Penguins Don't Fly: Reasoning about Generics through Instantiations and Exceptions

Emily Allaway[⋆] **Jena D. Hwang**[†] **Chandra Bhagavatula**[†]
Kathleen McKeown[⋆] **Doug Downey**[†] **Yejin Choi**[†‡]
[⋆]Columbia University, USA
[†]Allen Institute for Artificial Intelligence, USA
[‡]Paul G. Allen School of Computer Science & Engineering, University of Washington, USA
eallaway@cs.columbia.edu

Abstract

Generics express generalizations about the world (e.g., "birds can fly"). However, they are not universally true – while sparrows and penguins are both birds, only sparrows can fly and penguins cannot. Commonsense knowledge bases, that are used extensively in many NLP tasks as a source of world-knowledge, can often encode generic knowledge but, by-design, cannot encode such exceptions. It is crucial to realize the specific instances when a generic statement is true or false. In this work, we present a novel framework to generate pragmatically relevant true and false instances of a generic. We use pre-trained language models, constraining the generation based on insights from linguistic theory, and produce $\sim 20k$ EXEMPLARS for ~ 650 generics. Our system outperforms few-shot generation from GPT-3 (by 12.5 precision points) and our analysis highlights the importance of constrained decoding for this task and the implications of generics EXEMPLARS for language inference tasks.

1 Introduction

Generalities (e.g., "birds can fly") facilitate efficient information processing and reasoning (Mercier and Sperber, 2017) and thus underlie many commonsense KBs (CKBs) (e.g., ConceptNet (Speer et al., 2017)). In particular, edges in a CKB are assumed to represent generalities (i.e., they do not always need to be true), *not* universal statements. While this allows the representation of salient commonsense knowledge without exhaustive annotation (i.e., an open-world assumption (Reiter, 1978b)), it also results in resources that are less informative for more specialized knowledge (e.g., "bird" has the *CapableOf* relation while "penguin" does not in Figure 1). Therefore, it is necessary to identify when a generality in a CKB can and cannot be applied to specific examples (e.g., which types of bird are *CapableOf* "fly"), in order for machines to effectively use CKBs as a source of rich commonsense knowledge.

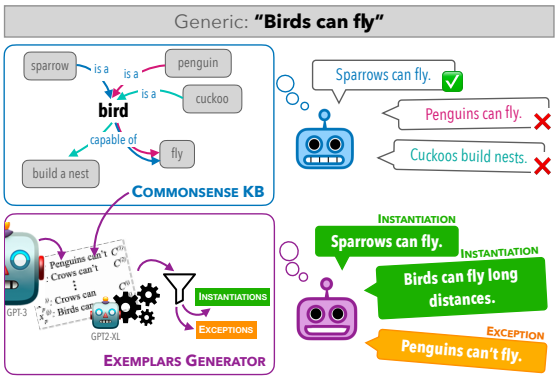

Figure 1: Many commonsense KBs contain tuples that capture generalized knowledge through concept-to-concept relationships. However, inferences derived over multiple such tuples can sometimes lead to incorrect statements like (e.g., "Penguins can fly.") To facilitate reasoning over commonsense knowledge, we use generics like "Birds can fly" to generate truthful statements where the generic holds (INSTANTIATIONS) and where it does not hold (EXCEPTIONS).

The generality on a CKB edge can often be expressed as a *generic*: a relationship between a *concept* ("Birds") and a *property* ("fly") without a quantifier[1] signaling prevalence of the property with respect to the concept (Figure 1). Generics have been extensively studied in semantics, philosophy, and psychology for their puzzling properties, such as generalizing about an uncommon property (e.g., "Mosquitoes carry malaria.[2]" Krifka 1987; Cohen 1996). Furthermore, by definition, generics have both INSTANTIATIONS—cases where the specified relationship holds (e.g., "Sparrows can fly") and EXCEPTIONS—cases where it does not hold (e.g., "Penguins cannot fly") (collectively **EXEMPLARS**). Identifying and producing these EX-

[1]Explicit quantification (e.g., "*Most* birds can fly", "Birds *usually* fly") blocks exceptions and is excluded from this study.

[2]Approximately 7-9% of the females of the species *Anopheles* (one among 3500 species) transmit malaria (CDC).

EMPLARS is crucial for understanding the generalities in CKBs and human reasoning.

In this work, we present a novel computational framework for constructing and generating EXEMPLARS for a generic that incorporates various theories from semantics. Bringing together categories of generics (Leslie, 2007, 2008) and exceptions (Greenberg, 2007) (see categorization in Table 1), we use generics partially based on ConceptNet (Bhagavatula et al., 2022 in prep) and automatically generate 8429 EXCEPTIONS and 11771 INSTANTIATIONS. We conduct human evaluation of our output, as well as detailed analysis.

Recent advances in language modeling have been extremely successful at generating text for a range of tasks in a few-shot manner (e.g., GPT-3 (Brown et al., 2020)). However, such generation is both expensive and does not provide the degree of control necessary for this task. Therefore, in this work we present a novel constrained generation approach using the NeuroLogic Decoding algorithm (Lu et al., 2021) with output constraints derived from semantic theories. Our system both outperforms (by 12.5 precision points on average) and is more controllable than few-shot generation.

Our contributions are as follows : **(1)** we present a **novel framework** grounded in linguistic theory for representing generics and EXEMPLARS, **(2)** we present the first, to the best of our knowledge, **method** to automatically generate generic EXEMPLARS and show it **outperforms GPT-3**, and **(3)** we present analysis showing the **importance of explicit linguistic modeling** for this task and **the insufficiency of current NLI** methods for representing default inheritance reasoning.

Our system and data will be publicly available.

2 Related Work

Generics have been studied extensively in semantics, philosophy, and psychology to develop a single logical form for all generics (Lewis and Keenan, 1975; Carlson, 1977, 1989; Krifka, 1987) or a probabilistic definition (Cohen, 1996, 1999, 2004; Kochari et al., 2020), categorize generics (Leslie, 2007, 2008), and analyze specific types (Prasada and Dillingham, 2006, 2009; Haward et al., 2018; Mari et al., 2012; Krifka et al., 2012). Mechanisms to tolerate EXCEPTIONS have also been proposed (Kadmon and Landman, 1993; Greenberg, 2007; Lazaridou-Chatzigoga and Stockall, 2013) but these are primarily theoretical and use carefully chosen examples. In contrast, our work combines these EXCEPTION tolerance mechanisms with generic categorization and proposes a novel, large-scale, computational framework for EXEMPLARS.

While large-scale CKBs capture a range of commonsense knowledge (Speer et al., 2017; Sap et al., 2019; Forbes et al., 2020; Bhakthavatsalam et al., 2020; Hwang et al., 2021), they contain necessarily incomplete (i.e., the open-world assumption (Reiter, 1978b)) *general* knowledge. Furthermore, since generalities can be made about both common and uncommon phenomena, the generalities in CKBs are not indicative of high real-world prevalence. In order to remedy these shortcomings, in our work we categorize generics (i.e., provide an approximation of frequency) and generate instances when a generality can and cannot be applied to subtypes or individuals.

