	*Is $candidateCharacter explicitly doing the action $action ? Or is it the action **might** happen?*

The upcoming examples use *narrator* and the following situation for context: *Not tipping the bartender at the club.*

Values

	Label	Description	Example
■	Explicitly not	It's explicitly written that they don't do this	*Tipping the bartender*
■	Probably not	Most likely not; they probably don't do this	*Enjoying the drinks*
■	Hypothetical	We can't say / no evidence	*Going clubbing every day*
■	Probable	Most likely; hints are written	*Paying for drinks*
■	Explicit	It's written in the situation	*Going to the club*

A.1.14 *Crowdsourcing*

Workers undergo an extensive vetting process before working on RoTs. This includes a paid qualification (qual) with a quiz on each of the guidelines and a manual review of sample RoTs. Workers then pass the qual move to a staging pool where they can work on a small number of situations, and all of their RoTs are manually reviewed for adherence to the guidelines. After graduating from the staging pool, workers enter the main group of RoT writers and annotators. For every batch of data, we perform spot checks on the RoTs written and annotated by the main group, as well as send feedback to all of the workers answering any questions we receive. We continuously update the instructions with clarifications, new examples, and answers to questions.

A.1.15 *Annotator Demographics*

With an extensive qualification process, 137 workers participated in our tasks. Of those, 55% were women and 45% men. 89% of workers identified as white, 7% as Black. 39% were in the 30-39 age range, 27% in the 21-29 and 19% in the 40-49 age ranges. A majority (53%) of workers were single, and 35% were married. 47% of workers considered themselves as middle class, and 41% working class. In terms of education level, 44% had a bachelor's degree, 36% some college experience or an associates degree. Two-thirds (63%) of workers had no children, and most lived in a single (25%) or two-person (31%) household. Half (48%) our workers lived in a suburban setting, the remaining half was evenly split between rural and urban. Almost all (94%) of our workers had spent 10 or more years in the U.S.

A.1.16 *Demographics and Annotations*

We analyze the demographic variation in RoT and action annotations, using a set of 400 RoTs that were annotated by 50 workers each. In addition to the demographic variables described in §a.1.15, we also consider the political leaning of the state in which the worker resides (self-reported), by assigning each state a value based on the state-level

	RoT Agreement	Action Agreement	Cultural Pressure	Social Judgment
Gender (M: 0, F: 1)	0.070***	0.104***	*n.s.*	*n.s.*
Urbanness	0.065***	0.085***	*n.s.*	*n.s.*
Education	0.022**	0.037***	*n.s.*	0.025***
Politics (rep: 0, dem: 1)	0.052***	0.075***	0.023**	*n.s.*
Household size	0.059***	0.080***	*n.s.*	*n.s.*
Social class	*n.s.*	*n.s.*	*n.s.*	*n.s.*
Income	-0.027*	*n.s.*	*n.s.*	*n.s.*
Age	*n.s.*	*n.s.*	*n.s.*	*n.s.*

Table a.1: Correlations between worker demographics and categorical RoT annotations, Bonferroni corrected for multiple comparisons ($p < 0.0001$: ***, $p < 0.001$: **, $p < 0.01$: *).

voting patterns in the last four national elections (yielding five-point scale from 100% republican to 100% democratic).

For our analyses, we run a generalized linear model regressing the RoT categories on all z-scored demographic variables, and report the β coefficients from that model. In our action moral judgment analyses, we control for actions; for action agreement, we control for the action and the moral judgement; for the RoT agreement and action pressure, we control for individual RoTs. Our results for categorical RoT annotations are shown in Table a.1.

AGREEMENT (ROT AND ACTION) The projection of how many people agree with the judgement is correlated with various demographic characteristics. Specifically, judgments of actions, being a woman and living in an urban setting was most strongly correlated with ascribing high agreement to the judgment. Other associations include higher education, household size, and inferred political leaning based on state of residency.

For RoT agreement, we find similar but weaker associations. Additionally, we find a small correlation between income and social class and ascribing higher agreement.

CULTURAL PRESSURE The only variable correlated with feeling culturally pressured is the political leaning of the state where workers are located, though the effect is small.

SOCIAL JUDGMENT Similar to action agreement. Effects are somewhat weaker, but workers being women, highly educated, or younger are associated with selecting higher (better) judgment to actions.

A.2 EXPERIMENTAL DETAILS

GENERATIVE MODELS We use the Transformers package (Wolf et al., 2019) to implement our models. We train all the models for a single epoch with a batch size of 64, with the random seed 42. Each input and output sequence is prefixed with a special token indicating its type (e.g. `[attrs]`, `[rot]`, `[action]`). We also define a special token for each attribute value (e.g. `<morality-ethics>`, `<bad>`, `<all>`, `<against>`). We initialize the special token embeddings with the embedding of their corresponding words, taking the average for multiword expressions. For example, $\vec{v}_{\text{<bad>}} = \vec{v}_{\text{bad}}$, $\vec{v}_{\text{<morality-ethics>}} = (\vec{v}_{\text{morality}} + \vec{v}_{\text{ethics}})/2$.

b

APPENDIX MATERIAL FOR NEURAL NATURALIST

B.1 ALGORITHMIC APPROACH TO DATASET CONSTRUCTION

We present here an algorithmic approach to collecting a dataset of image pairs with natural language text describing their differences. The central challenge is to balance empirical desiderata—mainly, sample coverage and model relevance—with practical constraints of data quality and cost. This algorithmic approach underpins the dataset collection we outlined in the chapter body.

B.1.1 *Goals*

Our goal is to collect a dataset of tuples (i_1, i_2, t), where i_1 and i_2 are images, and t is a textual comparison of them. We can consider each image i as drawn from some domain $\mathcal{D} \in \{\textit{furniture, trees, ...}\}$, or a completely open domain of all concepts. There are several criteria we would like to balance:

1. **Coverage** A dataset should sufficiently cover $\mathcal{D}$ so that generalization across the space is possible.
2. **Relevance** Given the capabilities for models to distinguish i_1 and i_2, t should provide value.
3. **Comparability** Each pair $(i_1,\ i_2)$ must have sufficient structural similarities that a human annotator can reasonably write t comparing them. Pairs that are too different will yield lengthy and uninteresting descriptions without direct contrasting statements. Pairs that are too similar for human perception may yield *"I can't see any difference."*[1]
4. **Efficiency** Image judgements and textual annotations require human labor. With a fixed budget, we would like to yield a dataset of the largest size possible.

We describe sampling algorithms for addressing these issues given the choice of a domain.

1 This hints at the same sweet spot the fine-grained visual classification (FGVC) community studies, like cars (Krause et al., 2013b), aircraft (Maji et al., 2013), dogs (Khosla et al., 2011), and birds (Van Horn et al., 2018; Wah et al., 2011).

B.1.2 *Pivot-Branch Sampling*

Drawing a single image i from domain $\mathcal{D}$, there is a chance $p \in [0, 1]$ that each image is ill-suited for comparisons. For example, i might be out-of-focus or contain multiple instances.

If a pair of images is drawn, and each has probability p of being discarded, then $\frac{1}{(1-p)^2}$ times more pairs must be selected and annotated. For example, if $p = \frac{2}{3}$, then the annotation cost is scaled by 2.25. This severely impacts annotation *efficiency*.

To combat this, we employ a stratified sampling strategy we call *pivot-branch sampling*. Each image on one side of the comparison (say, i_{pivot}) is vetted individually, and k images on the other side (say, i_{branch}) are sampled to produce pairs. With k-times fewer i_{pivot} images, it is feasible to check each instance for usability. This lowers the annotation cost scale to $\frac{1}{1-p}$ (e.g., with $p = \frac{2}{3}$, this is 1.5).

Splitting our selection from $\mathcal{D}$ into two parts allows us to define two distinct sampling strategies. One choice is for $s_{\text{pivot}}(\mathcal{D})$ to select pivot images. The second is for $s_{\text{branch}}(\mathcal{D}, i_{\text{pivot}}, k)$ to sample k images given a single pivot image.

B.1.3 *Designing $s_{pivot}(\mathcal{D})$*

Selecting i_{pivot} are important because each will contribute to k image pairs in a dataset. Here we consider the case where there are class labels $c \in \mathcal{C}$ available for each image in the domain. We propose selecting s_{pivot} to sample uniformly over $\mathcal{C}$. This strategy attempts to provide coverage over $\mathcal{D}$ using class labels as a coarse measure of diversity. It accounts for category-level dataset bias (e.g., where most images belong to only a few classes). This pushes the need to address *relevance* and *comparability* to the sampling procedure for branched images.

B.1.4 *Designing $s_{branch}(\mathcal{D}, i_{pivot}, k)$*

Given each pivot image i_{pivot}, we will choose k images from $\mathcal{D}$ for comparison. We can make use of additional functions and structure available on $\mathcal{D}$:

$\mathcal{V}(i_1, i_2) \longrightarrow [0, 1]$
A function that measures the visual similarity between any two images.

$\mathcal{T}(\mathcal{D})$
A taxonomy over $\mathcal{D}$, with image class labels $c \in \mathcal{C}$ as leaves.

We can partition $k = k_v + k_t$ to sample k_v visually-similar images using and k_t taxonomically related images. A simple strategy for visually similar images is to pick

$$\underset{i' \in D, i' \neq i_{\text{pivot}}}{\operatorname{argmin}} \; \mathcal{V}(i_{\text{pivot}}, i')$$

k_v times without replacement. This samples the k_v most visually similar images to i_{pivot}, excluding the image itself.

To employ taxonomic information, we propose a walk over mutually exclusive subsets of $\mathcal{T}(\mathcal{D})$. We define a function $a_{\mathcal{T}(\mathcal{D})}(c, \ell)$ that gives the set of other taxonomic leaves that share a common ancestor exactly ℓ taxonomic levels above c, and no levels lower. More formally, if we use $p(c, c', \ell)$ to express that c and c' share a parent ℓ taxonomic levels above c, then we can define:

$$a_{\mathcal{T}(\mathcal{D})}(c, \ell) = \{c' \,:\, p(c, c', \ell) \wedge \nexists_{\ell' < \ell} \, p(c, c', \ell')\}$$

The function $a_{\mathcal{T}(\mathcal{D})}(c, \ell)$ partitions the taxonomy $\mathcal{T}(\mathcal{D})$ into disjoint subtrees. For example, $a_{\mathcal{T}(\mathcal{D})}(c, 1)$ are the set of sibling classes to c which share its direct parent; $a_{\mathcal{T}(\mathcal{D})}(c, 2)$ are the set of cousin classes to c which share its grandparent, but *not* its parent.

We can employ $a_{\mathcal{T}(\mathcal{D})}(c, \ell)$ by choosing class c from our pivot image i_{pivot} and varying ℓ. As we increase ℓ, we define mutually exclusive sets of classes with greater taxonomic distance from c.

To sample images using this scheme, we can further split our k_t budget for taxonomically sampled images into $k_t = k_{t_1} + k_{t_2} + \cdots + k_{t_\ell}$ for ℓ different levels. Then, if we write the set of classes $C_\ell = a_{\mathcal{T}(\mathcal{D})}(c, \ell)$, we can sample k_{t_ℓ} images from C. One scheme is to perform round-robin sampling: rotate through each class $c_\ell \in C_\ell$ and sample sample one image from each until k_{t_ℓ} are chosen.

B.1.5 *Analyzing* $s_{branch}(\mathcal{D}, i_{pivot}, k)$

Given a good visual similarity function $\mathcal{V}$, image pairs will exhibit enough similarity to satisfy requirement that they be semantically close enough to be *comparable*. They may also be so visually similar that comparability is difficult. However, this aspect counter-balances with *relevance*: if $\mathcal{V}(i_1, i_2)$ is small under a visual model, but their differences are describable by humans, their difference description has high value because it distinguishes two points with high similarity in visual embeddings space.

