*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530


# Exceptions, Instantiations, and Overgeneralization: Insights into How Language Models Process Generics

Emily Allaway*
Columbia University

Chandra Bhagavatula
Allen Institute for AI

Jena D. Hwang
Allen Institute for AI

Kathleen McKeown
Columbia University

Sarah-Jane Leslie
Princeton University

*Large language models (LLMs) have garnered a great deal of attention for their exceptional generative performance on commonsense and reasoning tasks. In this work, we investigate LLMs' capabilities for generalization using a particularly challenging type of statement: generics. Generics express generalizations (e.g., birds can fly) but do so without explicit quantification. They are notable because they generalize over their* instantiations *(e.g., sparrows can fly) yet hold true even in the presence of* exceptions *(e.g., penguins do not). For humans, these generic generalization play a fundamental role in cognition, concept acquisition, and intuitive reasoning. We investigate how LLMs respond to and reason about generics.*

*To this end, we first propose a framework grounded in pragmatics to automatically generate both* exceptions *and* instantiations*– collectively* exemplars*. We make use of focus – a pragmatic phenomenon that highlights meaning-bearing elements in a sentence – to capture the full range of interpretations of generics across different contexts of use. This allows us to derive precise logical definitions for* exemplars *and operationalize them to automatically generate* exemplars *from LLMs. Using our system, we generate a dataset of* $\sim 370k$ exemplars *across* $\sim 17k$ *generics and conduct a human validation of a sample of the generated data.*

*We use our final generated dataset to investigate how LLMs' reason about generics. Humans have a documented tendency to conflate universally quantified statements (e.g., all birds can fly) with generics. Therefore, we probe whether LLMs exhibit similar overgeneralization behavior in terms of quantification and in property inheritance. We find that LLMs do show evidence of overgeneralization, although they sometimes struggle to reason about* exceptions*. Furthermore, we find that LLMs may exhibit similar non-logical behavior to humans when considering property inheritance from generics.*

* Columbia University, New York, NY, USA. Email: eallaway@cs.columbia.edu

Action editor: Marianna Apidianaki, Abdellah Fourtassi, and Sebastian Padó. Submission received: 17 December 2023; revised version received: 30 April 2024; accepted for publication: 19 June 2024.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




## 1. Introduction

Large language models (LLMs) have garnered a great deal of attention for their exceptional performance on a range of reasoning tasks, including commonsense reasoning. In this work, we investigate the ability of LLMs to reason about generalizations, using a particular type of statement that is fundamental to human reasoning: ***generics***. Generics express generalizations about the world (e.g., birds can fly) but do so without explicit quantification (e.g., without quantifiers such as "most", "some", "all"). Since generics generalize over their INSTANTIATIONS (e.g., sparrows can fly) while holding true even in the presence of EXCEPTIONS (e.g., penguins cannot fly), they are challenging to reason about. Our work investigates how LLMs reason about generics and how this compares to human capabilities.

For humans, generalization is a fundamental component of cognition, knowledge acquisition, and reasoning. And generics appear to be the default mechanism for this generalization; children acquire generics earlier than explicitly quantified statements and, along with adults, fall back on generics in cognitively challenging situations (Leslie 2007, 2008; Leslie and Gelman 2012; Meyer, Gelman, and Stilwell 2011; Hollander, Gelman, and Star 2002). One benefit for humans of generic generalizations is that they support flexible and efficient reasoning: they allow humans to reason with incomplete information and draw inferences in novel situations (Asher and Morreau 1995). However, despite this, there has been limited investigation in NLP on the capabilities of computational models to reason about generics. Since LLMs underlie most natural language reasoning systems, it is crucial to understand whether they have similar flexible reasoning abilities (e.g., about generics) to humans. Therefore, the goal of this work is to provide insights into how LLMs process generics.

The present investigation is carried out in two stages. In the first stage, we propose a new theoretically-grounded framework ***GenerIX*** that specifies logical-form based definitions for multiple interpretations of generics and their INSTANTIATIONS and EXCEPTIONS (collectively, henceforth **EXEMPLARS**). To incorporate contextually-sensitive interpretations into the definitions, *GenerIX* uses the notion of pragmatic ***focus***—a phenomenon that highlights meaning-bearing elements in a sentence and relates to discourse context. We also introduce ***ExempliFI***, a model that operationalizes the formal definitions to automatically generate EXEMPLARS from LLMs. Our focus-sensitive framework for generics provides us with flexibility and control over the generation of EXEMPLARS with *ExempliFI*. For example, "birds can fly" expresses a generalization about the kind birds, so EXCEPTIONS will be birds that cannot fly (e.g., penguins); with focus on BIRDS, "BIRDS can fly" emphasizes birds in contrast to other animals that can fly (e.g., bats), making those alternatives the EXCEPTIONS. By incorporating multiple focus interpretations, we use *ExempliFI* to generate a diverse dataset of $\sim 370k$ EXEMPLARS for $\sim 17k$ generics covering different usages. We validate the dataset's quality using human evaluation on a subset of the generated data. The generated EXEMPLARS include both knowledge-based (e.g., "ostriches can't fly") and reasoning-based outputs (e.g., "a bird with a broken wing cannot fly"). The increased quality and diversity of our EXEMPLARS dataset allows us to use the EXEMPLARS to probe LLM's reasoning about generics.

In the second stage of this work, we use our generated dataset to probe how LLMs reason about generics. In particular, we concentrate on two human phenomena, namely, *overgeneralization* and *inheritance reasoning*. The fundamental role of generics in human generalization has been demonstrated through studies on the **G**eneric **O**ver**G**eneralization (**GOG**) effect (Leslie, Khemlani, and Glucksberg 2011). The GOG effect is the documented tendency of humans to treat universally quantified statements (e.g., all birds can fly) as generics and therefore compatible with EXCEPTIONS (Khemlani et al. 2007; Meyer, Gelman, and Stilwell 2011; Leslie, Khemlani, and Glucksberg 2011), when in fact, from a logical point of view, a universally quantified statement cannot be true if there are exceptions to it. Therefore, using our EXEMPLARS dataset, we first probe

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




whether LLMs exhibit similar overgeneralization behavior when reasoning about quantification. We find that LLMs do show evidence of the GOG effect and EXCEPTIONS do not entirely eliminate this effect.

Generic generalizations play an important role for humans in reasoning about property inheritance (e.g., if Polly is a bird and we know birds can fly, can Polly fly?). Humans can draw such inferences from generics, and, when presented with counterexamples (e.g., Bob is a penguin and Bob cannot fly), may revise their conclusions to accommodate the new information (Elio and Pelletier 1996; Pelletier and Elio 2005). We use our EXEMPLARS dataset to probe whether LLMs exhibit similar behavior in reasoning about property inheritance. Our results show that while LLMs do make property inheritance inferences based on generics, they are less consistent about making adjustments in their reasoning when presented with new information.