The application of generics to specific individuals is influenced by prototypicality (Rips, 1975; Osherson et al., 1990), with small sets of prototypical norms collected in cognitive science for a range of kinds (Devereux et al., 2014; McRae et al., 2005; Overschelde et al., 2004). However, recent work has shown that neural models have only moderate success at mimicking human prototypicality (Misra et al., 2021; Boratko et al., 2020) or producing commonsense facts without guidance (Petroni et al., 2019) and additionally exceptions are often not prototypical. Hence, we combine neural models with a KB of concepts, using linguistic-theory-guided decoding, to generate generics EXEMPLARS.

Reasoning with generics is closely related to non-monotonic reasoning (Ginsberg, 1987b,a); specifically default inheritance reasoning (Brewka, 1987; Hanks and McDermott, 1986; Horty and Thomason, 1988; Imielinski, 1985; Poole, 1988; Reiter, 1978a, 1980). Contrary to the proposed solutions for linguistic tests on default inheritance reasoning (Lifschitz, 1989) (e.g., can a conclusion about inheritance be inferred based on provided evidence?), later works showed that the presence of generics EXEMPLARS in the evidence impacts what humans perceive as the correct answer (Elio and Pelletier, 1996; Pelletier and Elio, 2005; Pelletier, 2009). These results highlight the importance of identifying generics and analyzing how to accurately model their relationships in machine reasoning. While natural language inference (NLI), a form of deductive reasoning that has been well studied in NLP (i.a., Dagan et al. (2013); Bowman

	Categories	Template	
EXCEP.	quasi-def & char.	$[\mathtt{K} + r]^{input}\ [\sim\mathtt{P}]^{comp}$	t1
		$[\mathtt{K}_{sub} + r]^{input}\ [\sim\mathtt{P}]^{comp}$	t2
	principled & char.	$[\mathtt{K} + \neg r]^{input}\ [\mathtt{P}_{sub}]^{comp}$	t3
		$[\mathtt{K}_{sub} + \neg r]^{input}\ [\mathtt{P}]^{comp}$	t4
INST.	all	$[\mathtt{K}_{sub} + r]^{input}\ [\mathtt{P}]^{comp}$	t5
		$[\mathtt{K} + r]^{input}\ [\mathtt{P}_{sub}]^{comp}$	t6
		$[\mathtt{K}_{sub} + r]^{input}\ [\mathtt{P}_{sub}]^{comp}$	t7

Table 2: Templates for generating EXEMPLARS, derived from their logical forms (§3.3). *sub* indicates a subtype, K the concept, P the property, $\sim$P its pragmatic negation (§3.2). *comp* is the completion.

produce tsunamis" for the generic "Quakes produce seismic waves"). Therefore, an EXCEPTION satisfies $\sim L_G$ (i.e., $\neg L_G$ where $\neg T$ is replaced by $\sim T$). We define $\sim T$ to be the **pragmatic negation** of type T. That is, $\sim T$ is satisfied by i which does *not* satisfy T but is contextually relevant to T within generic G. By requiring contextual relevance, the EXCEPTION should be informative about G. For example, if $\sim L_G$ is

$$\text{QUAKE}(x) \wedge \textit{produce}(x, y) \wedge \sim\text{SEISMICWAVE}(y)$$

then (x, y) = ("quakes", "tsunamis") satisfies $\sim L_G$, while the tuple (x, y) = ("quakes", "conspiracy theories") does not: it is not informative about G. The logical form of the EXCEPTION depends on the type of the generic (see Table 1).

3.4 Logical Forms to Templates

Based on our proposed formulae (Table 1) for EXEMPLARS we define seven templates for generation (Table 2). Each template represents an instance that satisfies the logical form of an EXEMPLAR, potentially with subtypes. Each template consists of two sets of content requirements: for the *input* and for the *completion* (i.e., the decoder output).

For INSTANTIATIONS, we define three templates with subtypes of the concept, property, or both. However, for EXCEPTIONS we subtype only the concept *or* property to avoid pragmatically irrelevant or uninformative instances. For example, for the generic "Birds can fly", "Penguins can't fly long distances" doesn't mean penguins can't fly *short distances* (i.e., they might still be able to fly).

4 Methodology

We propose a pipeline system to automatically generate generics EXEMPLARS. Our system takes as input a generic G and the templates derived from its type[3] (§3.4), and outputs a set of generated EXEMPLARS (Figure 2). First, the system **assembles** and populates the templates according to the input generic (§4.1, Figure 2). Then, filled templates are converted into prompts and constraints that control the **generation** decoding process (§4.2). Finally, the output is **filtered** to remove non-viable (§4.3, Figure 2) or pragmatically irrelevant (§4.4, Figure 2) EXEMPLARS.

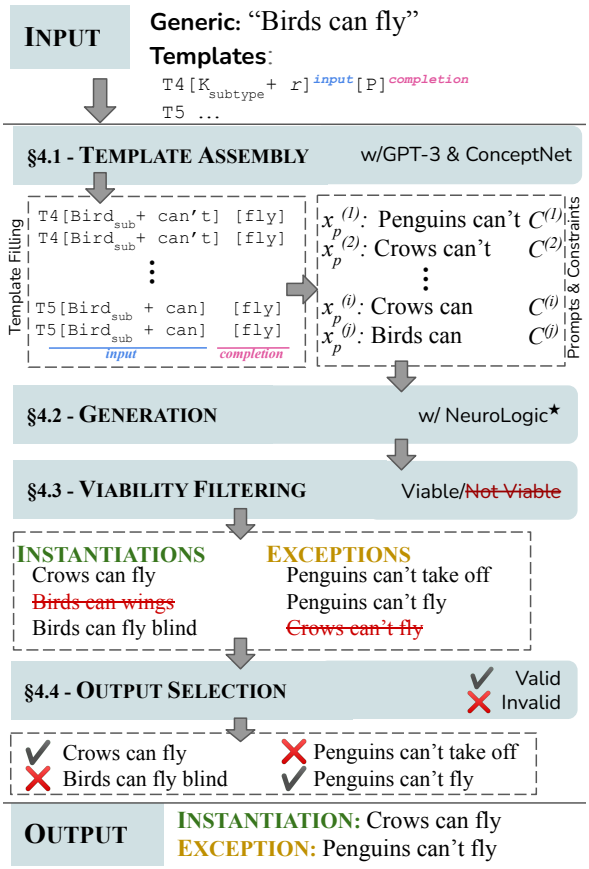

Figure 2: Overview of our method for an input generic.

4.1 Template Assembly

To populate our templates (defined Table 2), we use a dependency parser[4] to identify text spans for the concept, relation, and property in a generic. Then, **(i)** we extract subtypes for the concept and property and use these to populate **(ii)** the input template (i.e., generation prompts x_p) and **(iii)** the completion template (i.e., lexical constraints $\mathcal{C}$).

(i) Subtype Extraction We extract subtypes using both ConceptNet (Speer et al., 2017)[5] and GPT-3 (Brown et al., 2020). GPT-3 increases the coverage and diversity of subtypes, since many natural and valid subtypes may be missing from Concept-

[3] We assume the generic's type is known.