The use of the taxonomy $\mathcal{T}(\mathcal{D})$ complements $\mathcal{V}$ by providing controllable *coverage* over $\mathcal{D}$ while maintaining *relevance* and *comparability*. Tuning the range of ℓ values used in the taxonomic splits $a_{\mathcal{T}(\mathcal{D})}(c, \ell)$ ensures comparability is maintained. Clamping ℓ below a threshold ensures images have sufficient similarity, and controlling the proportion of k_{t_ℓ} for small values of ℓ mitigates the risk of too-similar image pairs.

Similarly, we can adjust the relevance of taxonomic sampling by controlling the distribution of $k_{t_1} \ldots k_{t_\ell}$ with respect to the particular structure of the taxonomy $\mathcal{T}(\mathcal{D})$. If the taxonomy is well-balanced, then fixing a constant k_{t_ℓ} will draw proportionally more samples from subtrees close to c. This can be seen by considering that $a_{\mathcal{T}(\mathcal{D})}(c, \ell)$ defines exponentially larger subsets of $\mathcal{T}(\mathcal{D})$ as ℓ increases. Drawing the same number

of samples from each subset biases the collection towards relevant pairs (which should be more difficult to distinguish) while maintaining sparse coverage over the entirety of $\mathcal{D}$.

B.2 DETAILS FOR CONSTRUCTING BIRDS-TO-WORDS DATASET

We provide here additional details for constructing the Birds-to-Words dataset. This is meant to link the high level overview in Section 4.1 with the algorithmic approach presented in the previous section (Appendix b.1).

B.2.1 *Clarity*

To build a dataset emphasizing fine-grained comparisons between two animals, we impose stricter restrictions on the images than iNaturalist research-grade observations (photographs). An iNaturalist observation that is research-grade indicates the community has reached consensus on the animal's species, that the photo was taken in the wild, and several other qualifications.[2] We include four additional criteria that we define together as *clarity*:

1. **Single instance**: A photo must include only a single instance of the target species. Bird photography often includes flocks in trees, in the air, or on land. In addition, some birds appear in male/female pairs. For our dataset, all of those photos must be discarded.
2. **Animal**: A photo must include the animal itself, rather than a record of it (e.g., tracks).
3. **Focus**: A photo must be sufficiently in-focus to describe the animal in detail.
4. **Visibility**: The animal in the photo must not be too obscured by the environment, and must take up enough pixels in the photo to be clearly described.

B.2.2 *Pivot Images*

To pick pivot images, we first uniformly sample from the set of 9k species in the taxonomic CLASS *Aves* in iNaturalist. We consider only species with at least four recorded observations to promote the likelihood that at least one image is *clear*. We also perform look-ahead branch sampling to ensure that a species will yield sufficient comparisons taxonomically. For each species, we manually review four images sampled from this species to select the clearest image to use as the pivot image. If none are suitable, we move to the next species. With this manual process, we select 405 species and corresponding photographs to use as pivot i_1 images.

2 More details on iNaturalist research-grade specification: https://www.inaturalist.org/pages/help#quality

B.2.3 *Branching Images*

See Section 4.1.3 for the description of selecting $k_v = 2$ visually similar branching images using a function $\mathcal{V}(i_1, i_2)$. We highlight here the use of the taxonomy $\mathcal{T}(\mathcal{D})$ to select $k_t = 10$ branching images with varying levels of taxonomic distance.

For the class c corresponding to image i_1, we split the taxonomic tree into *disjoint* subtrees rooted $\ell \in \{1..5\}$ taxonomic levels above c. Each higher level *excludes* the levels beneath it. For example, at $\ell = 1$ we consider all images of the same species as i_1; at $\ell = 2$, we consider all images of the same genus as i_1, but that have a *different* species. We set each $k_{t_\ell} = 2$ for a total of $k_t = 10$.

B.2.4 *Annotations*

CLARITY Annotators first label whether i_1 and i_2 are *clear*. While we manually verified each i_1 is clear, each i_2 must still be vetted.[3] Starting from 405 pivot images i_1, and selecting $k = 12$ branching images i_2 for each, we annotated a total of 4,860 image pairs. After restricting images to have $\geq \frac{4}{5}$ positive clarity judgments, we ended up with the 3,347 image pairs in our dataset, a retention rate of 68.9%.

QUALITY We vet each annotator individually by manually reviewing five reference annotations from a pilot round, and perform random quality assessments during data collection. We found that manually vetting the writing quality and guideline adherence of each individual annotator vital for ensuring high data quality.

B.3 MODEL DETAILS

For the image embedding component of our model, we use a ResNet-101 network as our CNN. We use a model pretrained on ImageNet and fix the CNN weights before starting training for our task. We also experimented with an Inception-v4 model, but found ResNet-101 to have better performance.

For both the Transformer encoder and decoder, we use $N = 6$ layers, a hidden size of 512, 8 attention heads, and dot product self-attention. Each paragraphs is clipped at 64 tokens during training (chosen empirically to cover 94% of paragraphs). The text is preprocessed using standard techniques (tokenization, lowercasing), and we replace mentions referring to each image with special tokens ANIMAL1 and ANIMAL2.

For inference, we experiment with greedy decoding, multinomial sampling, and beam search. Beam search performs best, so we use it with a beam size of 5 for all reported results (except the decoding ablations, where we report each).

We train with Adagrad for 700k steps using a learning rate of .01 and batch size of 2048. We decay the learning rate after 20k steps by a factor of 0.9. Gradients are clipped at a magnitude of 5.

3 Annotators would occasionally agree that a particular i_1 images was in fact unclear, upon which we removed it and all corresponding pairs from the dataset.

Photograph	Attribution
Fig. 4.1: Top and bottom left	salticidude *(CC BY-NC 4.0)* https://www.inaturalist.org/observations/20863620
Fig. 4.1: Top right	Patricia Simpson *(CC BY-NC 4.0)* https://www.inaturalist.org/observations/1032161
Fig. 4.1: Bottom right	kalamurphyking *(CC BY-NC-ND 4.0)* https://www.inaturalist.org/observations/9376125
Fig. 4.2: Top left	Ryan Schain https://macaulaylibrary.org/asset/58977041
Fig. 4.2: Top right	Anonymous eBirder https://www.allaboutbirds.org/guide/Song_Sparrow/media-browser/66116721
Fig. 4.2: Right, 2nd from top	Garth McElroy/VIREO https://www.audubon.org/field-guide/bird/song-sparrow#photo3
Fig. 4.2: Right, 3rd from top	Myron Tay http://orientalbirdimages.org/search.php?Bird_ID=2104&Bird_Image_ID=61509&p=73
Fig. 4.2: Right, 4th from top	Brian Kushner https://www.audubon.org/field-guide/bird/blue-jay
Fig. 4.2: Bottom, left	A. Shcherbakov
Fig. 4.2: Bottom, right	prepa3tgz-11bwv518 *(CC BY-NC 4.0)* https://www.inaturalist.org/observations/23184228
Fig. 4.4: Top	jmaley *(CC0 1.0)* https://www.inaturalist.org/observations/31619615
Fig. 4.4: Bottom	lorospericos *(CC BY-NC 4.0)* https://www.inaturalist.org/observations/30605775
Fig. 4.5: Top left, left	wildlife-naturalists *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/13223248
Fig. 4.5: Top left, right	Colin Barrows *(CC BY-NC-SA 4.0)* https://www.inaturalist.org/photos/2642277
Fig. 4.5: Top middle, left	charley *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/13379419
Fig. 4.5: Top middle, right	guyincognito *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/26314681
Fig. 4.5: Top right, left	Chris van Swaay *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/18941543
Fig. 4.5: Top right, right	Jonathan Campbell *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/20120523
Fig. 4.5: Middle left, left	John Ratzlaff *(CC BY-NC-ND 4.0)* https://www.inaturalist.org/photos/647514
Fig. 4.5: Middle left, right	Jessica *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/5595152
Fig. 4.5: Middle middle, left	i_c_riddell *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/1331149
Fig. 4.5: Middle middle, right	Pronoy Baidya *(CC BY-NC-ND 4.0)* https://www.inaturalist.org/photos/5027691
Fig. 4.5: Middle right, left	Nicolas Olejnik *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/2006632
Fig. 4.5: Middle right, right	Carmelo López Abad *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/892048
Fig. 4.5: Bottom left, left	Luis Querido *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/13052253
Fig. 4.5: Bottom left, right	copper *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/22043211
Fig. 4.5: Bottom middle, left	vireolanius *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/13550702
Fig. 4.5: Bottom middle, right	Mathias D'haen *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/14943695
Fig. 4.5: Bottom right, left	tas47 *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/10691998
Fig. 4.5: Bottom right, right	Nik Borrow *(CC BY-NC 4.0)* https://www.inaturalist.org/photos/13776993

B.4 IMAGE ATTRIBUTIONS

The table above provides attributions for all photographs used in the Neural Naturalist chapter.

C

APPENDIX MATERIAL FOR SCARECROW

C.1 SCARECROW ANNOTATION SCHEMA

Here, we present in greater detail the SCARECROW annotation span types.[1] A visual summary is shown in Figure c.1.

While we annotate using this schema, the essence of our study is to embrace language users' abilities to detect when something may be wrong with text. In other words, we do not wish for our span definitions to get in the way of humans describing problems with text. To this end, we encourage researchers to embrace label back off (to coarser categories), merging labels (based on empirical observations), and refining the annotation ontology over time. The central goal is to collect *what people find wrong with text.*

C.1.1 *Language Errors*

We define five categories of *language errors*, which concern the selection of ideas in a text and how they are expressed. These range from grammar and syntax problems to issues of semantics and pragmatics.

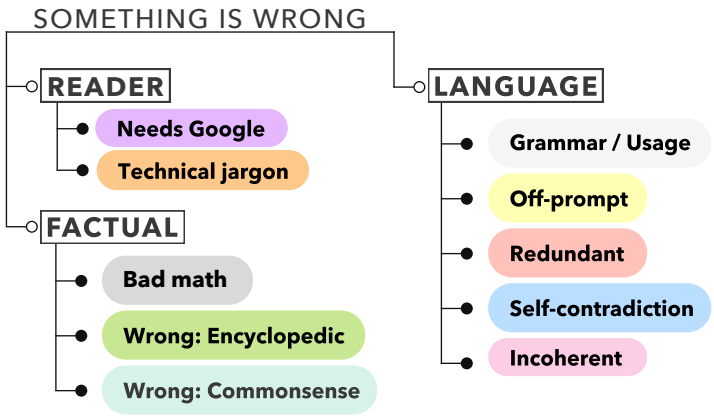


Figure c.1: A visualization of SCARECROW spans: three categories (reader, language, and factual) composed of ten types. Annotators choose directly from the ten span types.

C.1.1.1 ***Grammar and Usage***

This category of errors includes missing words, extra words, and incorrect or out of order words.

1 All example annotations here are our own. Many are provided to annotators during training.

EXAMPLE
A PhD student from the University of Kent in the UK claims to have discovered a clever way to explain the positive **emoticons** in cats.

Explanation: *The word should probably be "emotions."*

We also label **Grammar and Usage** for inserted words or small phrases that could be deleted to resolve the issue:

A couple is facing criticism for their extravagant birthday party. The bewitching pair had first stripped down to fishnets **and backward.**

Explanation: *This phrase can simply be deleted.*

We avoid partitioning **Grammar and Usage** errors into more detailed categories based on the observation that large language models produce fewer issues of syntax and diction (aside from **Redundant** errors, described next). As such, we focus instead on semantic and pragmatic errors, captured by the upcoming span types.

C.1.1.2 *Redundant*

While "redundant" can also include extra unnecessary information, we specifically use the **Redundant** label to mark repetition. In identifying redundant text, our schema annotates both the **antecedent** (first mention) and the **redundant text** (when the repetition occurs). Sometimes the exact word or phrase will be repeated.

EXAMPLE
Many merchants worry about the possibility of **poor service** **or service** for certain categories of customers.

Other times, generated text expresses the same idea repeatedly using different words.