Our contributions are as follows: **(1)** we propose a novel pragmatics-based computational framework to define and represent generics and EXEMPLARS; **(2)** we operationalize our framework and propose a system to automatically generate EXEMPLARS for a range of pragmatic and contextual interpretations of generics; **(3)** we generate a large scale, high quality dataset of EXEMPLARS, improving over prior work; **(4)** we investigate how LLMs reason about generics and show that LLMs exhibit similar non-logical behavior to humans when considering quantification and property inheritance. In the remainder of the paper, we first provide an overview and background on generics and EXEMPLARS (§2). Next, we discuss our pragmatically grounded framework *GenerIX* for EXEMPLARS (§3) and how we operationalize this in our system *ExempliFI* to automatically generate EXEMPLARS (§4). Then we present our investigations into how LLMs reason about generics (§5). Finally, we present the details of our generation system and the system validation results (§6).

## 2. Generics and EXEMPLARS: An Overview

Generics are statements that express generalizations about the world (e.g., "tigers are striped", "ducks lay eggs") and are notable for their lack of quantification. That is, a generic statement (e.g., "birds can fly") describes a *relation* between a *concept* and a *property* without explicit quantification (e.g., quantifiers such as "all" or adverbs of quantification such as "normally" or "usually"). Generics are challenging to analyze semantically for a number of reasons. First, the truth of a generic is not related to prevalence of the property. For example, "ducks lay eggs" is felicitous while "ducks are female" are is not, despite the relevant populations of ducks in both instances being nearly identical (Leslie and Lerner 2016). Secondly, the lack of quantification in generics allows them to have both **INSTANTIATIONS** (i.e., examples where the generic does apply) and **EXCEPTIONS** (i.e., counterexamples to the generic)—collectively **EXEMPLARS**.

### 2.1 Analyzing Generics

Generics have been extensively studied in semantics and philosophy with the goal of developing truth conditional semantic analyses (e.g., Lewis 1975; Carlson 1977, 1989; Krifka 1987). Specifically, these works aim to provide formal methods to determine the circumstances under which a generic is or is not true[1]. Many of these frameworks for analyzing generics propose a special generic operator, which has a similar role to quantifiers like "all". (Carlson 1977). However, it is not clear what the precise semantics of this operator should be. While generics like "a cat

1 There is debate about whether generics should even have truth values (cf. Krifka et al. 1995). We do not take a side in this debate and instead we use "true" for both the formal meaning (i.e., having a truth value) and the less formal meaning (i.e., acceptable to people—"ducks lay eggs" is acceptable while "ducks are female" is not).

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




has a tail" suggest that "most" would be an appropriate understanding of the generic operator, a generic such as "mosquitoes carry malaria" makes such analyses untenable (Krifka et al. 1995) since few mosquitoes actually carry malaria[2]. While our work makes use of a generic operator, we do not make any claims about its semantics.

Probabilistic approaches handle some of the issues raised by a generic operator by considering relative probabilities (Cohen 1996, 1999, 2004). For example, "mosquitoes carry malaria" would be true because a mosquito is relatively more likely to carry malaria than a randomly chosen insect. However, relative probabilities alone are not sufficient. Consider the example "bees are sterile", which is not an acceptable generic even though a randomly chosen bee *is* more likely to be sterile than another randomly chosen insect[3]. Recently, improvements to such probabilistic analyses have been proposed which make correct truth predictions for a larger number of generics. In particular, van Rooij and Schulz (2019); Kochari, Van Rooij, and Schulz (2020) model a causal link between the concept and property in a generic in order to predict a generic's truth; Tessler and Goodman (2019) propose a Bayesian model of belief updating for interpreting generics that incorporates vagueness and context along with prevalence.

Additionally, studies from psychology have argued that generics are a default mode of cognition (Leslie 2007, 2008). In particular, studies have shown that both children and adults will often accept false quantified statements as true (i.e., "all cats have tails") if they consider the corresponding generic true, a phenomenon known as the Generic Overgeneralization Effect (Leslie, Khemlani, and Glucksberg 2011; Khemlani et al. 2007; Meyer, Gelman, and Stilwell 2011). This behavior continues, though in an attenuated form, even when people are presented with evidence that contradicts the quantified statement (e.g., with EXCEPTIONS) (Leslie, Khemlani, and Glucksberg 2011; Karczewski, Wajda, and Poniat 2020). For example, people who have recently judged that male ducks do not lay eggs will nonetheless accept "all ducks lay eggs" on almost 20 percent of trials. If they are not prompted to consider whether male ducks lay eggs, the universal claim "all ducks lay eggs" is accepted on approximately 50 percent of trials (Leslie, Khemlani, and Glucksberg 2011).

Ralethe and Buys (2022) recently investigated this tendency in LLMs, providing preliminary evidence that LLMs also exhibit a Generic Overgeneralization Effect. However, they considered the use of existential quantifiers such as "some" to be evidence of the effect, even though there is no overgeneralization involved in judging that some ducks lay eggs (instead, this just is a straightforwardly true statement). In our work, we use EXEMPLARS to document the overgeneralization effect in a large range of LLMs, and do so in a way that stays faithful to the original psychology experiments by only considering genuine cases of overgeneralization (that is, overgeneralizing from a generic to a universal).

### 2.2 Identifying Generics

Studies in NLP on generics typically focus on identifying generics within text. In particular, models are trained to predict generic expressions using discrete features (Reiter and Frank 2010; Friedrich et al. 2015; Friedrich and Pinkal 2015; Friedrich, Palmer, and Pinkal 2016; Govindarajan, Durme, and White 2019) or rule-based approaches (Suh 2006; Bhakthavatsalam, Anastasiades, and Clark 2020). Early works annotated and predicted whether an expression was a true generic at both the clause and NP level (e.g., "cats" has a non-specific referent in "cats have tails"), often within corpora for information extraction tasks, such as coreference resolution (Poesio 2004). While these works follow linguistically-based annotation guidelines to label generics, the resulting

2 Only 7-9% of the females of the species *Anopheles* (only one of 3500 mosquito species) transmit malaria (CDC 2022).
3 The majority of bees in a colony are sterile worker bees (MAAREC 2011).

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




corpora are relatively small (cf. Friedrich et al. 2015). On the other hand, more recent large-scale corpora, extracted from real-world texts, aim to capture generalizations, regardless of whether they are linguistically generics (Bhakthavatsalam, Anastasiades, and Clark 2020; Bhagavatula et al. 2022). Our work makes use of the latter type of corpora in order to generate a wide range of EXEMPLARS.