[4] https://spacy.io/

[5] Relations: IsA, InstanceOf, Synonym

Net (e.g., modifier phrases attached to a concept: "*young* Arctic fox"). We only use GPT-3 for subtypes of the concept, since by increasing the diversity in the prompt we may encourage diversity in the generated properties (see details Appendix E).

(ii) Input Template We populate the input template by constructing generation prompts. Following the template, each prompt consists of either the concept (or a subtype) and the relationship (or its negation) (see Table 2). To each prompt, we additionally prepend the generic itself and a connective (e.g., "however"). We rank the prompts by perplexity and use the top k_p prompts for generation.

(iii) Completion Template Following the templates, we want to constrain the generation output to describe the property (or a subtype) or its pragmatic negation (see Table 2). We construct a set of completion constraints (e.g., $\mathcal{C}^{(i)}$ in Figure 2 specifies "fly" should be in the completion) using lexical items including subtypes, synonyms, and morphological derivations.

4.2 Generation

In order to generate output that has a specific pragmatic relation to the input without requiring training, we use the NeuroLogic* (Lu et al., 2021) decoding algorithm. NeuroLogic* is an unsupervised decoding algorithm that takes as input a prompt x_p and set of lexical constraints $\mathcal{C}$ and produces a completion of the prompt $\hat{y}$ which has high likelihood given the prompt *and* high satisfaction of the constraints (estimated throughout the decoding). A lexical constraint consists of a set of n-grams $w = (w_i^1, \ldots, w_i^m)$ and is satisfied when at least one $w_i \in w$ is in $\hat{y}$ (inclusion constraints) or is not in $\hat{y}$ (exclusion constraints).

By using the input prompts (as x_p) and completion constraints (as $\mathcal{C}$) derived from our templates (§4.1), we can control the output content, syntactic form, *and* pragmatic relevance. We note that since we cannot concretely define the set of relevant potential candidates for a property's pragmatic negation (§3.2), decoding constraints *must* be used to generate EXCEPTIONS.

Output Ranking We rank the outputs from NeuroLogic* by template and prompt and we take the top k_r outputs as potential EXEMPLARS. The outputs are ranked by perplexity (for fluency) and by the probability of a specific NLI label (for relevance) and we average the two ranks. For NLI labels, we hypothesize that a good EXCEPTION aligns with NLI's contradiction, as does a good INSTANTIATION with entailment (see Figure 2). While this alignment is useful for ranking, the relationship between the EXEMPLARS and NLI labels is not this straightforward in reality, as we will discuss (§6.2). Note that ranking only by perplexity could limit the diversity of the output, since small variations (e.g., word order changes) may result in multiple similar outputs ranked highly, and could also result in non-salient outputs (e.g., output "Hats can be made of many materials" for the generic "Hats are made of wool") ranked highly.

4.3 Filtering For Viability

Since pre-trained language models have a tendency to hallucinate facts (Rohrbach et al., 2018) or produce non-specific output (e.g., "Birds can do things"), we apply a viability filter to the ranked output generations. Specifically, we train a discriminator to predict whether an output is viable (i.e., true *and* sufficiently specific that it could be an EXEMPLAR) or not, using human annotated examples (see Appendix C for details). Generations predicted not viable by the trained discriminator are removed from the dataset.

4.4 Output Selection

After removing the non-viable generations, our final task is to select the generations that are pragmatically relevant (i.e., **valid**; correctly follows a template) EXEMPLARS. To do this, we collect gold labels from humans for whether an EXEMPLARS is valid. These annotations produce two sets of binary labels; one set each for INSTANTIATIONS and EXCEPTIONS.[6] Then, we train two validity discriminators: one for EXCEPTIONS, one for INSTANTIATIONS. The trained validity discriminators are used to rank and select the best generations for each generic as our system output.

5 Experiment Details

We discuss our experimental setup, with full hyperparameters in Appendix D.

5.1 Data Source

We use a subset of the GenGen dataset (Bhagavatula et al., 2022 in prep), a set of 30K generics

[6] Since a generation that is not an INSTANTIATION *in not necessarily* an EXCEPTION (and vise versa), these cannot be directly combined into a single multi-class labeling task.

built upon common everyday concepts (e.g., "hammers") and relations (e.g., "used for") sourced from resources such as GenericsKB (Bhakthavatsalam et al., 2020) and ConceptNet (Speer et al., 2017). The dataset includes a diverse variety of concepts, including general knowledge ("Dogs bark"), locative generics ("In a hotel, you will find a bed"), and comparative generics ("Cars are faster than people"). We use 653 generics from GenGen, excluding human referents as the concept (e.g., nationalities, professions) due to concerns of social biases.

5.2 Annotations

All annotations are done using Amazon Mechanical Turk with three annotators per HIT (paid at $15/hour on average) and processed using MACE (Hovy et al., 2013) to filter annotators and determine the most likely label. We note that while all tasks achieve moderate inter-annotator agreement, the complex pragmatics of generics make these tasks difficult for human annotators.

For **generic type** (§3.1), we conduct two annotation passes to partition *all 653 generics* into the three groups in Table 1. The Fleiss' κ (Fleiss, 1971) is 0.41 and 0.58 for the first and second pass respectively. Our categorization results in 296 quasi-definitional, 125 principled, and 232 characterizing generics. For the **viability filter** (§4.3), we annotate a set of 7665 *system generations* from 150 generics. The Fleiss' κ is 0.53.

For **EXEMPLAR gold labels** (§4.4), we use separate annotation tasks for INSTANTIATIONS and EXCEPTIONS (see Appendix C for details) with Fleiss' κ of 0.40 and 0.45 respectively. For training each discriminator, we randomly sample and annotate $\sim 1k$ system generations from $\sim$300 generics. For human evaluation (§6.1), the *top* 5 discriminator-ranked generations for *all generics* from both our system and the baseline are annotated.

5.3 Discriminators

For all discriminators, we fine-tune RoBERTa (Liu et al., 2019). All labeled data is split 80/10/10 into train/dev/test such that all generations for a particular generic are in the same data partition.

For our viability discriminators (§4.3), the accuracy on the test set is 75.2. The accuracies of the trained validity discriminators on their respective test sets are 77.4 for INSTANTIATIONS and 75.0 for EXCEPTIONS.

		Subtype Src		
		G3	CN	G3+CN
Generated	Output (§4.2)	42272	10496	52768
	Viable (§4.3)	22865	5452	28317
Valid (§4.4)	EXCEP.	6221	2208	8429
	INST.	10983	788	11771
	TOTAL	**17204**	**2996**	**20200**

Table 3: Statistics of the generated dataset, with GPT-3 (G3) and ConcepNet (CN) subtypes used.

5.4 Few-Shot Baseline

As a baseline for generation, we use GPT-3 (Brown et al., 2020) with few-shot prompting. Specifically, for each template (Table 2) we construct a few-shot prompt (Appendix E) that consists of three examples. Each example is two sentences: first the generic, second a connective (e.g., "But also") followed by an EXEMPLAR that adheres to the desired template. A fourth generic and connective is appended to the prompt and the model should then generate a completion that follows the illustrated template. This setup is very similar to the prompts to our system, except our system is not provided examples and GPT-3 is not provided with subtypes (when appropriate for the template).