EXAMPLE
They then made decisions based on Kondo's instructions, to the extent that they **created de-cluttered spaces** **and got rid of clutter and clutter-filled spaces**.

C.1.1.3 *Off-Prompt*

The prompt is a human-written sentence used as context from which the model generates a continuation. Models sometimes generate text that is unrelated to the prompt.

EXAMPLE
Prompt: Dogs are the new kids.
Generation: Statistics suggest that most Americans would be happier with dogs than children. **In fact, four out of five don't even visit the dentist annually, much less every six months.** Dog owners report much higher rates of happiness than non-dog owners.

Other times, the text may be related, but it contradicts what is stated in the prompt.

EXAMPLE
Prompt: China sets new record for Economic Growth
Generation: The Chinese economy **fell 10% this month, the third such loss this year.**

C.1.1.4 ***Self-Contradiction***

When a model generates text that contradicts the prompt, that is labeled as **Off-Prompt**. But when a model generates text that contradicts *itself*, that is labeled as **Self-Contradiction**. We also mark the **antecedent** (original statement).

> **EXAMPLE**
> McDonald's is considering a design which will replace the **cardboard packaging.** Mr Gore-Cotter said: "We recognise the concern around waste. We are now looking at a new design that minimises the **plastic bag."**
>
> ***Explanation:*** *The idea of minimizing the plastic bag contradicts the stated goal of replacing cardboard packaging.*

> **EXAMPLE**
> Mall of America plans to **lay off and furlough hundreds of its employees. It has no plans to restrict the number of hours workers can work.**
>
> ***Explanation:*** *Furloughed workers are explicitly restricted from working.*

C.1.1.5 ***Incoherent***

Generated text is sometimes grammatical, not redundant, on prompt, and not contradictory, but still confusing. We provide the **Incoherent** label for such sentences.

> **EXAMPLE**
> Melody Mitsugi, 28, had never given her kids cheese toast before her husband **drew a map of it on her toast.**
>
> ***Explanation:*** *One can't exactly draw a map of Cheese Toast, and one probably wouldn't draw it on toast itself.*

> **EXAMPLE**
> Cats naturally show anxiety and fear by at times **breaking apart different parts of the brain in an attempt to keep the others from escaping.**
>
> ***Explanation:*** *It's difficult to even imagine what is happening in this passage.*

C.1.2 *Factual Errors*

We define three categories of factual errors, which encompass known incorrect statements.

C.1.2.1 ***Bad Math***

Generated text will sometimes have issues with basic mathematical operations of known quantities (e.g., *"half of ten apples is four"*), problems converting fixed units (e.g., *m to cm*).

> **EXAMPLE**
> One account, @Iain_Rowling1, had over 500,000 followers at one point, but in just four days they fell by **around half - some 4,000.**

We also include problems converting currencies that are wildly implausible under modern assumptions (e.g., *$1 US = £10*).

EXAMPLE
... compared with just over £1,000 **($18,868)** for previous versions of Samsung's flagship phone.

C.1.2.2 *Commonsense*

These errors mark spans that violate our everyday basic understanding of the world. Though it is challenging to precisely define *commonsense knowledge* (Liu and Singh, 2004), we include non-encyclopedic knowledge and basic reasoning.

The following example concerns broadly sensible numerical ranges.

EXAMPLE
The picture is from high above the South Pole, where close to **100,000** Astronauts live and work.

Explanation: *Even if we don't know the exact number of astronauts in space, it is common knowledge that 100k is far too many.*

The next example involves world knowledge, akin to scripts (Schank and Abelson, 1977).

EXAMPLE
You can get the dress custom-made and stitched at your favorite **spa.**

Explanation: *Spas don't offer stitching.*

The following example involves lexical entailment.

EXAMPLE
The thinness of our bodies isn't an answer to all common human health problems like **obesity** or diabetes

Explanation: *While most of the statement is acceptable, it's impossible to be "thin" and "obese" at the same time.*

The final example involves time.

EXAMPLE
Now in 2021, NASA is measuring California wildfire temperatures using an instrument on the International Space Station. This year's record-shattering heat has had global repercussions in **2017**, forcing sea level rise on California and increasing the risk of deadly wildfires.

Explanation: *Events in 2021 can't affect events in 2017.*

C.1.2.3 *Encyclopedic*

These errors are ones that we *know* are factually wrong, and that we could look up in, say, Wikipedia.

EXAMPLE
Japanese Prime Minister Justin Trudeau said he will be halting all imports and exports until the current situation can be contained.

Explanation: *Justin Trudeau is the Prime Minister of Canada, not Japan.*

The distinction between **Encyclopedic**, and the upcoming **Technical Jargon** and **Needs Google** errors, depend on the reader's knowledge.

EXAMPLE
The gas contains something known as **phyto-romatic acid, a common chemical element in the periodic table.**

Explanation: *Acids aren't elements.*

C.1.3 *Reader Issues*

We define two categories of reader issues. These are words or statements a reader cannot verify without using an external resource.

C.1.3.1 ***Technical Jargon***

Sometimes generated text includes specific words from a field that requires expertise to understand.

EXAMPLE
In Chile, an 800-megawatt **photovoltaic** plant was built for a record low cost of $129 per megawatt-hour last year.

Which words are jargon depends on the reader's particular expertise. This means **Technical Jargon** spans are more accurately thought of as *potential issues* rather than known errors.

EXAMPLE
He uses a spirit **mash** made from white corn and malted barley and a **neutral grain**, which he describes as a "whiskey grain."

C.1.3.2 ***Needs Google***

Many facts—especially those involving specific people, events, dates, or numbers—could be categorized as encyclopedic knowledge. However, whether the fact is accurate may require additional verification by the everyday reader. To make this distinction between *known* encyclopedic knowledge and *trivia*, we introduce this label to denote that a reader would need to search online to verify whether it is true.

We instruct annotators to *not* look up facts marked with the **Needs Google** span. We do this to keep the focus of the task on classification, rather than factuality detection. As a result, **Needs Google** spans mark *statements that would need to be verified*, rather than known errors.

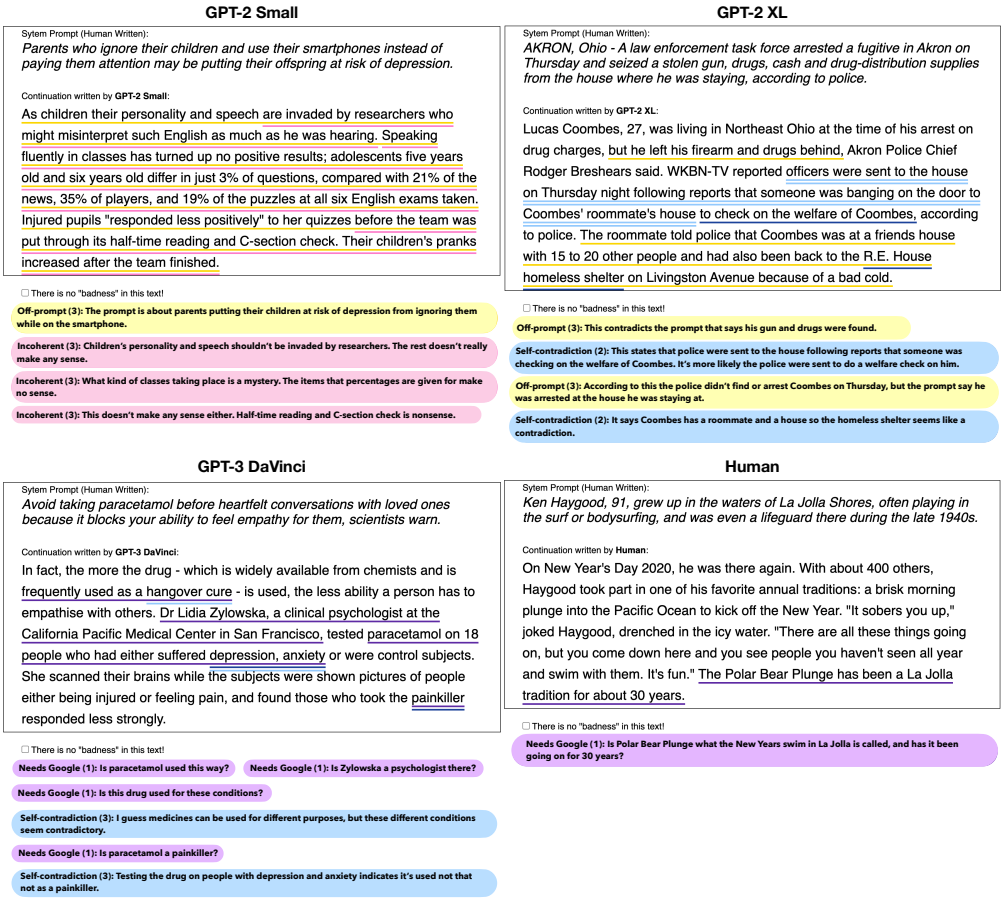


Figure c.2: Example SCARECROW annotations of three model generations and one ground truth continuation, demonstrating the shift in number, type, and severity of errors. The entirety of the **GPT-2 Small** generation is **Off-Prompt** and/or **Incoherent**, with high severity (3/3). **GPT-2 XL** is instead only about two-thirds covered by errors—still sometimes **Off-Prompt**, but also **Self-Contradiction**, and with high severity (2–3/3). In contrast, **GPT-3 DaVinci** receives several **Needs Google** marks—less severe than errors, as they only indicate that fact-checking is needed—though it also commits two high-severity **Self-Contradiction** errors by generating inconsistent claims. The **Human** (ground-truth) continuation only receives one **Needs Google** span.

EXAMPLE

It was promoted by **Dr. Michael Fanning, the Executive Director of the Foundation for Mental Health Awareness, Inc.**

Explanation: *A reader would likely need to look up whether there is a Dr. Fanning who holds this position.*

EXAMPLE

... an **800-megawatt photovoltaic plant** was built for a **record low cost of $129 per megawatt-hour** last year.

Explanation: *In addition to potential* ***Technical Jargon*** *spans, there are at least*

two ***Needs Google*** *spans: 1. whether such a plant can be roughly 800-megawatt, 2. whether $129/megawatt-hour is a sensible cost measure, and the value is reasonable.*

To illustrate the annotation methodology and schema in practice, we present four complete example annotations in Figure c.2. This figure also illustrates how much variation we see across models.

C.2 ANNOTATION DETAILS

C.2.1 *Error Severity*

We provide here examples for each of the three error severity levels, which we also give to annotators during training.

EXAMPLE
Paul Campbell-Hughes, from the University of Aberdeen, explains how **she** managed to locate colonies of honey bees in Kent.

Severity: 1. *Since Paul is usually a male name, the model should have used "he." But this error is pretty minor.*

EXAMPLE
Paul Campbell-Smith, a PhD student from the University of Kent in the UK, claims to have discovered a clever way to explain the positive **emoticons** in cats.

Severity: 2. *The word should probably be "emotions." We can guess what was being said, but it's definitely wrong.*

EXAMPLE
Prompt: Whether you're on Facebook, Instagram, Snapchat or TikTok, many people make huge efforts to curate the best version of themselves online.
Generation: **This year we've got something for you: a Love Match Custom Size Poster featuring Mather, Phoenix, Kashun and all her friends, divided among six different covers, creating a beautiful custom size poster for your own personal high school reunion.**

Severity: 3. *Even ignoring the end of the generation (a poster for a personal high school reunion?), this whole generation is way off the prompt and does not make sense.*

C.2.2 *Crowdsourcing Details*

Our annotation process requires significant training and time investment during annotation. We use Amazon Mechanical Turk (AMT) for all data collection.