### 2.3 Generics EXEMPLARS

Although generics themselves have been studied extensively, there has been comparatively less work on EXEMPLARS. From a formal perspective, the main approach to EXEMPLARS has been to aim to formalize *how* generics tolerate EXCEPTIONS (Kadmon and Landman 1993; Greenberg 2007). Specifically, Greenberg (2007) proposed that EXCEPTIONS can be identified via a causal relationship implicit in many generics. For example, according to this approach, "birds fly" implies that some aspect of birds causes them to be able to fly (e.g., having functioning wings). When the causal relationship is blocked in individuals (e.g., birds with broken wings, birds with disproportionately small wings such as penguins) then they are EXCEPTIONS. Recent probabilistic approaches lend support to this hypothesis since they include similar notions of causality (Kochari, Van Rooij, and Schulz 2020; van Rooij and Schulz 2019). Although these mechanisms characterize EXCEPTIONS, they are primarily theoretical and not readily operationalizable. In contrast, our work aims to provide a framework that is computationally operationalized.

Recently, Allaway et al. (2023) proposed a theory-grounded computational approach to generating generics EXEMPLARS. Specifically, they used categories of generics (Leslie 2007, 2008; Khemlani, Leslie, and Glucksberg 2009) to partition generics into three sets, each with a distinct logical form. For example, categories include generics that describe principled connections (Prasada and Dillingham 2006, 2009; Haward et al. 2018) or definitions (Krifka et al. 2012). They then defined EXEMPLARS as individuals that satisfy the logical form (INSTANTIATIONS) or its negation (EXCEPTIONS) for each category. The generation was done using a constrained decoding algorithm paired with GPT-2 and information extracted from GPT-3.

Our work here instead uses ideas from pragmatics (§3.1) to both specify logical forms for generics EXEMPLARS and define prompts to generate EXEMPLARS. Although the constrained-decoding approach from Allaway et al. (2023) outperforms their baseline, there are two difficulties with their approach and we remedy these in our current work. First, assigning generics to categories requires specifying an interpretation for each generic. Doing so is not only challenging (cf., Krifka et al. 1995), it limits the scalability of the system, since all new generics must first be categorized. Therefore, our current work assumes every generic has multiple interpretations. Secondly, the logical forms proposed by Allaway et al. (2023) are too permissive in what is allowed as an EXEMPLAR. For example, in the framework of Allaway et al. (2023), the generic "birds can fly" has EXCEPTIONS that are either types of birds that cannot fly (e.g., penguins) or types of flight that birds cannot do (e.g., fly above 20,000 feet[4]). But the latter EXCEPTIONS (i.e., types of flight) are not intuitively exceptions, given our natural understanding of "birds can fly." To remedy this, we define EXEMPLARS using linguistic structures to more carefully constrain the generated output.

One usage of EXEMPLARS in NLP is countering social biases. Psychology studies have shown that generics influence and transmit social biases (Leslie 2014; Rhodes, Leslie, and Tworek 2012; Leshin, Leslie, and Rhodes 2021) and this can be particularly harmful in the case of stereotypes, especially about dangerous qualities (Leslie 2017). Drawing on these results, recent studies in NLP have investigated using EXEMPLARS to generics as a means of countering social bias implications in hate-speech (Allaway et al. 2022; Mun et al. 2023). Rather than investigate

4 Most birds fly substantially lower than 20,000 feet except during migration (Ehrlich, Dobkin, and Wheye 1988).

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




the use of EXEMPLARS, our work concentrates on developing a linguistically-founded framework to generate high-quality EXEMPLARS.

| Term | Definition | § |
|---|---|---|
| Concept $K$ | In a generic, usually a type or kind (e.g., "cats" in "cats are cute"). | 3.1 |
| Property $P$ | In a generic, usually a quality or ability (e.g., "cute" in "cats are cute"). | |
| QUD | Question under discussion – what a discourse is centered around. | 3.1.1 |
| Focus | Highlights prominent elements of a sentence. Marked with capitals in our work. | |
| $\phi_{F=y}$ | An assertion with focus on $y$ (e.g., $\phi$ = "CATS are cute" would be $\phi_{F=\text{CATS}}$). | |
| $\text{ALT}^{F=y}$ | Focus-alternatives – alternatives to the focus-marked constituent $y$ in a sentence. Includes T itself. | |
| Topic | What a sentence is about, often the syntactic subject. | |
| Tripartite structure | Partitions the semantic material of a sentence into two parts, RESTRICTOR and SCOPE, along with a quantifier. | 3.1.2 |
| RESTRICTOR | Specifies the quantifier domain in a tripartite structure. | |
| SCOPE | Specifies the properties attributed to the domain in a tripartite structure. | |
| *Gen* | Generic operator, acts as an adverb of quantification in a tripartite structure. | |
| Default Form | Default interpretation of a generic which is without focus. | 3.2.1 |
| Concept/Property-Focused Form | The interpretation of a generic where the focus is on the concept or property. | |
| $\sim T$ | Exotype – contextually-relevant alternatives to $T$ that are not $T$ itself (i.e., $\sim T = \text{ALT}^{F=T} - T$). | 3.3.1 |

Table 1: Glossary of terminology and symbols used in §3.

## 3. *GenerIX*: A Framework for EXEMPLARS

The interpretation of a generic depends on whether and how elements in it may be focused or stressed. Intuitively, if a speaker utters "birds can fly" with no particular stress or emphasis, they are making a general claim about birds, to the effect that they can fly. Compare, however, how the natural interpretation shifts if the word "birds" is uttered with heavy emphasis: "BIRDS can fly". Now the speaker may be naturally understood as making a claim about things that can fly, namely that they are birds. (For example, "BIRDS can fly" might be naturally uttered to, say, correct a child who has incorrectly asserted that squirrels can fly.)

This above example illustrates how EXEMPLARS for a generic correspondingly depend on how it is interpreted. For example, a type of bird that cannot fly (e.g., penguin) is a valid EXCEPTION to the generic "birds can fly" but not to the generic "BIRDS can fly"; for the latter, the speaker is asserting that birds *in particular* can fly, as compared to other animals, and so EXCEPTIONS will be other animals that can fly (e.g., bats, flying squirrels). Since generics can have diverse interpretations, we aim to generate EXEMPLARS for multiple interpretations[5]. Therefore, we develop our framework using ideas from pragmatics that allow us to use a cohesive formalism to both represent multiple interpretations of a generic and derive the corresponding EXEMPLARS.