Note that our goal is *not* to produce the best possible generations from GPT-3 but rather to show that constrained generation from GPT-2 (i.e., NeuroLogic*) outperforms (and is cheaper and more computationally feasible) than a natural use of GPT-3.

6 Evaluation

To evaluate our approach, we conduct a human evaluation (§6.1), as well as detailed analysis (§6.2). Our results show that our approach produces a large set of high quality generations for this difficult task. They also highlight current limitations in machine reasoning and potential directions for future work.

Observations Using our computational framework, we generate 20200 EXEMPLARS for 653 generics (Table 3). While close to half the output generations are untrue or not viable, the majority of viable generations are valid EXEMPLARS.

Example system generations are in Table 4. We observe that our system can successfully generate valid EXEMPLARS with subtypes of both the concept (e.g., "angina pectoris" vs. "a chest pain" in (b)) and the property (e.g., "small game" vs. "hunting" in (c)). Furthermore, it produces valid EXCEPTIONS with both the simpler relation-negation tem-

	Generic	INSTANTIATION	EXCEPTION
(a)	"Bleaches may be used to whiten the teeth."	"non-toxic bleaches can be used to remove discoloration" (t7)	"A bottle of liquid bleach should not be used to whiten the teeth" (t4)
(b)	"A chest pain has a physical cause."	"an angina pectoris has an underlying cause" (t5)	"a chest pain has an emotional or psychological origin" (t1)
(c)	"A gun are used for hunting."	"a shotgun is used for small game" (t7)	"semiautomatics can be used for target practice" (t2)

Table 4: Examples of generated INSTANTIATIONS and EXCEPTIONS. The template used in the prompt for generation is indicated in parentheses (see Table 2).

plates (i.e., templates t3/t4; see (a)) *and* with relevant pragmatic negations (i.e., templates t1/t2; see (b) and (c)). These highlight the success of our system in producing high-quality EXEMPLARS.

6.1 Human Evaluation

To quantitatively evaluate our system, we compute precision at k (for $k = 1$ and $k = 5$) using our human-annotated judgements (§5.2) (Table 5).

Our model outperforms the few-shot baseline (i.e., GPT-3) in all cases, and by a large gap (average 12.5 points). This is especially significant for EXCEPTIONS, which are more challenging to generate than INSTANTIATIONS, and where the baseline performance is close to random. Since generics are defaults, it follows that INSTANTIATIONS should be easier to produce than EXCEPTIONS. The fact that more generated INSTANTIATIONS are true (71% versus 40%) and more true INSTANTIATIONS are accepted by the discriminator (73% versus 36%), compared to the EXCEPTIONS, supports this intuition. Hence, the large improvements by our model over the baseline are significant towards generating these difficult EXCEPTIONS.

Additionally, we examine our model performance across templates. Specifically, we compute the fraction of generations for a template that annotators label as valid, using the same number[7] of generations for both models for a specific template (Table 6). We see that not only does our model outperform the baseline for the majority of templates, these templates constitute the majority of the generations ('#Gens' in Table 6).

Note that the performance comparison by template does not account for generations that are accepted *because* they do not adhere to the desired template. Therefore, we conduct a manual analysis of the best 40 baseline (i.e., GPT-3) generations per template, ranked by perplexity. For EXCEPTIONS, the baseline produces on average only 2.5/40 generations that fit the desired templates for t2-t4. Additionally, for the one EXCEPTION template, t1, where most baseline generations fit the template (37/40), our model still outperforms the baseline. For INSTANTIATIONS, the baseline performs slightly better (average 10/40 fitting generations) but still poorly. From this we observe that not only is the baseline not controllable, our model outperforms the baseline in cases when it does adhere to output requirements.

[7]The models produce similar numbers of generations on all templates except t5.

	EXCEPTIONS		INSTANTIATIONS	
	$P@1$	$P@5$	$P@1$	$P@5$
GPT-3	0.515	0.557	0.762	0.686
Ours	**0.615**	**0.595**	**0.909**	**0.876**

Table 5: Precision at k ($P@k$).

	EXCEPTIONS				INSTANTIATIONS		
	t1	t2	t3	t4	t5	t6	t7
#Gens	429	970	36	203	1159	58	890
GPT-3	0.64	**0.52**	**0.56**	0.52	0.78	**0.71**	0.50
Ours	**0.69**	**0.52**	0.42	**0.59**	**0.86**	0.62	**0.86**

Table 6: Precision by template. #Gens is per template and is minimum of the models.

6.2 Discussion

Does controllability matter? We ablate the decoding algorithm by removing the constraints (i.e., using beam search) (Table 7a). Although both systems condition their outputs on the same prompts, NeuroLogic*, with linguistic-theory-guided constraints, produces over seven times as many unique generations as unconstrained decoding (i.e., beam search). Additionally, the proportion of valid generations (i.e., accepted by our discriminators) is nearly twice as many for NeuroLogic*. This illustrates the importance of incorporating linguistic-theory-based control into decoding in order to generate a large set of unique, and valid, EXEMPLARS.

Do CKBs contain sufficiently rich information? We probe whether a CKB (i.e., ConceptNet) con-

	Beam		NeuroLogic*	
	#Gens	%Val	#Gens	%Val
EXCEP.	5119	9.7	30060	14.4
INST.	2185	38.0	22708	51.8
ALL	7304	18.2	52768	30.5

(a) Decoding method ablation: beam search vs. NeuroLogic*.

	MLM		CN		G3	
	#Gens	%Val	#Gens	%Val	#Gens	%Val
EXCEP.	10350	18.7	7619	12.6	22441	15.0
INST.	4459	59.7	2877	27.4	19831	55.4
ALL	14809	31.0	10496	16.7	42272	33.9

(b) Subtype ablation: MLM, ConceptNet (CN), GPT-3 (G3).

Table 7: Ablation results. #Gens: generations after ranking and filtering. %Val: percent accepted by the corresponding validity discriminator.

	EXCEP.		INST.	
	$P@1$	$P@5$	$P@1$	$P@5$
Ours	0.615	0.595	0.909	0.876
+ NLI-neu	0.524	0.532	0.906	**0.889**
+ NLI-sim	**0.808**	**0.775**	0.862	0.860
+ NLI-neu-sim	0.620	0.532	**0.910**	0.888

Table 8: Precision at k with NLI label filtering. NLI-sim is contradiction for EXCEP., entailment for INST.

tains sufficiently rich type information to produce EXEMPLARS. Specifically, we vary the source of subtypes in the template-based prompts and constraints for our system, comparing ConceptNet (CN) to extracting commonsense knowledge from language models (i.e., from GPT-3 prompting (G3) and GPT-2 masked-language model (MLM) (Devlin et al., 2018; Taylor, 1953) infilling).

We observe that, in fact, CN subtypes result in the fewest generations, with the lowest proportions valid (Table 7b). In contrast, using GPT-3 for subtypes produces the most generations. Although using MLM for subtypes produces fewer generations than using GPT-3, the proportion of valid generations is comparable and hence MLM could be used as a substitute if using GPT-3 is not feasible. This shows that while CKBs such as ConceptNet are a good source of generalities, producing EXEMPLARS requires knowledge that may not always be encoded within the CKB. Therefore, generating EXEMPLARS is important for accessing relevant knowledge beyond what is in CKBs and enabling tools that can effectively use CKBs in reasoning.