TRAINING For training, we first pay each worker $40 to take an extensive qualification task, which both trains them in the span categorization scheme and quizzes their understanding. After initial training on selecting spans and marking error severity, we train workers to identify each type of error. For each error type, we provide workers with an English description, examples, an exercise where they must mark the span, and

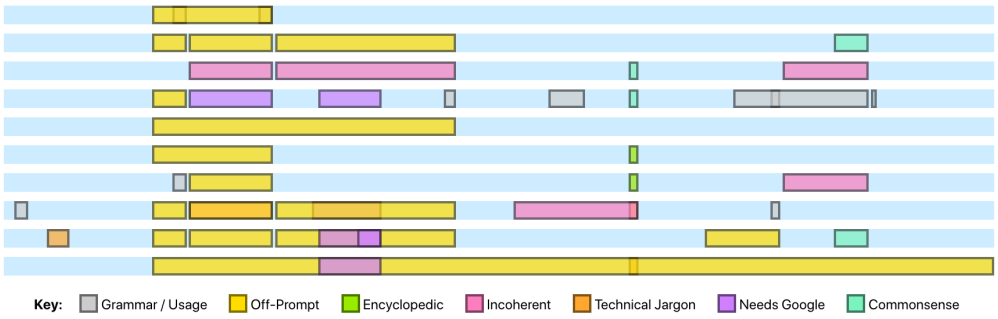


Figure c.3: A visual representation of the 10 annotations we collected for one paragraph. Each blue bar represents one annotator, where the width of the bar represents the text of the paragraph. Colored bars drawn on top of the blue bar represent spans marked as errors. We draw bars semi-transparently to show overlapping errors. We can see that some problematic spans (e.g., the **Off-Prompt** section) are marked by almost all workers and given the same label. Other spans are marked by only a subset of the workers (e.g., **Commonsense** and **Incoherent** spans on the right side), or have some label disagreement.

a quiz to classify the correct error type in pre-marked spans. After training all ten error categories, workers attempt a real annotation task where they annotate a paragraph using the full annotation tool. We pass workers if they score above 90% on the quiz questions, and produce good annotations using the full tool (via manual inspection). This training is available at `https://yao-dou.github.io/qual/`.

ANNOTATION For the annotation task, workers annotate each paragraph using a custom annotation interface (shown partially in Figure 5.5), for which we pay $3.50. We calculated $3.50 per annotation by aiming to pay workers at least $15/hour. After several annotation rounds, we observed considerable variation in time per annotation, so this cost should not be necessarily seen as a requirement for SCARECROW annotations.

C.3 DATA QUALITY

Identifying and classifying errors in potentially noisy machine generated text is a challenging task. How consistent are the annotations collected from crowd workers? In this section, we examine the agreement and variability of the collected annotations.

At a high level, we observe either acceptable or high inter-annotator agreement across error categories. For rare error types such as **Bad Math**, high agreement stems from the prevalence of spans with no error. For such categories, *we recommend treating each annotator as a low-recall, high precision judge*, and considering the information from their aggregate annotations. Figure c.3 gives an example of the perspective gained by viewing all 10 annotations of a single generation.

AGREEMENT Table c.1 shows token-level inter-annotator agreement statistics aggregated over all collected data. Since a single annotator can label a single span with

Error	Krippendorff's α	*Two Agree* (%)
Bad Math	0.99	30
Commonsense	0.88	20
Encyclopedic	0.98	12
Grammar and Usage $^{>1}$	0.72	30
Incoherent	0.73	49
Off-Prompt	0.71	61
Redundant	0.88	38
Self-Contradiction	0.87	26

Table c.1: Per-token inter-annotator agreement metrics by error category. The $^{>1}$ indicates that we omit severity-1 **Grammar and Usage** errors in all analyses in this chapter due to higher variance; including them would drop the Krippendorf's α to 0.56.

multiple errors, we break the agreement statistics down by error category. We report Krippendorff's α coefficient, a chance-corrected measure of agreement for multiple annotators (Krippendorff, 2018). Due to computational constraints, we calculate this coefficient per generation and report the average across the dataset. The agreement shown here is high for most categories (>0.8) and acceptable (>0.6) for all error types.

The Krippendorff measure may be deceptively high for some error types such as **Bad Math**, where 99% of tokens are not annotated with this error. The *Two Agree* measure in Table c.1 gives a different characterization of this data. *Two Agree* for a given error label is the percentage of tokens labeled by at least one annotator that were also labeled by one or more additional annotators. This metric allows us to see where annotators agree that particular errors exist while ignoring the majority of tokens (for most error categories) which annotators agree are not errors. *Two Agree* shows significantly lower rates for sparse errors with high Krippendorff scores, such as **Encyclopedic**. However, it reveals stronger agreement among **Incoherent** and **Off-Prompt** errors than might be expected given the Krippendorff coefficient.

A limitation for both metrics is the use of token-based overlap.

BOOTSTRAP One issue we face is high variance of annotations. To determine the impact of this variance for lower-data settings, we perform a bootstrap analysis using largest subset of our data (GPT-3, top-p = 0.96, t = 1, f.p.= 0, for which we have annotations of 200+ generations). We choose 50 generations (roughly 500 annotations) and calculate the error statistics therein. We repeat this process 1000 times and report the mean, standard deviation, and coefficient of variation in Table c.2. We also calculate the coefficient of variation for different numbers of samples, shown in Figure c.4. We see that as the number of samples increases, the coefficient of variation decreases as expected, though less precipitously after 30 examples. These results show that with as few as 50 documents, the SCARECROW error analysis should yield relatively robust results. However, this varies by error type: rare errors like **Bad Math** and **Encyclopedic** show greater variance. Here, again we repeat our recommendation to treat

annotations for these categories in aggregate. These results motivate our collection of at least 500 annotations per condition studied.

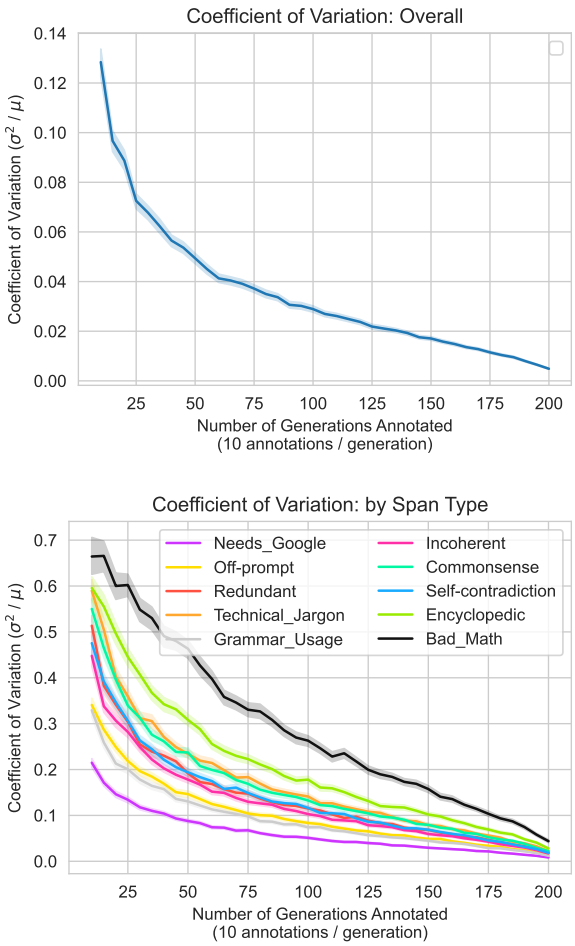


Figure c.4: Change in coefficient of variation as number of bootstrap samples increases overall (top), and by span type (bottom), with 95% confidence intervals. Data shown for GPT-3 with apples-to-apples decoding configuration (top-p = 0.96, t = 1, no *f.p.*).

C.4 FURTHER ANALYSIS

Here we analyze aspects of the data annotation omitted from the chapter body.

C.4.1 *Topics*

As noted in §5.4.3, we collect data using prompts drawn primarily from 12–14 news topics. For conciseness, we show results only for GPT-3, and only for the standard apples-to-apples decoding configuration.

Error	mean	std.	c.v. (%)
Bad Math	8.51	3.78	44.5
Commonsense	39.40	8.67	22.0
Encyclopedic	13.56	3.94	29.1
Grammar and Usage	126.19	16.81	13.3
Incoherent	96.89	16.58	17.1
Off-Prompt	167.29	23.39	14.0
Redundant	114.77	22.53	19.6
Self-Contradiction	60.54	11.94	19.7
Technical Jargon	100.95	24.09	23.9
Needs Google	482.84	42.22	8.7
Total errors	1268.48	55.59	19.72

Table c.2: Bootstrap analysis (sampling 50 generations) of error *counts*, by category (c.v. is the coefficient of variation).

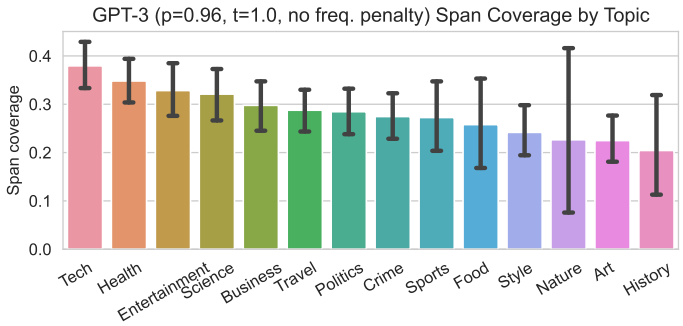


Figure c.5: Average span coverage for different topics (GPT-3 generations with apples-to-apples decoding configuration), with 95% confidence intervals. While the majority of topics display no significant trend, we observe that more technical topics such as *Tech* and *Health* are covered by a higher density of error spans than *Style* and *Art*.

Figure c.5 plots, based on the prompt topics, the average portion of the generation that is covered by error spans. While there is no significant difference between most topics, the results do indicate that generating text in more technical domains leads to higher span counts.

Figure c.6 shows individual span prevalence by topic. The top heatmap normalizes each topic (column) independently. **Needs Google** issues and **Off-Prompt** errors dominate the span types, with a few exceptions: for *History*, and *Nature* articles, **Redundant** trumps **Off-Prompt** as a source of errors.

For the bottom, if we instead normalize by error label (row), we can observe which topics are more prone to certain error types than others. For example, we can see **Bad Math** errors are most common in *Business* and *Health* generations; *Entertainment*

causes the most **Self-Contradiction** errors; and **Technical Jargon** appears more frequently in articles about *Business, Technology,* or *Health.*

C.4.2 *Error explanations*

Figure C.7 displays word clouds for common unigrams and bigrams found in the error explanations for each error type, and Figure C.8 shows the average explanation lengths for each error type. For **Technical Jargon**, **Redundant**, and **Needs Google** error types, the prominent words do not provide much illumination and they have short average explanation length, indicating that the explanations are straightforward affirmations of the category (*"I think this is financial jargon," "The information is repeated,"* or *"I would need Google to check this."*). But for categories like **Encyclopedic** and **Bad Math**, we observe some coarse trends: "year" is prevalent in both, "movie" appears in **Encyclopedic**, and "million" is present in **Bad Math**, which suggests that the explanations are more likely from outside knowledge and needs some calculation (*"The iPhone uses a lightening connector not a L-shaped connector," or "5000 feet is 1524 meters."*)

Figure C.9 presents a few representative explanations for four error types, taking particular note of their explanation lengths (Figure C.8). . Both **Self-Contradiction** and **Redundant** errors have antecedents, but their explanations are markedly different. Explanations for **Self-Contradiction** contain more information describing the particular semantics that is reversed, which are less obvious at first glance than other errors. On the other hand, **Redundant** errors are more straightforward to spot, often involving simple lexical overlap, and so don't require elaboration.

Explanations for **Commonsense** contain the true commonsense knowledge that the text violates, which may take several words to explain. But an explanation for a **Grammar and Usage** error simply corrects the error; as these errors are easier to fix, the explanation lengths are often short.

C.5 FUTURE WORK

We outline several further directions of study centering around the SCARECROW annotation framework, considering both natural implications and broader steps.