Our work draws on analyses of generics that argue that the semantic material of a generic can be partitioned into two pieces and that varying this partition allows us to obtain different interpretations of the generic (Carlson 1989). These analyses are formalized using two components: tripartite structures and focus. In our work, we use tripartite structures as the mechanism

5 Determining the interpretation of a generic is a central, and unresolved, question in the literature on generics (cf. Krifka et al. 1995).

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




for representing the logical forms for generics and EXEMPLARS. Focus then maps a specific interpretation to a logical form and is also used in the definitions of EXEMPLARS.

In the following, we will first briefly review the linguistic background on focus (§3.1.1) and tripartite structures (§3.1.2). Then, we will discuss our framework *GenerIX*—**Gene**rics **R**easoning with **I**nstantiations and e**X**ceptions. Specifically, we will discuss how we construct logical forms for generics (§3.2) and how we then use these to derive precise definitions and logical forms for generics EXEMPLARS (§3.3). A summary of the definitions and terms used throughout this section is provided in Table 1.

### 3.1 Linguistic Background

Our framework uses two ideas from pragmatics (focus and tripartite structures) which we review here. Both ideas are related to the notion of *information structure* (Roberts 1996)—how information is packaged in a sentence (for a review, see Krifka 2008). **Focus** is a means of highlighting relevant meaning-bearing elements in a sentence (Kadmon 2001, originally from Jackendoff (1972)); it marks the focus of attention within discourse (Grosz 1977; Sidner 1979; Grosz, Joshi, and Weinstein 1983, 1995). For example, in English, stress is often used to mark focus. The focus not only determines the *question under discussion* (**QUD**) in a particular context, it also contributes to the partitioning of semantic material within a sentence. Such partitioning can be formally represented using a **tripartite structure**. This tripartite structure represents the sentence as specifying restricted quantification over a domain (Partee 1991). Since generics lack quantification, work with tripartite structures in the generics literature proposes a special generic operator, ***Gen***, to fill the role of quantifier and which we use in our work. We review details of these ideas below.

**3.1.1 Focus and QUDs.** Consider two speakers having a conversation. If speaker A says "cats are cute" and speaker B says "dogs are cute", there is no conflict between their statements (i.e., both assertions can easily be true). But if speaker A says "CATS are cute" and speaker B says "DOGS are cute", they are disagreeing with each other over which animals are cute. In the latter case, *focus* is used (i.e., through emphasis) to indicate an implicit contrast. We note that while this contrasting interpretation is not necessitated by focus, our discussion concentrates on it because this interpretation with generics gives rise to EXCEPTIONS.

Notice, though, that speaker B only succeeds in disagreeing with speaker A if he gives an example of something that is cute *and relevant to the context of discussion.* For example, if speaker B asserts "HEADBANDS are cute" then, unless the context is very unusual, he does not succeed in disagreeing with speaker A. Speaker A's assertion was to the effect that, of the *relevant alternatives*, cats are the ones that are cute. Here, the relevant alternatives would be naturally understood as other animals.

More formally, focus on an element of a sentence leads the sentence to be interpreted against a backdrop of alternatives (Rooth 1992)[6]. This contextually determined set contains the relevant alternatives to the focused item. The sentence with focus then asserts that it is the focused element (here "cats") that has the attributed property (being cute) as opposed to the other members of the set of alternatives (other animals). We will denote this set of **focus-alternatives** $\text{ALT}^{F=y}$. [7]

6 Although focus is often indicated through prosodic features (e.g., intonation, stress), determining focus is a complex problem and we therefore assume it is given.

7 Note that $\text{ALT}^{F=y}$ is *not* a set of propositions. The *focus set* (i.e., set of alternative propositions) for the original statement $\phi$ with focus on constituent $y$ can be obtained as $D = \{\phi(x) : x \in \text{ALT}^{y}\}$ where $\phi(x)$ is the original statement with constituent $y$ replaced with the variable $x$. For example, the set of alternative propositions for "CATS are cute" is $D = \{x \text{ are cute} : x \in \text{ALT}^{\text{CATS}}\}$.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




The focus also indicates what a discourse is about. For example, "CATS are cute" would be part of a discussion about which animals are cute. In fact, "DOGS are cute" would be a natural "counterclaim" to the assertion that cats are the central cute animal. The primary question that these assertions answer is the *question under discussion* (**QUD**) and it can be derived from the focus of an assertion.

A question can be viewed formally as denoting as set of possible answers (i.e., the domain of answers) (Hamblin 1973; Roberts 1996). This answer domain can be derived by replacing each wh-element[8] in the question with a variable $x$ and then filling in each variable with its possible values. For example, the question "who is cute?" corresponds to the answer domain containing the propositions "$x$ is cute" where $x$ is a valid possible cute thing. From this, we observe that the answer domain for a question can be constructed using the focus alternatives for an answer to the question. To see why this is the case, note that the set of possible values for $x$ is the set of focus alternatives for any of the possible answers to the question (e.g., "CATS are cute", "DOGS are cute", etc.). This is because each possible answer will have the focus on the constituent that answers the question (i.e., replaces the wh-element) and so the focused constituent is the filled-in value for $x$. The correspondence between QUD and focus means that, given the focus of an assertion, we can obtain the QUD and vice versa.

In sentences without focus (e.g., "birds can fly"), the QUD can still be determined from the sentence's *topic*[9], which indicates what the sentence is about (Vallduví and Engdahl 1996; Partee 1991). We will assume the topic is the syntactic subject when there is no focus[10], since in English this is often the case (Vallduví and Engdahl 1996; Von Fintel 1994). Intuitively, the QUD for an assertion $\phi_{T=t}$ with topic $t$ will be "what is true about $t$?". For example, the sentence "cats are cute" has the topic "cats" and therefore the QUD is "what is true about cats?".