Does NLI impact EXEMPLARS? Since generics EXEMPLARS are closely related to default inheritance (i.e., nonmonotonic) reasoning, NLI is a natural task with which to investigate machine reasoning about EXEMPLARS. Thus, we examine whether controlling the NLI relation between generics and EXEMPLARS improves precision. Specifically, we compute the NLI label between the generic (premise) and EXEMPLAR (hypothesis) and exclude generations that do not have a specific predicted NLI label: 'contradiction' for EXCEPTIONS and 'entailment' for INSTANTIATIONS (NLI-sim), or 'neutral' (NLI-neu), or NLI-sim and 'neutral'. We find that by controlling the NLI relation, we improve precision for EXCEPTIONS by 19.3 points (Table 8). However, for INSTANTIATIONS NLI label filtering has a negligible impact on precision. Therefore, we observe that controlling NLI relations can improve EXCEPTION quality but is less beneficial for INSTANTIATIONS. Additionally, note that the alignment with NLI labels is not actually as straightforward as observed, which we discuss next.

Can NLI sufficiently represent EXEMPLARS? Although we observe an alignment between *predicted* NLI labels and EXEMPLARS, this actually indicates systematic NLI-model errors, deriving from the insufficiency of NLI schema for capturing the nuances of generics EXEMPLARS.

Consider the sentences in Figure 3, relating to the generic "Birds can fly". We see that *only false* statements (i.e., not EXCEPTIONS) can contradict the generic as premise, since the lack of explicit quantification does not preclude the existence of exceptions to the generic. Therefore, EXCEPTIONS *should actually be labeled neutral* by NLI with respect to the generic, since EXCEPTIONS are not "unlikely to be true given the information in the premise" (Dagan et al., 2013) (i.e., NLI contradictions). With INSTANTIATIONS, we observe that the NLI relationship may be *either* neutral or entailment. These theorized alignments, coupled with our prior observations about EXEMPLARS and predicted NLI labels, highlight the complicated relationship between NLI and EXEMPLARS and the systematic errors made by NLI models when presented with such pairs involving default inheritance reasoning.

Additionally, the examples in Figure 3 highlight that the NLI neutral label does not distinguish between statements that are true but not entailed or contradictory (e.g., "Penguins cannot fly") and statements that may not even be true (e.g., "This bird can fly"). Our generics EXEMPLARS

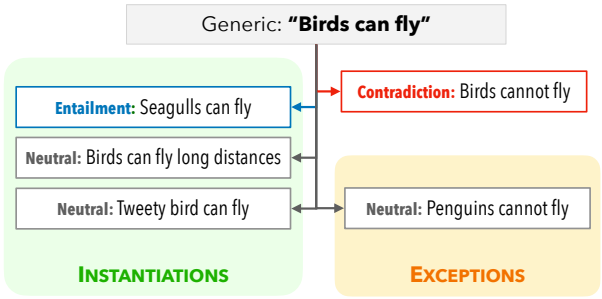

Figure 3: Example generic with EXEMPLARS and correct NLI labels.

emphasize the importance of developing a more fine-grained notion of NLI to model this default inheritance reasoning.

7 Conclusion

In this work, we draw on insights from linguistics to propose a novel computational framework to automatically generate valid EXEMPLARS for generics, as a step towards capturing the nuances of human reasoning for generics. Our system generates $\sim 20k$ EXEMPLARS for 653 generics and outperforms GPT-3 at generating viable examples, while remaining more controllable. We also demonstrate the limitations of CKBs and the importance of explicit linguistic modeling in generating EXEMPLARS. That is, the importance of linguistic-theory-based decoding and semantics-based filtering with NLI. Finally, we highlight the inability of current NLI models to reason about and represent the default-inheritance-reasoning relationship between generics and EXEMPLARS.

References

Chandra Bhagavatula, Jena D. Hwang, Keisuke Sakaguchi, Ronan Le Bras, Qin Lianhui, Peter West, Ximing Lu, Doug Downey, and Yejin Choi. 2022 in prep. Neuro-symbolic generic induction [unpublished manuscript].

Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. 2020. Genericskb: A knowledge base of generic statements. *ArXiv*, abs/2005.00660.

Michael Boratko, Xiang Lorraine Li, Rajarshi Das, Timothy J. O'Gorman, Daniel Le, and Andrew McCallum. 2020. Protoqa: A question answering dataset for prototypical common-sense reasoning. *ArXiv*, abs/2005.00771.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Gerhard Brewka. 1987. The logic of inheritance in frame systems. In *IJCAI*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Greg N. Carlson. 1977. Reference to kinds in english. In *Ph.D. dissertation, University of Massachusetts, Amherst*.

Greg N. Carlson. 1989. On the semantic composition of english generic sentences. In Gennaro Chierchia, Barbara H Partee, and Raymond Turner, editors, *Properties, Types and Meaning, Vol. II. Semantic Issues*. Dordrecht: Kluwer.

CDC. About malaria. https://www.cdc.gov/malaria/about/biology/index.html. Accessed: 2022-01-15.

Ariel Cohen. 1996. *Think generic! The meaning and use of generic sentences*. Carnegie Mellon University.

Ariel Cohen. 1999. Generics, frequency adverbs, and probability. *Linguistics and philosophy*, 22(3):221–253.

Ariel Cohen. 2004. Generics and mental representations. *Linguistics and Philosophy*, 27(5):529–556.

Robin Cooper, Richard Crouch, Jan Van Eijck, Chris Fox, Josef Van Genabith, Jan Jaspers, Hans Kamp, Manfred Pinkal, Massimo Poesio, Stephen Pulman, et al. 1994. Fracas: A framework for computational semantics. *Deliverable D6*.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220.

Barry Devereux, Lorraine K. Tyler, Jeroen Geertzen, and Billi Randall. 2014. The centre for speech, language and the brain (cslb) concept property norms. *Behavior Research Methods*, 46:1119 – 1127.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Renée Elio and Francis Jeffry Pelletier. 1996. On reasoning with default rules and exceptions. In *Proceedings of the 18th conference of the Cognitive Science Society*, pages 131–136.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

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 *EMNLP*.

M. Ginsberg. 1987a. Introduction. Morgan Kaufmann, Los Altos, CA.

Matthew L. Ginsberg. 1987b. Readings in nonmonotonic reasoning. In *AAAI*.

Yael Greenberg. 2007. Exceptions to generics: Where vagueness, context dependence and modality interact. *Journal of Semantics*, 24(2):131–167.

Steve Hanks and Drew McDermott. 1986. Default reasoning, nonmonotonic logics, and the frame problem. In *AAAI*.

Paul Haward, Laura Wagner, Susan Carey, and Sandeep Prasada. 2018. The development of principled connections and kind representations. *Cognition*, 176:255–268.

John F. Horty and Richmond H. Thomason. 1988. Mixing strict and defeasible inheritance. In *AAAI*.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with mace. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. In *AAAI*.

Tomasz Imielinski. 1985. Results on translating defaults to circumscription. In *IJCAI*.

Nirit Kadmon and Fred Landman. 1993. Any. *Linguistics and philosophy*, 16(4):353–422.