C.5.1 *SCARECROW Studies: Simple*

FIND THE BEST-PERFORMING GPT-3 DECODING HYPERPARAMETERS. We observed that for GPT-3, a frequency penalty value of 1 with argmax sampling produced fewer error spans than any other configuration (Fig. 5.4). We have not tried varying the frequency penalty to values *between* 0 and 1, or adding any *presence penalty* (§5.4.2), both of which then allow for fresh explorations of top-p and temperature.

STUDY DECODING PARAMETERS IN SMALLER MODELS. How good can (a fine-tuned) GPT-2 get? We saw decoding parameters considerably impacted GPT-3's per-

formance, moving it from edging out Grover to error rates close to humans (Fig. 5.4). Could such decoding changes have a similar effect on a GPT-2-sized model? Or might a smaller model favor different decoding hyperparameteres?

BACK-OFF ANNOTATIONS. We observed good annotator agreement given the complexity of the task, but the odds that two annotators agree exactly on each span's type and boundaries remains only moderate (§c.3). We did not try backing-off (a) error types into coarser categories (e.g., language, factual, reader issue) or even to binary presence; (b) span boundaries into phrase or sentence-level annotations. Applying a type of back-off could also allow clustering methods to discover different error ontologies.

IMPROVE AUTOMATIC ERROR DETECTION. While we present baseline results for automatic span error detection (§5.6), we anticipate that significant progress is still available in this new task.

C.5.2 *SCARECROW Studies: Complex*

ALIGN MULTIPLE ANNOTATIONS. In the current work, we largely treat annotators independently, with the exception of measuring their overlap to study agreement (§c.3) or taking their union to train prediction model (§5.6). However, we might consider other ways of viewing the 10 annotations for each generation together. For example, we might consider the aggregate decision of *whether* a token is labeled with *any* span a measure of how noticeable or jarring an error is. This measure may be related to error severity, but may be distinct from it.

One might also consider formal methods for computing annotation alignments. The Gamma measure, proposed by Mathet, Widlöcher, and Métivier (2015), satisfies the long list of criteria needed to align and measure SCARECROW annotations: spans of multiple types, with gaps, full and partial span overlap, more than three annotators, and the potential to merge or split annotations (which we have not addressed in this chapter). While we performed experiments with this measure, we experienced difficulties producing intuitive alignments with the authors' software, which disallows configuring parameters of the mixed-integer programming problem.[2] Emerging concurrent work (Titeux and Riad, 2021) offers a reimplementation of this measure that exposes additional parameters, which may be a promising avenue. However, it is possible that aligning annotations is a challenging task on its own that might require use of the explanations.

CHARACTERIZE ERROR NUANCE. Related to the previous point about error alignment, one might study whether model size affects span agreement. Anecdotally, errors from larger models like GPT-3—even of the same type, like **Commonsense** errors—are

2 The mixed-integer programming approach is also computationally intensive; e.g., memory alone prevented us from computing alignments for pilot studies with twenty annotators, even on a machine with 500GB of RAM.

more difficult to describe without careful consideration, and may also be more difficult to identify.

CHARACTERIZE REPETITION. Our quantitative studies of **Redundant** errors (e.g., Figs. 5.9 and 5.8) point to semantic repetition as the major issue that emerges as models are scaled. Though this effect may be mitigated by changes to the decoding algorithm (like the frequency penalty), we still observe that models have difficulty striking a balance of repetition. With excessive paraphrasing, generated text seems *stuck* on an idea. But equally, if a generation moves too quickly between ideas without linking them together or to an overall theme, the text lacks coherence. We posit that the issue of **Redundant** text emerges as the shadow of encompassing issues of narrative structure and discourse.

C.5.3 *Broadening SCARECROW*

CONSTRAINED GENERATION This chapter focuses on open-ended generation (§5.2), but a natural extension of this method would be to assessing constrained generation tasks, such as machine translation.

NEW ERROR TYPES Especially if considering a novel task setting, new error types may prove useful. For example, in constrained generation, one might consider an **Adequacy** error, which—as in machine translation—would indicate that the meaning of a span diverges from what is expected given the generation constraints. Furthermore, one might need to introduce annotations on the provided (not generated) text to account for desired semantic components that are *missing* from the generated text. Or, perhaps for a dialog setting, one might introduce a **Generic** label, which would indicate that a portion of the generation is otherwise coherent and correct, but offers a lack of new information.[3]

CORPUS-LEVEL EVALUATION Other work has considered the evaluation of natural language generations at-scale, looking at distributional properties of the text (Caccia et al., 2020; Pillutla et al., 2021). We suggest that these views are complementary to instance-based, human evaluation proposed here, and combining the approaches could lead towards a more holistic view of generative evaluation. For example, while all **Self-Contradiction** errors right now are *within-document*, one could similarly identify *cross-document* contradiction errors, where a model is inconsistent at a more global scale.

3 Such generic language may be seen as violating Grice's Maxims (Grice, 1975), for example, by providing a dearth of information *quantity*, or by flouting improper *manner* by lacking brevity.

C.5.4 *Applications*

DETECTING FACTUALITY One potential application of the SCARECROW data could be using the **Needs Google** spans as a dataset of its own. In addition to training models to identify spans that require verification, one could go a step further and consider *evidence retrieval* for each span, and even propose a classification task.[4]

EDITING ERRORS One errors can be detected, can they be fixed? The difficulty and scope of fixing SCARECROW-identified errors may depend on the span type, as error fixes may have cascading effects in the rest of the document.

4 Minimally, **Needs Google** spans from human-authored reputable news text should (hopefully) all be factually correct.

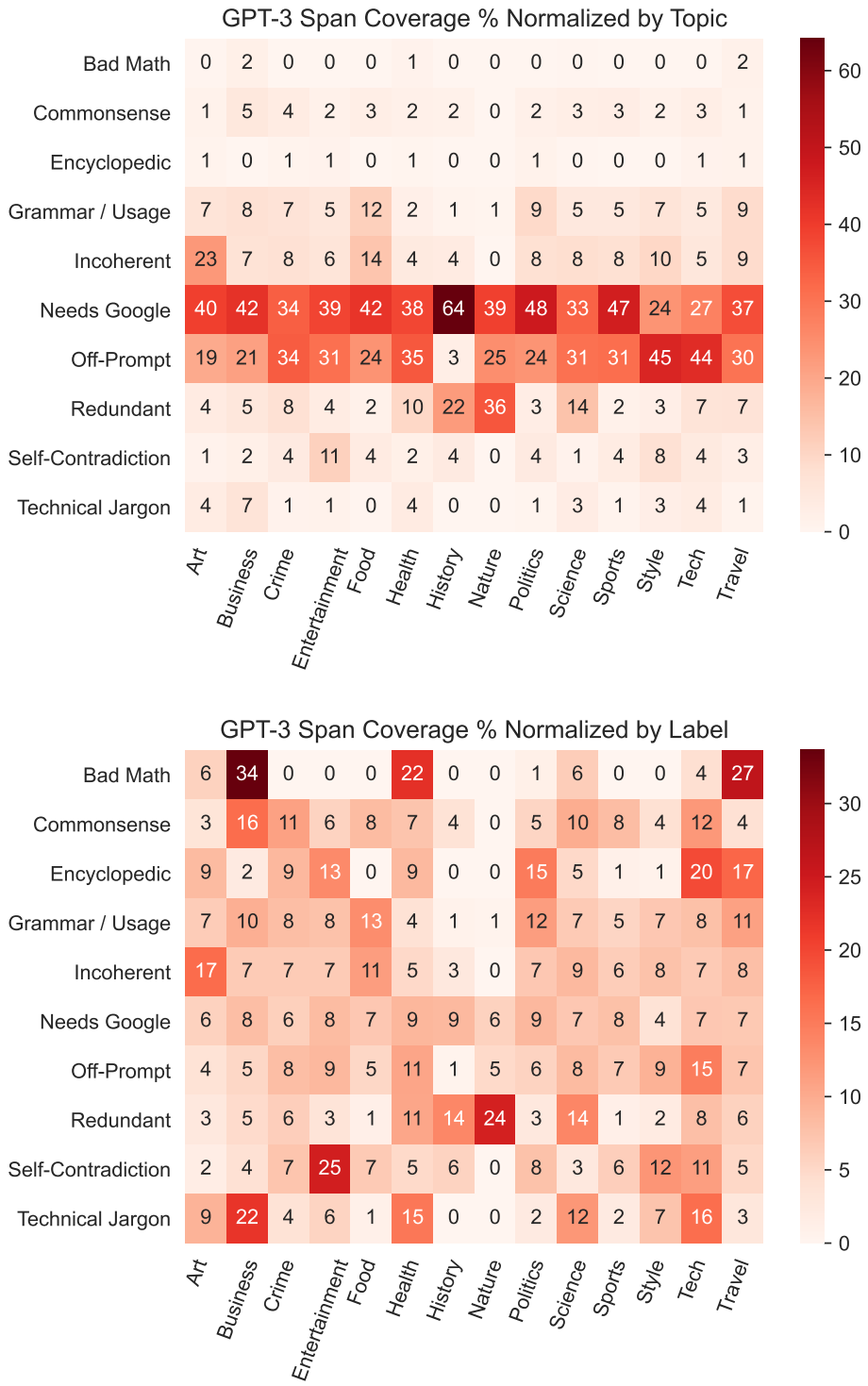


Figure c.6: Span coverage across both topic (x-axis) and span label (y-axis) for GPT-3 generated spans (*apples-to-apples* decoding: $p = 0.96$, $t = 1$, and no frequency penalty). **Top:** normalized by topic (column); **bottom:** normalized by span type (row).

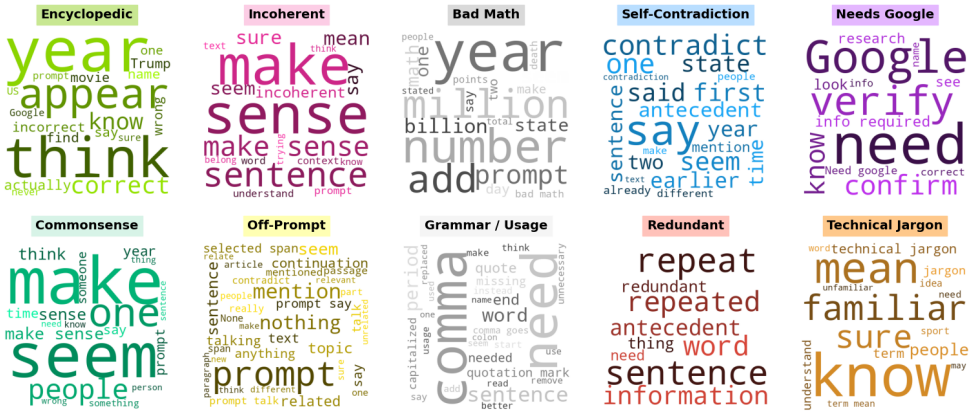


Figure c.7: Common unigrams and bi-grams observed in the explanations written for each annotated span, grouped by span type.

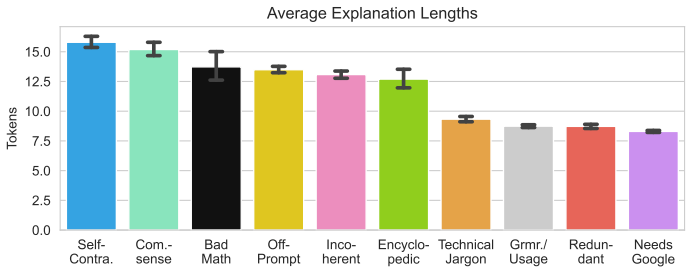


Figure c.8: Average number of tokens in explanation for each error type. We observe explanation length correlates with how obvious the error type is, where categories like **Grammar and Usage** and **Technical Jargon** are easier to find and explain than **Self-Contradiction** and **Commonsense**.