**3.1.2 Tripartite Structures.** One way of representing the partition of semantic material as determined by focus (or topic) is a *tripartite structure*. Consider the sentence "all birds are animals" which has no focus but whose topic is "birds". This sentence can be partitioned into three parts: a quantifier ("all"), the topic ("birds"), and the rest of the sentence ("are animals"). Tripartite structures are used to formally represent this partition. In particular, a **tripartite structure** (Lewis 1975) has the form

$$\textit{Quantifier}\ x\ [\textsc{Restrictor}(x)]\ [\textsc{Scope}(x)] \tag{1}$$

where the RESTRICTOR controls the domain of the quantifier and SCOPE[11] specifies the properties attributed to the quantified members of the domain. For example the partition of "all birds are animals" would be represented as

$$\text{All}\ x\ [\textsc{Bird}(x)]\ [\textsc{Animal}(x)]\ \approx \text{"all birds are animals"}. \tag{2}$$

Here the quantifier is "All", the domain specified by the RESTRICTOR is "birds" and the SCOPE specifies that "*quantifier* birds" (i.e., "*all* birds") have the property "is an animal". A statement

8 Roberts (1996) restricts her analysis to only "who" and "what" questions.
9 As has been frequently noted, terminology surrounding the notion of "topic" is chaotic. In particular, the term has been used to mean both a sentence topic and a discourse topic. Here topic means the sentence topic as related to information structure.
10 A sentence can have both focus and topic and the resulting sentence partitionings may not align. For simplicity, we partition the sentence only using the focus, unless there is no focus (or the entire sentence is in focus), in which case we use the topic. For a more detailed discussion on the relationship between topic and focus, we refer the interested reader to Chapter 2.3.4 of Von Fintel (1994).
11 Also called "nuclear scope" and "matrix" in the literature (e.g., Von Fintel 1994).

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530



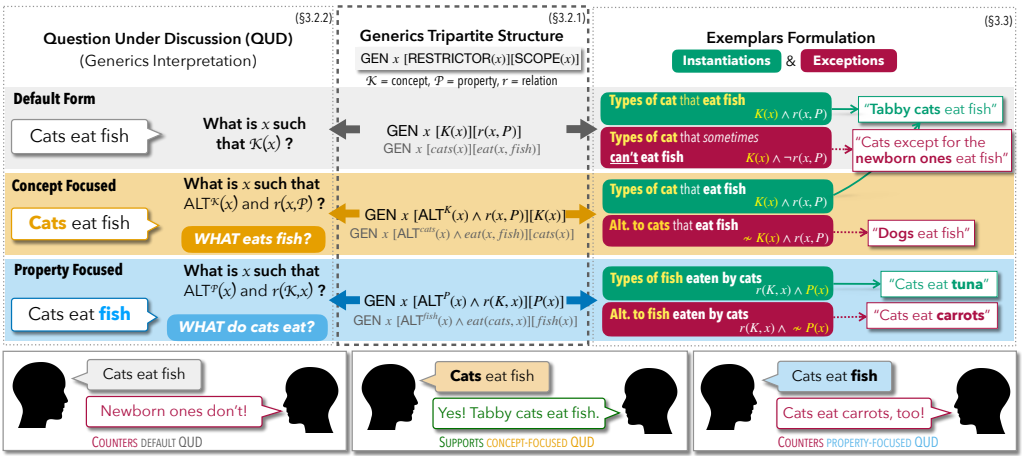




Figure 1: Overview of our *GenerIX* framework (§3).

represented with a tripartite structure is true if and only if *Quantifier* number of the members of the RESTRICTOR domain satisfy the SCOPE property (e.g., Eq. 2 is true if and only if *all* birds are animals).

Since generics do not have any explicit quantification, the **generic quantifier *Gen*** has been proposed[12], which acts as an adverb of quantification on a tripartite structure (Lewis 1975). For example, the generic "Birds can fly" can be represented as

$$\textit{Gen}\ x\ [\textsc{Bird}(x)]\ [\textsc{canFly}(x)] \qquad \text{True iff } \textit{Gen} \text{ birds can fly.} \tag{3}$$

The division of a generic into the RESTRICTOR and SCOPE is dependent on the *focus* and *topic*. In particular, we saw that for a sentence with no focus, the topic is mapped to the RESTRICTOR and the non-topic constituents to the SCOPE (e.g., as in Eq.3). If there is a focus, then the non-focused constituents map to the RESTRICTOR and the focused element to the SCOPE.

### 3.2 Logical Forms for Generics

We now specify logical forms for generics using tripartite structures (§3.1.2). As just discussed, the tripartite structure for a generic is dependent on the focus. Therefore, we specify logical forms for generics both with and without focus. We also discuss the QUD that corresponds with each logical form (§3.2.2). The QUD provides contextualization for interpreting generics and their EXEMPLARS. The logical forms are summarized in Fig. 1.

*Generics Terminology.* Throughout the following sections, we will use the following terminology for generics. Recall that a generic statement (e.g., "birds can fly") describes a *relation* between a *concept* and a *property*. Usually, a **concept *K*** is a type or kind (e.g., bird) while a **property *P*** may be an ability (e.g., fly) or quality (e.g., feathered). Note that statements *with* explicit quantification (e.g., "*Most* birds can fly") are not considered generics and are excluded from this

12 We will refer to *Gen* as a quantifier for the purposes of this paper, though there are technical reasons to think it should perhaps not be classified as one (e.g. Leslie 2007). Since these considerations do not impact anything in this paper, we keep terminology simple by referring to *Gen* as a quantifier.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




study. Additionally, we assume that the concept $K$ occurs in the subject position and that the property $P$ is part of the predicate[13].

**3.2.1 Tripartite Structures for Generics.** We will consider three interpretations of generics. First, we consider the default structure for a generic. Then we consider the interpretations of the generic with focus on the concept and property respectively[14]. Although it is also possible for the focus to be on the relation $r$, this is uncommon and therefore we will not consider these cases in our analysis.

*Default Form.* The default tripartite structure for a generic simply maps the syntactic form directly into the tripartite structure[15]. Specifically, we have

$$Gen\ x\ [K(x)]\ [r(x, P)] \qquad \text{(Default Form)}$$

where $K$ is the concept and $r(x, P)$ is true if the relation $r$ holds between individual $x$ and property $P$. Notice that Eq. 2 represents the default structure.

*Focused Form.* If a generic has a specific linguistic focus, the tripartite structure is determined by the location of the focus.