Arnold Kochari, Robert Van Rooij, and Katrin Schulz. 2020. Generics and alternatives. *Frontiers in Psychology*, 11:1274.

Manfred Krifka. 1987. An outline of genericity. Seminar für natürlich-sprachliche Systeme der Universität Tübingen.

Manfred Krifka et al. 2012. Definitional generics. *Genericity*, pages 372–389.

Dimitra Lazaridou-Chatzigoga and Linnaea Stockall. 2013. Genericity, exceptions and domain restriction: experimental evidence from comparison with universals. In *Proceedings of Sinn und Bedeutung*, volume 17, pages 325–343.

Sarah-Jane Leslie. 2007. Generics and the structure of the mind. *Philosophical perspectives*, 21:375–403.

Sarah-Jane Leslie. 2008. Generics: Cognition and acquisition. *Philosophical Review*, 117(1):1–47.

Sarah-Jane Leslie. 2017. The original sin of cognition: Fear, prejudice, and generalization. *The Journal of Philosophy*, 114(8):393–421.

David Lewis and Edward L. Keenan. 1975. *Adverbs of quantification*, page 3–15. Cambridge University Press.

Vladimir Lifschitz. 1989. Benchmark problems for nonmonotonic reasoning. In *Proceedings of the Second international Workshop on Non-monotonic Reasoning*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, et al. 2021. Neurologic a* esque decoding: Constrained text generation with lookahead heuristics. *arXiv preprint arXiv:2112.08726*.

Alda Mari, Claire Beyssade, and Fabio Del Prete. 2012. *Genericity*, volume 43. OUP Oxford.

Ken McRae, George S. Cree, Mark S. Seidenberg, and Chris McNorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37:547–559.

Hugo Mercier and Dan Sperber. 2017. *The enigma of reason*. Harvard University Press.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Kanishka Misra, Allyson Ettinger, and Julia Taylor Rayz. 2021. Do language models learn typicality judgments from text? *ArXiv*, abs/2105.02987.

Daniel N. Osherson, Edward E. Smith, Ormond Wilkie, Alejandro López, and Eldar Shafir. 1990. Category-based induction. *Psychological Review*, 97:185–200.

James P. Van Overschelde, Katherine A. Rawson, and John Dunlosky. 2004. Category norms: An updated and expanded version of the battig and montague (1969) norms. *Journal of Memory and Language*, 50:289–335.

Francis Jeffry Pelletier. 2009. Are all generics created equal? *Kinds, Things, and Stuff: Mass Terms and Generics*, pages 60–79.

Francis Jeffry Pelletier and Renée Elio. 2005. The case for psychologism in default and inheritance reasoning. *Synthese*, 146(1):7–35.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? In *EMNLP*.

David L. Poole. 1988. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47.

Sandeep Prasada and Elaine M Dillingham. 2006. Principled and statistical connections in common sense conception. *Cognition*, 99(1):73–112.

Sandeep Prasada and Elaine M Dillingham. 2009. Representation of principled connections: A window onto the formal aspect of common sense conception. *Cognitive Science*, 33(3):401–448.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

R. Reiter. 1978a. On reasoning by default. In *Proceedings of TINLAP-2*, pages 210–218, University of Illinois. Association of Computational Linguistics.

Raymond Reiter. 1978b. *On Closed World Data Bases*, pages 55–76. Springer US, Boston, MA.

Raymond Reiter. 1980. A logic for default reasoning. *Artificial Intelligence*, 13:81–132.

Lance J. Rips. 1975. Inductive judgments about natural categories. *Journal of Verbal Learning and Verbal Behavior*, 14:665–681.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics.

Rachel Rudinger, Vered Shwartz, Jena D Hwang, Chandra Bhagavatula, Maxwell Forbes, Ronan Le Bras, Noah A Smith, and Yejin Choi. 2020. Thinking like a skeptic: Defeasible inference in natural language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4661–4675.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *AAAI*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *"Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019a. HELP: A Dataset for Identifying Shortcomings of Neural Models in Monotonicity Reasoning. *ArXiv*, abs/1904.12166.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019b. Can Neural Networks Understand Monotonicity Reasoning? In *BlackboxNLP@ACL*.

A Limitations and Risks

The generics we source (see §5.1) are exclusively in English. Therefore, our approach may not be suited to all possible generics in all languages. In particular, our system does not handle generics where valid INSTANTIATIONS include negating (§3.2) the concept. This is due to the restriction that most English generation is left-to-right and it is not possible to define a closed set of possible concept negations for the prompt.

In this work, we do not generate EXEMPLARS for generics involving human referents (e.g., professions, nationalities). We exclude generics involving human referents to mitigate the risk of generating socially biased EXEMPLARS or harmful stereotypes (e.g., "Black folks go to jail for crimes" for the generic "People go to jail for crimes"). Additionally, handling of human stereotypes require methods that are beyond the scope of this paper. For example, a socially-aware EXCEPTIONS to a generic like "Girls wear dresses" would be "Boys wear dresses, too". This would require the understanding of the possible subtext of such a statement (e.g. "Only girls wear dresses"), which is beyond the current capabilities of this study and worthy of future exploration.

Finally, we note that while it is not the intended purpose of our system, a malicious user could still use our system to generate EXEMPLARS for a generic involving a person and propagate potentially harmful social biases.

B Generics Definitions

We condense the five generic types proposed by Leslie (2007, 2008) into our three categories (§3.1). The five types are:

- **Quasi-definitional**: generics concerning properties that are assumed to be universal among a concept. This is the same as our quasi-definitional category, see (a) Table 1. The property is considered a defining characteristic of the concept.
- **L-Principled**: generics concerning properties that are prevalent among a concept and are viewed as inherent, or connected in a principled way (Prasada and Dillingham, 2006, 2009; Haward et al., 2018). These generics are called principled in Leslie (2007, 2008). Note, these generics make up only one half of our "principled" category (§3.1). See first example for category (b) in Table 1; the second example there does *not* fit Leslie (2007, 2008)'s definition of principled (i.e., L-principled).
- **Striking**: generics describing properties that are uncommon and often dangerous, and members of the concept are *disposed* to possess them if given the chance (Leslie, 2017). For example, the striking generic "Sharks attack swimmers" assumes all sharks are capable of attacking swimmers. These generics constitute the second half of our "principled" category. See second example (not first) for category (b) in Table 1.
- **Majority characteristic**: generics concerning properties that are neither deeply connected to the concept nor striking but occur in the majority of members of the concept. These constitute one half of our "characterizing category". See example for (c) in Table 1.
- **Minority characteristic**: generics concerning properties that are neither deeply connected to the concept nor striking but occur in the minority of members of the concept. For example, "Lions have manes", since only adult male lions (the minority of the lion population) have manes. These constitute the second half of our "characterizing category".

Both L-principled and striking generics are true in-virtue-of a secondary factor and therefore we group these into one category (i.e., "principled"; see §3.1). For L-principled generics, this may be a factor that causes the property to occur in the concept (e.g., Birds can fly because they have wings). For striking generics, it is the assumed predisposition of the kind to possess the property if given the chance.