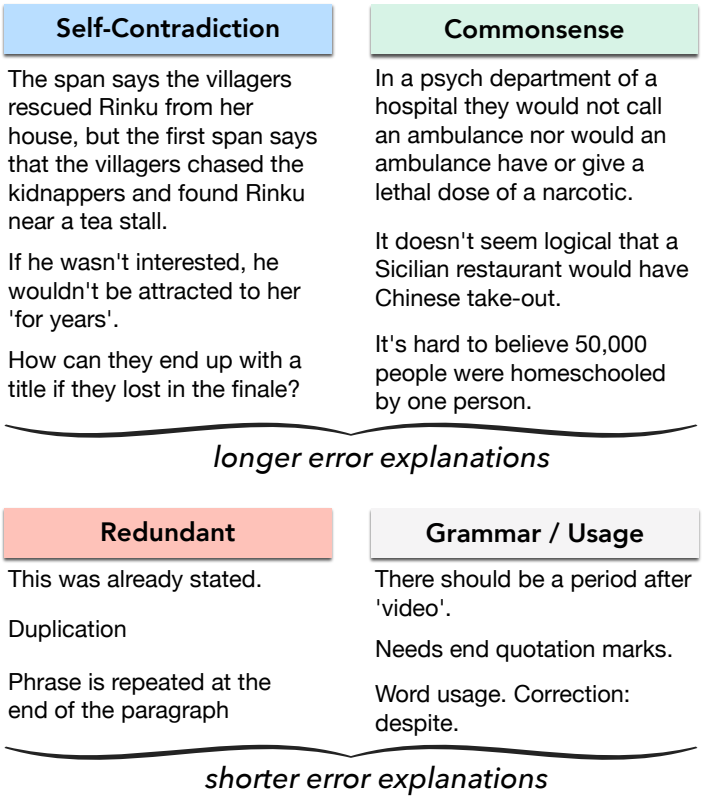


Figure c.9: Examples of error explanations from different error types that favor longer (top) and shorter (bottom) descriptions.

BIBLIOGRAPHY

Acharya, Anurag, Kartik Talamadupula, and Mark A Finlayson (2020). *An Atlas of Cultural Commonsense for Machine Reasoning*. arXiv: 2009.05664 [cs.AI].

Admoni, Henny and Brian Scassellati (2017). "Social eye gaze in human-robot interaction." In: *Journal of Human-Robot Interaction* 6, pp. 25 –63.

Angeli, Gabor and Christopher D Manning (2013). "Philosophers are Mortal: Inferring the Truth of Unseen Facts." In: *CoNLL*, pp. 133–142.

– (2014). "NaturalLI: Natural Logic Inference for Common Sense Reasoning." In: *EMNLP*, pp. 534–545.

Artzi, Yoav and Luke Zettlemoyer (2013). "Weakly Supervised Learning of Semantic Parsers for Mapping Instructions to Actions." In: *Transactions of the Association for Computational Linguistics* 1.1, pp. 49–62.

Bagherinezhad, Hessam, Hannaneh Hajishirzi, Yejin Choi, and Ali Farhadi (2016). "Are elephants bigger than butterflies? reasoning about sizes of objects." In: *arXiv preprint arXiv:1602.00753*.

Baker, Collin F, Charles J Fillmore, and John B Lowe (1998). "The berkeley framenet project." In: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pp. 86–90.

Baltrusaitis, Tadas, Chaitanya Ahuja, and Louis-Philippe Morency (2019). "Multimodal Machine Learning: A Survey and Taxonomy." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2, pp. 423–443. DOI: 10.1109/TPAMI.2018.2798607.

Banerjee, Satanjeev and Alon Lavie (2005). "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments." In: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72.

Bernstein, Basil (1960). "Language and social class." In: *The British journal of sociology* 11.3, pp. 271–276.

Bhagavatula, Chandra, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi (2020). "Abductive Commonsense Reasoning." In: *ICLR*.

Bisk, Yonatan, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Yue Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph P. Turian (2020). "Experience Grounds Language." In: *EMNLP*.

Blukis, Valts, Dipendra Misra, Ross A. Knepper, and Yoav Artzi (2018). "Mapping Navigation Instructions to Continuous Control Actions with Position Visitation Prediction." In: *Proceedings of the Conference on Robot Learning*. Zurich, Switzerland.

Bowdery, George J (1941). "Conventions and norms." In: *Philosophy of Science* 8.4, pp. 493–505.

Branwen, Gwern (2020). URL: https://www.gwern.net/GPT-3.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020). *Language Models are Few-Shot Learners*. arXiv: 2005.14165 [cs.CL].

Bruni, Elia, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran (2012). "Distributional semantics in technicolor." In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, pp. 136–145.

Caccia, Massimo, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin (2020). *Language GANs Falling Short*. arXiv: 1811.02549 [cs.CL].

Card, Dallas, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky (2020). "With Little Power Comes Great Responsibility." In: *Proceedings of EMNLP*.

Carlini, Nicholas, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel (2020). "Extracting Training Data from Large Language Models." In: *arXiv preprint arXiv:2012.07805*.

Carlson, Greg N (1977). "A unified analysis of the English bare plural." In: *Linguistics and philosophy* 1.3, pp. 413–457.

Celikyilmaz, Asli, Elizabeth Clark, and Jianfeng Gao (2021). *Evaluation of Text Generation: A Survey*. arXiv: 2006.14799 [cs.CL].

Charniak, Eugene (2000). "A maximum-entropy-inspired parser." In: *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Cheok, Ming Jin, Zaid Omar, and Mohamed Hisham Jaward (2019). "A review of hand gesture and sign language recognition techniques." In: *International Journal of Machine Learning and Cybernetics* 10, pp. 131–153.

Clark, Elizabeth and Noah A. Smith (June 2021). "Choose Your Own Adventure: Paired Suggestions in Collaborative Writing for Evaluating Story Generation Models." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 3566–3575. URL: https://www.aclweb.org/anthology/2021.naacl-main.279.

Das, Pradipto, Chenliang Xu, Richard F. Doell, and Jason J. Corso (2013). "A Thousand Frames in Just a Few Words: Lingual Description of Videos through Latent Topics and Sparse Object Stitching." In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2634–2641.

Davidov, Dmitry and Ari Rappoport (2010). "Extraction and approximation of numerical attributes from the web." In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 1308–1317.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database." In.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "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)*, pp. 4171–4186.

Ding, Haibo and Ellen Riloff (2016). "Acquiring Knowledge of Affective Events from Blogs Using Label Propagation." In: *AAAI*.

Dou, Yao, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A. Smith, and Yejin Choi (2021). *Scarecrow: A Framework for Scrutinizing Machine Text*. arXiv: 2107.01294 [cs.CL].

Dowty, David (1991). "Thematic proto-roles and argument selection." In: *language*, pp. 547–619.

Dugan, Liam, Daphne Ippolito, Arun Kirubarajan, and Chris Callison-Burch (2020). "RoFT: A Tool for Evaluating Human Detection of Machine-Generated Text." In: *arXiv preprint arXiv:2010.03070*.

Elster, Jon (2006). "Fairness and norms." In: *Social Research*, pp. 365–376.

Fillmore, Charles J (1976). "Frame semantics and the nature of language." In: *Annals of the New York Academy of Sciences* 280.1, pp. 20–32.

Fillmore, Charles J and Collin F Baker (2001). "Frame semantics for text understanding." In: *Proceedings of WordNet and Other Lexical Resources Workshop, NAACL*. Vol. 6.

Forbes, Maxwell and Yejin Choi (2017). "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)*.

Forbes, Maxwell, Jena D. Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi (2020). "Social Chemistry 101: Learning to Reason about Social and Moral Norms." In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Forbes, Maxwell, Christine Kaeser-Chen, Piyush Sharma, and Serge Belongie (2019). "Neural Naturalist: Generating Fine-Grained Image Comparisons." In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Fulgoni, Dean, Jordan Carpenter, Lyle Ungar, and Daniel Preoţiuc-Pietro (2016). "An empirical exploration of moral foundations theory in partisan news sources." In: *LREC*, pp. 3730–3736.

Gao, Qiaozi, Malcolm Doering, Shaohua Yang, and Joyce Y Chai (2016). "Physical causality of action verbs in grounded language understanding." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*. Vol. 1, pp. 1814–1824.

Gao, Tianyu, Adam Fisch, and Danqi Chen (2020). "Making Pre-trained Language Models Better Few-shot Learners." In: *arXiv preprint arXiv:2012.15723*.

Goldberg, Yoav and Jon Orwant (2013). "A dataset of syntactic-ngrams over time from a very large corpus of english books." In: *Second Joint Conference on Lexical and Computational Semantics (* SEM)*. Vol. 1, pp. 241–247.

Gordon, Jonathan and Lenhart K Schubert (2012). "Using textual patterns to learn expected event frequencies." In: *Proceedings of the Joint Workshop on Automatic*

Knowledge Base Construction and Web-scale Knowledge Extraction. Association for Computational Linguistics, pp. 122–127.

Gordon, Jonathan and Benjamin Van Durme (2013). "Reporting bias and knowledge acquisition." In: *Proceedings of the 2013 workshop on Automated knowledge base construction*. ACM, pp. 25–30.

Gordon, Jonathan, Benjamin Van Durme, and Lenhart K Schubert (2010). "Learning from the Web: Extracting General World Knowledge from Noisy Text." In: *Collaboratively-Built Knowledge Sources and AI*.

Graham, Jesse, Jonathan Haidt, and Brian A Nosek (May 2009). "Liberals and conservatives rely on different sets of moral foundations." en. In: *J. Pers. Soc. Psychol.* 96.5, pp. 1029–1046.

Grice, Herbert P. (1975). "Logic and conversation." In: *Syntax and Semantics*. Ed. by P. Cole and J. Morgan. Vol. 3: Speech Acts. Academic Press, New York.

Gu, Jing, Qingyang Wu, and Zhou Yu (2020). "Perception Score, A Learned Metric for Open-ended Text Generation Evaluation." In: *arXiv preprint arXiv:2008.03082*.

Guan, Jian and Minlie Huang (2020). "UNION: An Unreferenced Metric for Evaluating Open-ended Story Generation." In: *arXiv preprint arXiv:2009.07602*.

Guo, Ruiqi, Sanjiv Kumar, Krzysztof Choromanski, and David Simcha (2016). "Quantization based fast inner product search." In: *Artificial Intelligence and Statistics*, pp. 482–490.

Haidt, Jonathan (2012). *The righteous mind: Why good people are divided by politics and religion*. Vintage.

Haidt, Jonathan, Silvia Helena Koller, and Maria G Dias (1993). "Affect, culture, and morality, or is it wrong to eat your dog?" In: *Journal of personality and social psychology* 65.4, p. 613.

Hare, Richard Mervyn, Richard Mervyn Hare, Richard Mervyn Hare Hare, and Richard M Hare (1981). *Moral thinking: Its levels, method, and point*. Oxford: Clarendon Press; New York: Oxford University Press.

Hashimoto, Tatsunori, Hugh Zhang, and Percy Liang (June 2019). "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1689–1701. DOI: 10.18653/v1/N19-1169. URL: https://www.aclweb.org/anthology/N19-1169.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hendricks, Lisa Anne, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell (2016). "Generating visual explanations." In: *European Conference on Computer Vision*. Springer, pp. 3–19.

Hendrycks, Dan, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt (2020). *Aligning AI With Shared Human Values*. arXiv: 2008.02275 [cs.CY].

Hinton, Geoffrey E., Li Deng, Dong Yu, George E. Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew W. Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury (2012). "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups." In: *IEEE Signal Processing Magazine* 29, pp. 82–97.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean (2015). "Distilling the knowledge in a neural network." In: *arXiv preprint arXiv:1503.02531*.

Holtzman, Ari, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi (2020a). "The Curious Case of Neural Text Degeneration." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=rygGQyrFvH.

Holtzman, Ari, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi (2018). "Learning to write with cooperative discriminators." In: *arXiv preprint arXiv:1805.06087*.