Consider the example discussed above ("CATS are cute") with the focus on the concept. Recall that the example world contains cats, dogs, and headbands but the QUD is "which animals are cute". In the corresponding tripartite structure, we know that the focused element "CATS" will be mapped to the SCOPE while the non-focused consituents are mapped to the RESTRICTOR (§3.1.2). So initially the tripartite structure would seem to be

$$\text{✗}\ Gen\ x\ [\textit{are}(x, \text{CUTE})]\ [\text{CATS}(x)]. \qquad \text{(Incorrect Concept-Focused Form Example)}$$

However, we also need to include in the RESTRICTOR the condition $\text{ALT}^{\text{CATS}}(x)$. This condition lets us capture the fact that headbands are simply not relevant to the discourse at hand. It is necessary because the relation itself (part of the non-focused constituents) does not specify that $x$ must be relevant; it is possible to say a headband is cute so therefore $x =$ headbands satisfies $\textit{are}(x, \text{CUTE})$. The relevance of $x$ is ensured by the restriction that $x$ is in the set of focus alternatives to "CATS" (i.e., "cats", "dogs"). Therefore, the correct tripartite structure is

$$\text{✓}\ Gen\ x\ [\text{ALT}^{\text{CATS}}(x) \wedge \textit{are}(x, \text{CUTE})]\ [\text{CATS}(x)]. \qquad \text{(Concept-Focused Form Example)}$$

More generally, the tripartite structure for the concept-focused form is

$$Gen\ x\ [\text{ALT}^{K}(x) \wedge r(x, P)]\ [K(x)]. \qquad \text{(Concept-Focused Form)}$$

Analogously, if the focus is instead on the **property**, the tripartite structure is

$$Gen\ x\ [\text{ALT}^{P}(x) \wedge r(K, x)]\ [P(x)]. \qquad \text{(Property-Focused Form)}$$

13 Note that we consider the verb "to be" as the relation *are*. So "are feathered" would be $\textit{are}(x, \text{FEATHERED})$

14 For simplicity, we assume the generic does not have multiple foci and that the focus does not span multiple components of the generic (e.g., the entire predicate $r + P$).

15 Since the default form does not have focus, the topic determines the tripartite structure for the generic. Since by default, we interpret the concept (e.g., "kittens") as the topic the tripartite structure for the default form is as indicated.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




For example, the generic with property-focus "Elephants eat TREES" has the form $Gen\ x\ [\text{ALT}^{\text{TREE}}(x) \wedge \textit{eats}(\text{ELEPHANT}, x)]\ [\text{TREE}(x)]$.[16]

**3.2.2 QUDs for Generics.** For a generic, the QUDs provide natural language formulations of the multiple interpretations. Specifically, recall that a generic with some interpretation is an answer to the corresponding QUD. For example, the interpretation of a concept-focused generic can be expressed as "what is $X$ such the *relation* holds between $X$ and the property $P$?" (e.g., "what is X such that X eats fish?", or more colloquially "what eats fish?", for the generic "CATS eat fish"). We discuss how to obtain the QUD for each interpretation from the corresponding generic tripartite structure since these are an important component of operationalizing the logical forms for generics.

First, for a general tripartite structure the QUD is

$$\text{What is } x \text{ such that RESTRICTOR?} \qquad \text{(QUD)}$$

where "what" can be replaced by "who" if the element in focus is human. Therefore, the QUD for a generic with concept focus is

$$\text{What is } x \text{ such that } \text{ALT}^K(x) \text{ and } r(x, P)? \qquad \text{(Concept-focused QUD)}$$

For example, if the relevant alternatives under discussions were animals, the QUD for the assertion "ELEPHANTS eat trees" would be "what animal eats trees?". Similarly, the QUD for for a property-focused generic is

$$\text{What is } x \text{ such that } \text{ALT}^P(x) \text{ and } r(K, x)? \qquad \text{(Property-focused QUD)}$$

For example, if the relevant alternatives are plants the QUD for "Elephants eat TREES" would be "what plants do elephants eat?".

Note that from the definition of QUD, for the default interpretation of a generic the QUD will be "What is true about $K(x)$?" (from Default Form). While we use the focused QUDs in defining focused INSTANTIATIONS and EXCEPTIONS, we do not use the default QUD; it is too vague.

### 3.3 Logical Forms for EXEMPLARS

We use the the tripartite structures for generics to derive precise logical forms for EXEMPLARS. As discussed, the INSTANTIATIONS are examples that demonstrate the truth of the generic (e.g., "sparrows" for the generic "birds can fly") and the EXCEPTIONS are examples where the generic does not apply. Under the tripartite structure, the INSTANTIATIONS are then instances $x$ that satisfy *both* components of the tripartite structure and EXCEPTIONS are $x$ that satisfy the RESTRICTOR but not the SCOPE.

We will first define EXEMPLARS using a general tripartite structure (§3.3.1) and lay out terminology necessary for more precise definitions (§3.3.2). Then we derive the precise logical forms for INSTANTIATIONS (§3.3.3) and for focused and default EXCEPTIONS (§3.3.4).

16 If the focus is on the relation, the tripartite structure is derived analogously as

$$Gen\ x\ [\text{ALT}^R(x) \wedge x(K, P)]\ [R(x)] \qquad \text{(Relation Focused)}$$

where $R$ is the type of the relation $r$ and $\text{ALT}^R$ is the set of alternative relations to $r$. For example, the generic with focus "Cats PLAY WITH mice" has the form $Gen\ x\ [\text{ALT}^{\text{PLAY}}(x) \wedge x(\text{CAT}, \text{MOUSE})]\ [\text{PLAY}(x)]$.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




#### 3.3.1 Definitions.

*Instantiations.* In order for **INSTANTIATIONS** to demonstrate the truth of a generic, they must be examples $x$ that make both parts of the corresponding tripartite structure true. Specifically, $x$ must satisfy both RESTRICTOR and SCOPE:

$$x : \text{RESTRICTOR}(x) \wedge \text{SCOPE}(x) \qquad \text{(INSTANTIATIONS)}$$

For example, the INSTANTIATIONS for "birds can fly", which is represented by *Gen* $x$ $[\text{BIRD}(x)]$ $[\textit{can}(x, \text{FLY})]$ (Eq. 3), are types of birds or individual birds $x$ that *can fly* (e.g., puffins, eagles, my pet parrot).

Another way to view INSTANTIATIONS is in relation to the generics QUD. In particular, since a generic answers the QUD, the INSTANTIATIONS can be considered support for the generic assertion. For example, consider the concept-focused generic "CATS are cute", which has the QUD "what animals are cute?". Then, natural INSTANTIATIONS would specific cats that are cute (e.g., kittens, Scottish Fold cats), which support the generic's answer to the QUD.

*Exceptions.* **EXCEPTIONS** are examples where the property in SCOPE for the generic does not apply. In particular, they must *satisfy the domain restriction* of the generic (i.e., RESTRICTOR) but *do not* have the relevant property (i.e., do not satisfy SCOPE). That is, EXCEPTIONS to a generic are

$$x : \text{RESTRICTOR}(x) \wedge \neg\text{SCOPE}(x) \qquad \text{(EXCEPTIONS)}$$

For example, in Eq. 3, birds that can't fly (e.g., penguins, young albatrosses) satisfy the RESTRICTOR but not the SCOPE and so are legitimate EXCEPTIONS. Notice that examples that satisfy the SCOPE but not the RESTRICTOR are not valid EXCEPTIONS.