For quasi-definitional generics, because the property is considered defining to concept, there is no implied secondary factor in-virtue-of which the generic is true. Therefore, these generics are descriptive and we put them in a separate category from striking and L-principled generics.

Finally, majority and minority characteristic generics are ambiguous in their interpretation. For example, "Lions have manes" can be interpreted as being true in-virtue-of some secondary factor (e.g., as a signal of fitness) or as being a merely accidental relationship. If the interpretation is the former, then lions without manes are valid EXCEPTIONS (e.g., lion cubs, female lions), while if the

interpretation is the latter then then other attributes of lions are valid EXCEPTIONS (e.g., claws, fur).

Additionally, we note that a generic can focus on the presence of the property within the concept (e.g., "Birds can fly" is concerned with which birds can fly) **or** can focus on the presence of the concept within holders of the property (e.g., "Triangles have three sides" is more concerned with what concepts have three sides). We will say that the former kind of generic is *concept-oriented* and the latter is *property-oriented*. A generic can be both concept and property oriented if it is ambiguous between the two readings (e.g., "Aspirin relieves headaches").

In this work, we have discussed and used definitions only for concept-oriented generics. However, similar definitions and logical forms can be derived for property-oriented generics. Note that only the logical forms for quasi-definitional generics and their EXCEPTIONS change if the generic is property-oriented. In particular, the K and P in both logicals form for (a) in Table 1) can swapped to obtain the property-oriented versions. In this work, we do not deal with property-oriented generics and their EXEMPLARS due to the limitations of English generation, as mentioned in Appendix A.

C Annotation

For all annotation tasks, three annotators are used per HIT. When filtering annotators using MACE, we remove annotators with competence below 0.5 (or the median, if lower).

Generic Type Instructions for annotating generic types (§3.1) are shown Figure 4 (for the first pass) and Figure 5 (for the second pass). The first pass categorizes generics as either characterizing or not (either quasi-definitional or principled). The second pass categorizing non-characterizing generics as either quasi-definitional or principled.

Truthfulness Task Instructions for annotating output generations for truthfulness (§4.3) are shown in Figure 6.

EXEMPLARS Gold Labels For the INSTANTIATION template generations, annotators are asked whether the generation contradicts the original generic. Instructions are shown in Figure 8. However, for the exception template generations, an EXCEPTION is not a contradiction of the generic itself but of an associated logical form. For example, "Penguins cannot fly" does not actually contradict

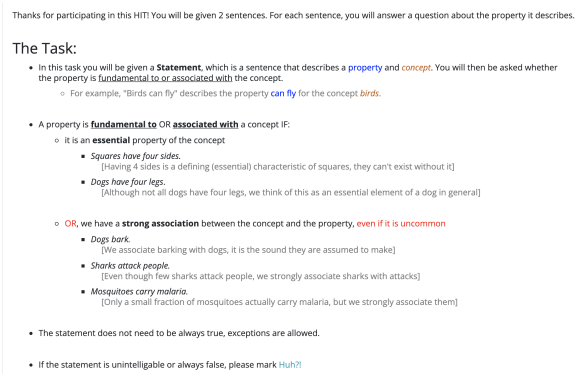

Figure 4: Task instructions for first part of the generic type categorization annotation (§5.2).

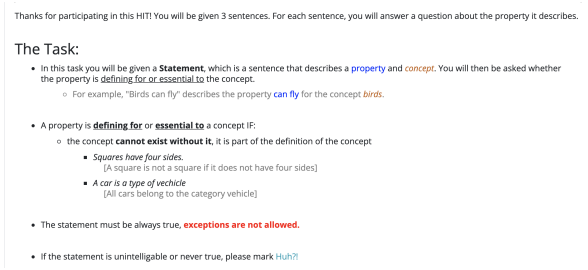

Figure 5: Task instructions for second part of the generic type categorization annotation (§5.2).

the generic itself ("Birds can fly") but a modified form of the generic involving quantification (i.e., "All birds can fly"). Therefore, we ask annotators whether the generation contradicts two modified forms of the generic. Instructions are shown in Figure 7.

We obtain modified forms of the generic by first converting the logical forms in Table 1 into a natural language templates by adding a universal quantifier. Then we apply the template to the generic itself. Specifically, from $K(x) \wedge r(x, y) \implies P(y)$ (e.g., for quasi-definitional generics) we derive "`[K] [REL] ONLY [P]`". For example, "mosquitoes drink *only* blood", which is contradicted by mosquitoes that drink something other than blood. Notice, that exceptions from templates 1 and 2 will contradict these statements. Similarly, for $K(x) \wedge P(y) \implies r(x, y)$ we derive "`ALL [K] [REL] [P]`". For example, "*All* birds can fly", which is contradicted by birds that cannot fly. Exceptions from templates 3 and 4 will contradict these statements.

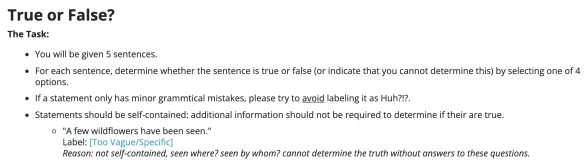

Figure 6: Task instructions for annotating truthfulness (§5.2).

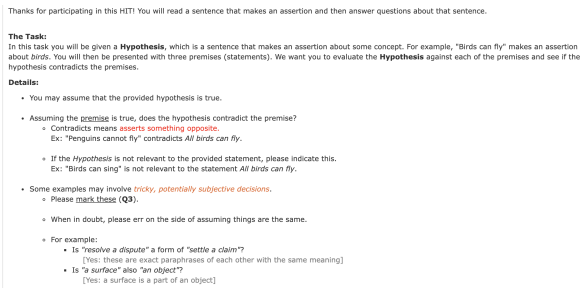

Figure 7: Task instructions for annotating validity of EXCEPTIONS (§5.2).

D Implementation Details

D.1 Data

We use the in-submission GenGen data (Bhagavatula et al., 2022 in prep). The dataset contains English generics, automatically generated via NeuroLogic* (Lu et al., 2021) with GPT2-XL. For this study, we source from the subset of GenGen's test set found to be valid by the discriminator with probability at least 0.5 (768 generics). Of these, we exclude all mentions of human referents (e.g., kinship labels, nationalities, titles, professions) and actions (e.g., studying for a test) to arrive at a dataset of 653 generics. We remove human referents using a seed set of human referent terms compiled based on WordNet (Miller, 1995) and will be provided with the system code. We remove mentions of actions by excluding generics beginning with "In order to". The GenGen dataset is licensed under CC-BY and our usage aligns with the intended use of the data.

Preprocessing We remove adverbs of quantification (i.e., usually, typically, generally) from the generics and exclude generics with verbs of consideration (i.e., consider, posit, suppose, suspect, think). We also convert hedging statements to more explicit forms (e.g., "may have to be" to "must be").

Partitions The data splits for training the truth discriminator and validity discriminators are shown in Table 9 and Table 10 respectively.