Holtzman, Ari, Jan Buys, Maxwell Forbes, and Yejin Choi (2020b). "The Curious Case of Neural Text Degeneration." In: *International Conference on Learning Representations*.

Hovy, Dirk and Diyi Yang (2021). "The Importance of Modeling Social Factors of Language: Theory and Practice." In: *NAACL*.

Howcroft, David M., Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser (Dec. 2020). "Twenty Years of Confusion in Human Evaluation: NLG Needs Evaluation Sheets and Standardised Definitions." In: *Proceedings of the 13th International Conference on Natural Language Generation*. Dublin, Ireland: Association for Computational Linguistics, pp. 169–182. URL: https://www.aclweb.org/anthology/2020.inlg-1.23.

Ilharco, Gabriel, Rowan Zellers, Ali Farhadi, and Hannaneh Hajishirzi (June 2021). "Probing Contextual Language Models for Common Ground with Visual Representations." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 5367–5377. DOI: 10.18653/v1/2021.naacl-main.422. URL: https://aclanthology.org/2021.naacl-main.422.

Izadinia, Hamid, Fereshteh Sadeghi, Santosh K Divvala, Hannaneh Hajishirzi, Yejin Choi, and Ali Farhadi (2015). "Segment-phrase table for semantic segmentation, visual entailment and paraphrasing." In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 10–18.

Jawahar, Ganesh, Benoit Sagot, and Djame Seddah (July 2019). "What Does BERT Learn about the Structure of Language?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3651–3657. DOI: 10.18653/v1/P19-1356. URL: https://aclanthology.org/P19-1356.

Jhamtani, Harsh and Taylor Berg-Kirkpatrick (2018). "Learning to describe differences between pairs of similar images." In: *arXiv preprint arXiv:1808.10584*.

Jurafsky, Daniel and James H. Martin (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 1st. USA: Prentice Hall PTR. ISBN: 0130950696.

Kagan, Jerome (1984). *The nature of the child*. Basic Books.

Kako, Edward (2006). "Thematic role properties of subjects and objects." In: *Cognition* 101.1, pp. 1–42.

Kallia, Alexandra (2004). "Linguistic politeness: The implicature approach." In: *Multilingua* 23.1/2, pp. 145–170.

Kasai, Jungo, Keisuke Sakaguchi, Lavinia Dunagan, Jacob Morrison, Ronan Le Bras, Yejin Choi, and Noah A. Smith (2021). *Transparent Human Evaluation for Image Captioning*. URL: https://arxiv.org/abs/2111.08940.

Khashabi, Daniel, Gabriel Stanovsky, Jonathan Bragg, Nicholas Lourie, Jungo Kasai, Yejin Choi, Noah A. Smith, and Daniel S. Weld (2021). *GENIE: A Leaderboard for Human-in-the-Loop Evaluation of Text Generation*. arXiv: 2101.06561 [cs.CL].

Khosla, Aditya, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei (2011). "Novel Dataset for Fine-Grained Image Categorization." In: *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*. Colorado Springs, CO.

Kipper, Karin, Hoa Trang Dang, Martha Palmer, et al. (2000). "Class-based construction of a verb lexicon." In: *AAAI/IAAI* 691, p. 696.

Kitts, James A and Yen-Sheng Chiang (2008). *Encyclopedia of Social Problems,* ed. by Vincent NEditor Parillo.

Koehn, Philipp (2009). *Statistical machine translation*. Cambridge University Press.

Kohlberg, Lawrence (1976). "Moral stages and moralization." In: *Moral development and behavior*, pp. 31–53.

Kojima, Noriyuki, Alane Suhr, and Yoav Artzi (2021). "Continual Learning for Grounded Instruction Generation by Observing Human Following Behavior." In: *Transactions of the Association of Computational Linguistics*.

Krause, Jonathan, Michael Stark, Jia Deng, and Li Fei-Fei (2013a). "3D Object Representations for Fine-Grained Categorization." In: *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*. Sydney, Australia.

– (2013b). "3D Object Representations for Fine-Grained Categorization." In: *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*. Sydney, Australia.

Krippendorff, Klaus (2018). *Content analysis: An introduction to its methodology*. Sage publications.

Lenat, Douglas B (1995). "CYC: A large-scale investment in knowledge infrastructure." In: *Communications of the ACM* 38.11, pp. 33–38.

Levin, Beth (1993). *English verb classes and alternations: A preliminary investigation*. University of Chicago press.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer (2019). "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." In: *arXiv preprint arXiv:1910.13461*.

Li, Xiang, Aynaz Taheri, Lifu Tu, and Kevin Gimpel (2016). "Commonsense knowledge base completion." In: *Proceedings of the 54th Annual Meeting of the Association for Com-*

putational Linguistics (ACL), Berlin, Germany, August. Association for Computational Linguistics.

Lin, Chin-Yew (2004). "Rouge: A package for automatic evaluation of summaries." In: *Text summarization branches out*, pp. 74–81.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick (2014). "Microsoft coco: Common objects in context." In: *European conference on computer vision*. Springer, pp. 740–755.

Liu, Hugo and Push Singh (2004). "ConceptNet—a practical commonsense reasoning tool-kit." In: *BT technology journal* 22.4, pp. 211–226.

Liu, Jiacheng, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi (2021). *Generated Knowledge Prompting for Commonsense Reasoning*. arXiv: 2110.08387 [cs.CL].

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019a). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv: 1907.11692 [cs.CL].

– (2019b). "RoBERTa: A Robustly Optimized BERT Pretraining Approach." In: *ArXiv* abs/1907.11692.

Lourie, Nicholas, Ronan Le Bras, and Yejin Choi (2020). "Scruples: A Corpus of Community Ethical Judgments on 32,000 Real-Life Anecdotes." In: *arXiv e-prints*. arXiv: 2008.09094.

Maji, Subhransu (2012). "Discovering a lexicon of parts and attributes." In: *European Conference on Computer Vision*. Springer, pp. 21–30.

Maji, Subhransu, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi (2013). "Fine-grained visual classification of aircraft." In: *arXiv preprint arXiv:1306.5151*.

Malle, Bertram F, Steve Guglielmo, and Andrew E Monroe (2014). "A theory of blame." In: *Psychological Inquiry* 25.2, pp. 147–186.

Manning, Christopher and Hinrich Schutze (1999). *Foundations of statistical natural language processing*. MIT press.

Marmpena, Mina, Angelica Lim, Torbjørn S. Dahl, and Nikolas J. Hemion (2019). "Generating robotic emotional body language with variational autoencoders." In: *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 545–551.

Mathet, Yann, Antoine Widlöcher, and Jean-Philippe Métivier (2015). "The unified and holistic method gamma (γ) for inter-annotator agreement measure and alignment." In: *Computational Linguistics* 41.3, pp. 437–479.

Mehrabian, Albert and Susan R Ferris (1967). "Inference of attitudes from nonverbal communication in two channels." In: *Journal of consulting psychology* 31.3, p. 248.

Mehrabian, Albert and Morton Wiener (1967). "Decoding of inconsistent communications." In: *Journal of personality and social psychology* 6.1, p. 109.

Michel, Jean-Baptiste, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. (2011). "Quantitative analysis of culture using millions of digitized books." In: *science* 331.6014, pp. 176–182.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Efficient estimation of word representations in vector space." In: *arXiv preprint arXiv:1301.3781*.

Mikolov, Tomas, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur (2010). "Recurrent neural network based language model." In: *Eleventh annual conference of the international speech communication association*.

Miller, George A (1995). "WordNet: a lexical database for English." In: *Communications of the ACM* 38.11, pp. 39–41.

Misra, Ishan, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick (2016). "Seeing through the Human Reporting Bias: Visual Classifiers from Noisy Human-Centric Labels." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2930–2939.

Monz, Christof and Maarten de Rijke (2001). "Light-weight entailment checking for computational semantics." In: *Proc. of the third workshop on inference in computational semantics (ICoS-3)*.

Mostafazadeh, Nasrin, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen (June 2016). "A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories." In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 839–849. DOI: 10.18653/v1/N16-1098. URL: https://www.aclweb.org/anthology/N16-1098.

Muller, Cornelia, Alan Cienki, Ellen Fricke, Silva H Ladewig, David McNeill, and Sedinha Tessendorf (2013). "Body-language-communication." In: *An international handbook on multimodality in human interaction* 1.1, pp. 131–232.

Murphy-Chutorian, Erik and Mohan Manubhai Trivedi (2009). "Head pose estimation in computer vision: A survey." In: *IEEE transactions on pattern analysis and machine intelligence* 31.4, pp. 607–626.

Narisawa, Katsuma, Yotaro Watanabe, Junta Mizuno, Naoaki Okazaki, and Kentaro Inui (2013). "Is a 204 cm Man Tall or Small? Acquisition of Numerical Common Sense from the Web." In: *ACL (1)*, pp. 382–391.

Nørregaard, J, B D Horne, and S Adalı (2019). "NELA-GT-2018: A large multi-labelled news dataset for the study of misinformation in news articles." In: *AAAI*. wvvw.aaai.org.

Novikova, Jekaterina, Ondřej Dušek, and Verena Rieser (2018). "RankME: Reliable human ratings for natural language generation." In: *arXiv preprint arXiv:1803.05928*.

Palmer, Martha, Daniel Gildea, and Paul Kingsbury (2005). "The proposition bank: An annotated corpus of semantic roles." In: *Computational linguistics* 31.1, pp. 71–106.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). "BLEU: a method for automatic evaluation of machine translation." In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, pp. 311–318.

Park, Dong Huk, Trevor Darrell, and Anna Rohrbach (2019). "Robust Change Captioning." In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4624–4633.

Pease, Barbara and Allan Pease (2008). *The definitive book of body language: The hidden meaning behind people's gestures and expressions*. Bantam.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). "GloVe: Global Vectors for Word Representation." In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

Pereira, Gonçalo, Rui Prada, and Pedro A Santos (Dec. 2016). "Integrating social power into the decision-making of cognitive agents." In: *Artificial Intelligence* 241, pp. 1–44.

Perkins, H Wesley and Alan D Berkowitz (1986). "Perceiving the community norms of alcohol use among students: Some research implications for campus alcohol education programming." In: *International journal of the Addictions* 21.9-10, pp. 961–976.

Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). "Deep Contextualized Word Representations." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237.

Pillutla, Krishna, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Yejin Choi, and Zaid Harchaoui (2021). *MAUVE: Human-Machine Divergence Curves for Evaluating Open-Ended Text Generation*. arXiv: 2102.01454 [cs.CL].

Qi, Peng, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning (July 2020). "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics, pp. 101–108. DOI: 10.18653/v1/2020.acl-demos.14. URL: https://aclanthology.org/2020.acl-demos.14.

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018a). "Improving Language Understanding by Generative Pre-Training." In.

– (2018b). "Improving language understanding by generative pre-training." In.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019a). "Language models are unsupervised multitask learners." In: *OpenAI Blog*.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. (2019b). "Language models are unsupervised multitask learners." In: *OpenAI blog* 1.8, p. 9.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu (2019). "Exploring the limits of transfer learning with a unified text-to-text transformer." In: *arXiv preprint arXiv:1910.10683*.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang (2016). "Squad: 100,000+ questions for machine comprehension of text." In: *arXiv preprint arXiv:1606.05250*.

Rashkin, Hannah, Maarten Sap, Emily Allaway, Noah A Smith, and Yejin Choi (2018). "Event2Mind: Commonsense Inference on Events, Intents, and Reactions." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 463–473.

Reboul, Anne (2015). "Why language really is not a communication system: a cognitive view of language evolution." In: *Frontiers in Psychology* 6, p. 1434. ISSN: 1664-1078. DOI:

10.3389/fpsyg.2015.01434. URL: https://www.frontiersin.org/article/10.3389/fpsyg.2015.01434.

Reed, Scott, Zeynep Akata, Honglak Lee, and Bernt Schiele (2016). "Learning deep representations of fine-grained visual descriptions." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–58.