We can also view EXCEPTIONS in relation to the generic and QUD. Specifically, the EXCEPTIONS can be considered to counter the generic assertion. For example, with a concept-focused generic "CATS are cute" (same QUD as above—"what animals are cute") the EXCEPTIONS would be other animals that are cute (e.g., puppies, hamsters); here there is a disagreement between the assertion "CATS are cute" and some alternative (e.g., "PUPPIES are cute").

**3.3.2 Terminology.** Before specifying the logical forms for EXEMPLARS, we must first lay out some terminology.

*$T(x)$ Satisfaction.* For some type $T$, we say that $T(x)$ is satisfied when $T$ is some individual, or group of individuals, of type $T$ (or one of its subtypes)[17]. For example, $\text{CAT}(x)$ is satisfied by an individual cat (e.g., my cat Mila) or a group of cats (e.g., house cats). Similarly, a relation $r(x, y)$ is satisfied by individuals $x$ and $y$ if it is true that $x$ *r*'s $y$. For example, *likes*($x$, SLEEP) is satisfied if individual $x$ likes sleep (e.g., if $x$ is a cat). The negation of a relation $\neg r$ is "not $r$". Namely, $\neg r(x, y)$ specifies that it is not true that $x$ *r*'s $y$ (e.g., $\neg$*likes*($x$, SLEEP) is true if $x$ does *not* like sleep). Similarly, $\neg T(x)$ is true if $x$ does not satisfy $T(x)$.

17 Although the literature (e.g., Krifka et al. 1995) often makes a distinction between a group of individuals and an established kind, we treat these equally in our work. For example, we treat "house cats" meaning the group of individuals of the kind *Felius catus* as the same as reference to the taxonomic kind *Felius catus*.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




*Exotype of* $T$. Additionally, we define the **exotype** of type $T$, denoted $\sim\!\boldsymbol{T}$ to be the set of contextually relevant alternatives to $T$ that *are not* $T$ *itself*. Therefore, we know that $\sim\!T(x)$ is true if $\neg T(x)$ is true *and* $x$ is in $\text{ALT}^T$. For example, in the example world discussed above (containing only cats, dogs, and headbands) the exotype $\sim\!\text{CAT} = \{\text{dogs}\}$.

### 3.3.3 Logical Forms for INSTANTIATIONS.

*Logical Form.* Recall that INSTANTIATIONS must satisfy both the RESTRICTOR and SCOPE for the generic (§3.3.1). By combining this with the tripartite structures for generics we can derive the logical forms for INSTANTIATIONS. For the default interpretation of a generic, we use the corresponding tripartite structure $Gen\ x\ [K(x)]\ [r(x,P)]$ (Default Form, §3.2.1) to derive the logical form for INSTANTIATIONS as

$$x : K(x) \wedge r(x,P) \qquad \text{(Default Form INSTANTIATIONS)}$$

For concept-focused INSTANTIATIONS we use the corresponding tripartite structure for concept-focused generics $Gen\ x\ [\text{ALT}^K(x) \wedge r(x,P)]\ [K(x)]$ (Concept-Focused Form, §3.2.1) to derive the logical form as

$$x:\ K(x) \wedge \text{ALT}^K(x) \wedge r(x,P) =\ K(x) \wedge r(x,P) \qquad \text{(Concept-Focused INSTANTIATIONS)}$$

where the simplification is due to the fact that specifying an alternative to $K$ that is also $K$ is equivalent to only specifying $K$. This means that the default INSTANTIATIONS and concept-focused INSTANTIATIONS are the same. For simplicity, we will refer to the concept- and default-focused INSTANTIATIONS as concept-focused. We can derive the property-focused INSTANTIATIONS analogously as

$$x:\ r(K,x) \wedge \text{ALT}^P(x) \wedge P(x) =\ r(K,x) \wedge P(x) \qquad \text{(Property-Focused INSTANTIATIONS)}$$

*QUD.* As noted above (§3.3.1), INSTANTIATIONS support the generic's answer to the interpretation-based QUD. The logical forms for INSTANTIATIONS supply two restrictions that make this relationship precise. For the concept-focused INSTANTIATIONS, these are: that $x$ should actually be a subtype of the concept and that the relation must hold between $x$ and the property. For example, specific types of cats that are cute, as discussed above, for the generic "CATS are cute".

Additionally, the INSTANTIATIONS can be directly related back to the QUD. This relationship is crucial for operationalizing the definitions of INSTANTIATIONS in a computational framework. Recall that the concept-focused interpretation of a generic is an answer to the corresponding QUD; namely, it answers *"What is x such that (i) $x$ is in* $\text{Alt}^K$ *and (ii) the relation $r$ holds between $x$ and $P$?"*(see Concept-focused QUD in §3.2.2). For example, a concept-focused generic such as "CATS are cute" answers the question "what is in the set of relevant alternatives to the concept (here perhaps the set of animals) and is cute?". So, the second condition, (ii), in the QUD is the same as the second condition for concept-focused INSTANTIATIONS. However, the first condition for concept-focused INSTANTIATIONS is more specific than in the QUD. Therefore, enforcing both conditions we arrive at the following:

> Concept-focused INSTANTIATIONS: Concept-focused INSTANTIATIONS are specific examples of the concept that also answer the QUD

Property-focused INSTANTIATIONS have an analogous relationship to the property-focused QUD.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




#### 3.3.4 Logical Forms for EXCEPTIONS.

*Logical Form for Focused Exceptions.* From EXCEPTIONS above (§3.3.1), EXCEPTIONS must satisfy the RESTRICTOR but not the SCOPE. So for a concept-focused generic, this means that EXCEPTIONS must satisfy $\text{ALT}^K(x) \wedge r(x, P)$ (RESTRICTOR) and not $K(x)$ (SCOPE). Specifically, concept-focused EXCEPTIONS are

$$x : \neg K(x) \wedge \text{ALT}^K(x) \wedge r(x, P) = \nsim K(x) \wedge r(x, P). \quad \text{(Concept-Focused EXCEPTIONS)}$$

where $\nsim K$ is the exotype (i.e., the set of relevant alternatives to $K$ that are not $K$ itself; §3.3.1). As with the INSTANTIATIONS, the logical form specifies two constraints: that $x$ is in the exotype of the concept and that the relation holds between $x$ and the property $P$. To continue with our example, the concept-focused EXCEPTIONS to "CATS are cute" would be contextually relevant non-cats that are cute (e.g., adorable puppies).

*QUD for Focused* Exceptions. For focused EXCEPTIONS the relationship to the QUD is again necessary for operationalizing the logical form. As with the INSTANTIATIONS above, the QUD enforces the second condition. Furthermore, the first condition for concept-focused EXCEPTIONS would be the same as that in the QUD if $x$ is allowed to be the concept itself. So

> Concept-Focused EXCEPTIONS: Concept-focused EXCEPTIONS are alternative answers to the concept-focused QUD for a generic that are *not the concept itself*.