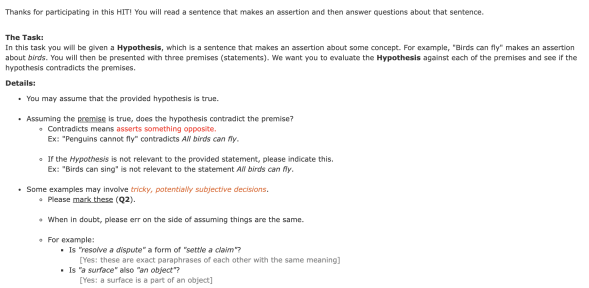

Figure 8: Task instructions for annotating validity of insts (§5.2).

	Train	Dev	Test	All
True	2831	412	433	3676
False/Non-salient	3180	367	442	3989
Total	6011	779	875	7665

Table 9: Data split statistics for truthfulness discriminator (§4.3).

D.2 Tools

For extracting components of the generic data we use spacy[8] for dependency parsing. We use *inflect*[9] to obtain plural and singular word forms and *mlconjug3*[10] to conjugate verbs. We use *nltk*[11] for additional synonyms.

D.3 Hyperparameters

To obtain subtypes from GPT-3 we use the *davinci* model and top-p sampling with $p = 0.9$, temperature 0.8 and maximum length 100 tokens. We use the top 5 sequences to obtain subtypes. For NLI scores, we use RoBERTa fine-tuned on MNLI (Williams et al., 2018) available from AllenNLP[12]. For the GPT-3 baseline we use the davinci model and top-p sampling 1.0, temperature 0.8, maximum length 50 tokens and top 5 sequences. Prompts for GPT-3 are given in Appendix E. GPT2-XL has 1.5 billion parameters, GPT-3 has 175 billion parameters. Our experiments are done using Quadro RTX 8000 GPUs.

For generation with NeuroLogic*, we use GPT2-XL (Radford et al., 2019) with a maximum length of 50 tokens and a beam size of 10 with temperature 10000000. We set the constraint satisfaction tolerance to 3. This means that at each step, only candidates whose number of satisfied constraints

[8] https://spacy.io/
[9] https://pypi.org/project/inflect/
[10] https://pypi.org/project/mlconjug3/
[11] https://www.nltk.org/
[12] https://demo.allennlp.org/textual-entailment/roberta-mnli

		Train	Dev	Test	All
EXCEPTION	Valid	342	35	35	412
	Invalid	462	72	53	587
	Total	804	107	88	999
INSTANTIATION	Valid	374	38	29	441
	Invalid	466	38	33	537
	Total	840	76	62	978

Table 10: Data split statistics for validity discriminators (§4.4).

is within three of the maximum so far are kept. The 'look ahead' is also set to 3; look ahead three generation steps during decoding to estimate future constraint satisfaction. During prompt construction, take the top $k_p = 10$ prompts. If the generic produced less than 10 prompts total, we take half so that low quality prompts are not used even if few are produced. After ranking the output, we keep the top $k_r = 10$ generations for a template, keeping at most 2 per prompt.

For the truth discriminator, we fine-tune the model for 5 epochs using a batch size of 16 and learning rate $1e-5$ and random seed 29725, selected by manual grid search.

For the validity discriminators, we fine-tune *the truth discriminator* for 3 epochs with a batch size of 16 and learning rate $3e-5$. The instantiation discriminator uses a random seed of 4427 and the exception discriminator 4457. Hyperparameters are again selected by manual grid search.

E GPT-3 Prompts

E.1 Subtyping

To obtain subtypes from GPT-3, we first categorize the kinds into six categories: person, animal, other living (e.g., plants), location, temporal (e.g., Thursday), and other (e.g., candle, soup) (Table 11). For each category, we construct a separate prompt for GPT-3 containing one type and five example subtypes. Then, for each kind we use the prompt from its assigned category to obtain subtypes. Note that we exclude all generics where the kind is "person". This is to avoid producing or repeating stereotypes.

To determine the category, we use seed lists, for person, animal, other living, and locative, or the presence of prepositional beginnings ("On", "In", "At", "During"), for locative and temporal. The "other" category encompasses all kinds that do not fit into another category.

E.2 Few-shot Baseline

The prompts for our few-shot baseline are shown in Table 12. The three examples in the table are provided each on a separate line. Appended to the prompt is a fourth generic and the necessary connective. The same connective is used across all exception (instantiation) templates and is chosen through manual experimentation. We use "But also" for EXCEPTIONS and "For example" for INSTANTIATIONS.

Category	Prompt Concept	Prompt Subtypes
Animal	birds	sparrow, canary, large bird,bird of prey, sea bird
Other living	apple tree	small apple tree, flowering apple tree,apple tree with ripe apples, granny smith apple tree, young apple tree
Locative	hotels	beach hotel, boutique hotel, resort, bed and breakfast, five star hotel
Temporal	day	morning, hot day, short day, afternoon, evening
Other	candles	scented candle, advent candle, tealight, candle made from beeswax, candle that smells floral
	can of soup	can of tomato soup, can of mushroom bisque, expired can of soup, unopened can of soup, organic can of soup

Table 11: Prompts for generating subtypes with GPT-3.

	Template	Prompt Examples
(1)	[KIND + REL]^{p} [NEG-PROP]$^{\mathcal{C}}$	Elephants are found in zoos. But also elephants are found in the wild in Africa. Viruses are spread through body fluids. But also viruses are spread in the air. A hair dryer is used to dry hair. But a hair dryer can also be used to dry clothes.
(2)	[KIND$_{sub}$ + REL]^{p} [NEG-PROP]$^{\mathcal{C}}$	Elephants are found in zoos. But also African elephants are found in the wild in Africa. Viruses are spread through body fluids. But also coronaviruses are spread in the air. A hair dryer is used to dry hair. But also an electric hair dryer can be used to dry clothes.
(3)	[KIND + NEG-REL]^{p} [PROP$_{sub}$]$^{\mathcal{C}}$	Dogs protect buildings from intruders. But also dogs do not protect apartment buildings from intruders. Cowsheds are found on farms. But also cowsheds are not found in orchards. The sun produces radiation. But also the sun does not produce x-rays.
(4)	[KIND$_{sub}$ + NEG-REL]^{p} [PROP]$^{\mathcal{C}}$	Birds can fly. But also penguins cannot fly. Ducks lay eggs. But also male ducks do not lay eggs. Dogs protect buildings from intruders. But also very small dogs do not protect buildings from intruders.
(5)	[KIND$_{sub}$ + REL]^{p} [PROP]$^{\mathcal{C}}$	Birds can fly. For example, seagulls can fly. Dogs protect buildings from intruders. For example, pitbulls protect buildings from intruders. Ducks lay eggs. For example, female ducks lay eggs.
(6)	[KIND + REL]^{p} [PROP$_{sub}$]$^{\mathcal{C}}$	Viruses are spread through body fluids. For example, viruses are spread through saliva. Dogs protect buildings from intruders. For example, dogs protect some private homes from intruders. Cowsheds are found on farms. For example, cowsheds are found on dairy farms.
(7)	[KIND$_{sub}$ + REL]^{p} [PROP$_{sub}$]$^{\mathcal{C}}$	Birds can fly. For example, Canadian geese fly long distances to migrate. Ostriches lay eggs. For example, female ostriches lay large spotted eggs. Elephants are found in zoos. For example, African elephants are found in most large zoos.

Table 12: Prompts for GPT-3 as Few-shot Baseline.