Reisinger, Drew, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme (2015). "Semantic Proto-Roles." In: *Transactions of the Association for Computational Linguistics* 3, pp. 475–488. ISSN: 2307-387X.

Reynolds, Laria and Kyle McDonell (2021). "Prompt programming for large language models: Beyond the few-shot paradigm." In: *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–7.

Rezapour, Rezvaneh, Saumil H Shah, and Jana Diesner (2019). "Enhancing the Measurement of Social Effects by Capturing Morality." In: *Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Minneapolis, USA: Association for Computational Linguistics, pp. 35–45.

Richardson, Matthew, Christopher JC Burges, and Erin Renshaw (2013). "Mctest: A challenge dataset for the open-domain machine comprehension of text." In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 193–203.

Rubinstein, Dana, Effi Levi, Roy Schwartz, and Ari Rappoport (2015). "How well do distributional models capture different types of semantic knowledge?" In: *ACL (2)*, pp. 726–730.

Rudinger, Rachel, Vered Shwartz, Jena D. Hwang, Chandra Bhagavatula, Maxwell Forbes, Ronan Le Bras, Noah A. Smith, and Yejin Choi (Nov. 2020). "Thinking Like a Skeptic: Defeasible Inference in Natural Language." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 4661–4675. DOI: 10.18653/v1/2020.findings-emnlp.418. URL: https://aclanthology.org/2020.findings-emnlp.418.

Sadeghi, Fereshteh, Santosh K Kumar Divvala, and Ali Farhadi (2015). "Viske: Visual knowledge extraction and question answering by visual verification of relation phrases." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1456–1464.

Sai, Ananya B, Akash Kumar Mohankumar, and Mitesh M Khapra (2020). "A Survey of Evaluation Metrics Used for NLG Systems." In: *arXiv preprint arXiv:2008.12009*.

Sap, Maarten, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith (2019a). "The risk of racial bias in hate speech detection." In: *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 1668–1678.

Sap, Maarten, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi (2020). "Social Bias Frames: Reasoning about Social and Power Implications of Language." In: *ACL*.

Sap, Maarten, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi (2019b). "ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning." In: *AAAI*.

Sap, Maarten, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi (2019c). "Social IQa: Commonsense Reasoning about Social Interactions." 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), pp. 4453–4463.

Schank, Roger C and Robert P Abelson (1977). *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology Press.

Shan, Caifeng, Shaogang Gong, and Peter W. McOwan (2009). "Facial expression recognition based on Local Binary Patterns: A comprehensive study." In: *Image Vis. Comput.* 27, pp. 803–816.

Sharma, Piyush, Nan Ding, Sebastian Goodman, and Radu Soricut (2018). "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)*. Vol. 1, pp. 2556–2565.

She, Lanbo and Joyce Y Chai (2016). "Incremental acquisition of verb hypothesis space towards physical world interaction." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Shweder, Richard A (1990). "In defense of moral realism: Reply to Gabennesch." In: *Child Development* 61.6, pp. 2060–2067.

Silberer, Carina, Vittorio Ferrari, and Mirella Lapata (2013). "Models of Semantic Representation with Visual Attributes." In: *ACL (1)*, pp. 572–582.

Smith, Noah A (2011). "Linguistic structure prediction." In: *Synthesis lectures on human language technologies* 4.2, pp. 1–274.

Sorower, Mohammad S, Janardhan R Doppa, Walker Orr, Prasad Tadepalli, Thomas G Dietterich, and Xiaoli Z Fern (2011). "Inverting Grice's maxims to learn rules from natural language extractions." In: *Advances in neural information processing systems*, pp. 1053–1061.

Su, Jong-Chyi, Chenyun Wu, Huaizu Jiang, and Subhransu Maji (2017). "Reasoning about Fine-grained Attribute Phrases using Reference Games." In: *International Conference on Computer Vision (ICCV)*.

Suhr, Alane, Stephanie Zhou, Iris Zhang, Huajun Bai, and Yoav Artzi (2018). "A Corpus for Reasoning About Natural Language Grounded in Photographs." In: *CoRR* abs/1811.00491. arXiv: 1811.00491. URL: http://arxiv.org/abs/1811.00491.

Sutton, Richard S. (2019). *The Bitter Lesson*. URL: http://www.incompleteideas.net/IncIdeas/BitterLesson.html (visited on 11/17/2021).

Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi (2017). "Inception-v4, inception-resnet and the impact of residual connections on learning." In: *Thirty-First AAAI Conference on Artificial Intelligence*.

Takamura, Hiroya and Jun'ichi Tsujii (2015). "Estimating Numerical Attributes by Bringing Together Fragmentary Clues." In: *HLT-NAACL*, pp. 1305–1310.

Tan, Hao and Mohit Bansal (2019). "Expressing visual relationships via language." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Tandon, Niket, Gerard De Melo, and Gerhard Weikum (2014). "Acquiring Comparative Commonsense Knowledge from the Web." In: *AAAI*, pp. 166–172.

Tandon, Niket, Charles Hariman, Jacopo Urbani, Anna Rohrbach, Marcus Rohrbach, and Gerhard Weikum (2016). "Commonsense in Parts: Mining Part-Whole Relations from the Web and Image Tags." In: *AAAI*, pp. 243–250.

Tay, Yi, Donovan Ong, Jie Fu, Alvin Chan, Nancy Chen, Anh Tuan Luu, and Christopher Pal (2020). "Would you Rather? A New Benchmark for Learning Machine Alignment with Cultural Values and Social Preferences." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5369–5373.

Tenney, Ian, Dipanjan Das, and Ellie Pavlick (July 2019). "BERT Rediscovers the Classical NLP Pipeline." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4593–4601. DOI: 10.18653/v1/P19-1452. URL: https://aclanthology.org/P19-1452.

Titeux, Hadrien and Rachid Riad (2021). "pygamma-agreement: Gamma γ measure for inter/intra-annotator agreement in Python." In.

Toutanova, Kristina, Dan Klein, Christopher D Manning, and Yoram Singer (2003). "Feature-rich part-of-speech tagging with a cyclic dependency network." In: *Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology-volume 1*. Association for Computational Linguistics, pp. 173–180.

Trott, Sean, Stefanie Reed, Victor Ferreira, and Benjamin Bergen (2019). "Prosodic cues signal the intent of potential indirect requests." In: *CogSci*, pp. 1142–1148.

Van Durme, Benjamin D (2010). *Extracting implicit knowledge from text*. University of Rochester.

Van Hee, Cynthia, Els Lefever, Ben Verhoeven, Julie Mennes, Bart Desmet, Guy De Pauw, Walter Daelemans, and Veronique Hoste (Sept. 2015). "Detection and Fine-Grained Classification of Cyberbullying Events." In: *Proceedings of the International Conference Recent Advances in Natural Language Processing*. Hissar, Bulgaria: INCOMA Ltd. Shoumen, BULGARIA, pp. 672–680. URL: https://www.aclweb.org/anthology/R15-1086.

Van Horn, Grant, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie (2018). "The inaturalist species classification and detection dataset." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need." In: *Advances in Neural Information Processing Systems*, pp. 5998–6008.

Vedantam, Ramakrishna, Samy Bengio, Kevin Murphy, Devi Parikh, and Gal Chechik (2017). "Context-aware captions from context-agnostic supervision." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 251–260.

Vedantam, Ramakrishna, C Lawrence Zitnick, and Devi Parikh (2015). "Cider: Consensus-based image description evaluation." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575.

Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan (2015). "Show and tell: A neural image caption generator." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164.

Volkova, Svitlana, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas (2017). "Separating Facts from Fiction: Linguistic Models to Classify Suspicious and Trusted News Posts on Twitter." In: *ACL*. Vancouver, Canada: Association for Computational Linguistics, pp. 647–653.

Vu, Hoa Trong, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura (2014). "Acquiring a dictionary of emotion-provoking events." In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pp. 128–132.

Wadden, David, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi (Nov. 2019). "Entity, Relation, and Event Extraction with Contextualized Span Representations." 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 5784–5789. DOI: 10.18653/v1/D19-1585. URL: https://www.aclweb.org/anthology/D19-1585.

Wah, Catherine, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie (2011). "The caltech-ucsd birds-200-2011 dataset." In.

Wang, Alex, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (2019). "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems." In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf.

Wang, Su, Greg Durrett, and Katrin Erk (2018). "Modeling Semantic Plausibility by Injecting World Knowledge." In: *NAACL-HLT*.

Wang, Tongzhou, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros (2018). "Dataset distillation." In: *arXiv preprint arXiv:1811.10959*.

Weber, René, J Michael Mangus, Richard Huskey, Frederic R Hopp, Ori Amir, Reid Swanson, Andrew Gordon, Peter Khooshabeh, Lindsay Hahn, and Ron Tamborini (Apr. 2018). "Extracting Latent Moral Information from Text Narratives: Relevance, Challenges, and Solutions." In: *Commun. Methods Meas.* 12.2-3, pp. 119–139.

West, Peter, Chandra Bhagavatula, Jack Hessel, Jena D. Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi (2021). *Symbolic Knowledge Distillation: from General Language Models to Commonsense Models*. arXiv: 2110.07178 [cs.CL].

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew (2019). "HuggingFace's Transformers: State-of-the-art Natural Language Processing." In: *ArXiv* abs/1910.03771.

Wood, Gavin, Kiel Long, Tom Feltwell, Scarlett Rowland, Phillip Brooker, Jamie Mahoney, John Vines, Julie Barnett, and Shaun Lawson (2018). "Rethinking engagement

with online news through social and visual co-annotation." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–12.

Wu, Xiang, Ruiqi Guo, Ananda Theertha Suresh, Sanjiv Kumar, Daniel N Holtmann-Rice, David Simcha, and Felix Yu (2017). "Multiscale quantization for fast similarity search." In: *Advances in Neural Information Processing Systems*, pp. 5745–5755.

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio (2015). "Show, attend and tell: Neural image caption generation with visual attention." In: *International conference on machine learning*, pp. 2048–2057.

Zellers, Rowan, Ari Holtzman, Elizabeth Clark, Lianhui Qin, Ali Farhadi, and Yejin Choi (June 2021a). "TuringAdvice: A Generative and Dynamic Evaluation of Language Use." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 4856–4880. URL: https://www.aclweb.org/anthology/2021.naacl-main.386.

Zellers, Rowan, Ari Holtzman, Matthew E. Peters, Roozbeh Mottaghi, Aniruddha Kembhavi, Ali Farhadi, and Yejin Choi (2021b). "PIGLeT: Language Grounding Through Neuro-Symbolic Interaction in a 3D World." In: *ACL/IJCNLP*.

Zellers, Rowan, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi (2019). "Defending Against Neural Fake News." In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., pp. 9054–9065. URL: http://papers.nips.cc/paper/9106-defending-against-neural-fake-news.pdf.

Zhang, Sheng, Rachel Rudinger, Kevin Duh, and Ben Van Durme (2017). "Ordinal Common-sense Inference." In: *Transactions of the Association for Computational Linguistics*.

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi (2019). "Bertscore: Evaluating text generation with bert." In: *arXiv preprint arXiv:1904.09675*.

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi (2020). "BERTScore: Evaluating Text Generation with BERT." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=SkeHuCVFDr.

Zhu, Yuke, Alireza Fathi, and Li Fei-Fei (2014). "Reasoning about object affordances in a knowledge base representation." In: *European conference on computer vision*. Springer, pp. 408–424.

Zou, James Y, Kamalika Chaudhuri, and Adam Tauman Kalai (2015). "Crowdsourcing feature discovery via adaptively chosen comparisons." In: *arXiv preprint arXiv:1504.00064*.

COLOPHON

This dissertation was typeset using the typographical look-and-feel `classicthesis`, developed by André Miede and Ivo Pletikosić, and further refined by Amrita Mazumdar.

Final Version as of December 16, 2021 (classicthesis v4.6).