In other words, for a concept-focused generic the EXCEPTIONS are contextually relevant *alternative concepts* (not the generic's concept) with the same property as in the generic (e.g., other furry animals that are cute). Similarly, if the generic is property-focused then the EXCEPTIONS are contextually relevant *alternative properties* (not the generic's property), for the same concept as in the generic (e.g., other characteristics of cats such as playful). The property-focused EXCEPTIONS are represented formally as

$$x : \neg P(x) \wedge \text{ALT}^P(x) \wedge r(K, x) = \nsim P(x) \wedge r(K, x). \quad \text{(Property-Focused EXCEPTIONS)}$$

where $\nsim P$ is the exotype of the property (i.e., relevant alternative properties that are not $P$).

*Logical form for Default Exceptions.* Unlike with the focused interpretation, the EXCEPTIONS to the default interpretation of a generic are *not* alternative answers to the corresponding QUD. So

> Default EXCEPTIONS: Default EXCEPTIONS are individuals where the relation *does not hold* between the concept and property.

Specifically, the logical form (combining EXCEPTIONS and Default Form) are

$$x : K(x) \wedge \neg r(x, P). \quad \text{(Default EXCEPTIONS)}$$

For example, the default EXCEPTIONS to the unfocused generic "cats are cute" would be cats that are not cute (e.g., hairless Sphynx cats, arguably)[18].

18 In contrast, alternative answers to the default QUD ("what is true about [CONCEPT]?") will be properties of the concept as a whole. Although default EXCEPTIONS *can* be written this way (e.g., "Penguin's can't fly" $\rightsquigarrow$ "Birds that are penguins can't fly"), the wording is unnatural.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| Reading | Generic | INSTANTIATION | EXCEPTION |
|---|---|---|---|
| Default | "Birds can fly" | "Owls can fly" $K(x) \wedge r(x, P)$ | "Penguins can't fly" $K(x) \wedge \neg r(x, P)$ |
| Concept-focus | "**Peru** has alpacas" | "The Andes in Peru have alpacas" $K(x) \wedge r(x, P)$ | "Chile has alpacas" $\sim K(x) \wedge r(x, P)$ |
| Property-focus | "Elephants eat **trees**" | "Elephants eat Baobab trees" $r(K, x) \wedge P(x)$ | "Elephants eat grasses" $r(K, x) \wedge \sim P(x)$ |

Table 2: Focus is indicated by **bold underline**. $K$ is the concept (blue), $P$ is the property (pink), $r$ is the relation, and $\sim$ is the *exotype* (§3.3). Note that as with Concept-Focused INSTANTIATIONS, for Property-Focused INSTANTIATIONS $\text{ALT}^P(x) \wedge P(x) = P(x)$.

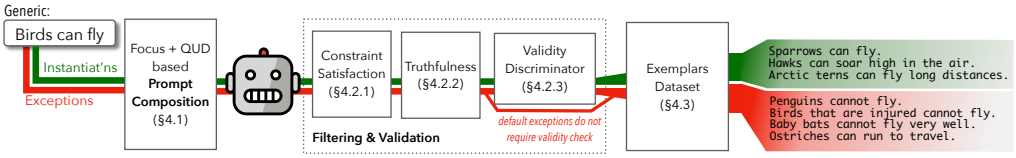



Figure 2: Overview of pipeline for our system *ExempliFI* that generates EXEMPLARS. This figure illustrates the generation of default EXCEPTIONS for the generic "Birds can fly".

## 4. *ExempliFI*: A System to Generate INSTANTIATIONS and EXCEPTIONS

For a generic, we generate EXEMPLARS using a four step system *ExempliFI*—**EXEMPL**ARS w**I**th **F**ocus **I**nterpretations. First, we use our *GenerIX* framework for EXEMPLARS (§3) to construct generation prompts for an LLM (§4.1). Specifically, we combine the QUDs for a generic (for multiple interpretations, with and without focus) with logical forms for EXEMPLARS to define prompts. After generating candidates from the LLM, we conduct two stages of filtering to ensure that the generations meet EXEMPLARS constraints and are truthful and valid (§4.2). Finally, we present our dataset for EXEMPLARS in section §4.3. The details and validation of our *ExempliFI* system are discussed in §6; implementation details are included in Appendix A.2.

### 4.1 Prompt Construction for Generation

In order to generate EXEMPLARS that have a specific pragmatic relation to a generic without requiring training, we prompt an LLM using instructions based on *GenerIX*. In particular, we use the QUDs for generics and the logical forms for EXEMPLARS to construct prompts for INSTANTIATIONS and EXCEPTIONS. In practice, determining focus (and therefore the QUD) is a complex problem and beyond the scope of this work. Therefore, we assume that each generic could be interpreted as having no focus, focus on the concept, or focus on the property. We then generate EXEMPLARS for all these interpretations

Since we consider all interpretations of a generic, we construct six prompts for generating INSTANTIATIONS and EXCEPTIONS. For the INSTANTIATIONS, we construct two prompts, one each for the concept-focused and property-focused interpretations. Note that the default INSTANTIATIONS are the same as the concept-focused INSTANTIATIONS (§3.3). Therefore, we will only generate INSTANTIATIONS for the focused readings explicitly. For the EXCEPTIONS, we construct four prompts, two prompts for the focused EXCEPTIONS and two prompts for the default EXCEPTIONS. We discuss the details of the prompt construction below and show example prompts in Table 3.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| Prompt Type | Example Prompt |
|---|---|
| Prompt-$I_K$ | List types of "birds" that can *fly*: |
| Prompt-$I_P$ | List types of "*trees*" that elephants eat: |
| Prompt-$E_K$ | List alternative answers to the question "where has *alpacas*":<br>- Peru has *alpacas*. |
| Prompt-$E_P$ | List alternative answers to the question "what elephants eat":<br>- Elephants eat *trees*. |
| Prompt-$D_1$ | Generally, birds can *fly*.<br>However, some birds cannot *fly*. These include: |
| Prompt-$D_2$ | Generally, birds can *fly*.<br>However, sometimes birds cannot *fly*. For example when they: |

Table 3: Example prompts for generating EXEMPLARS from an LLM for the generics in Table 2. The concept is marked with blue and the property with *pink italics*.

*Instantiations.* Recall that INSTANTIATIONS are (i) answers to the generic's focused QUD and are (ii) specific examples of the focused element in the generic (see Concept-Focused INSTANTIATIONS and Property-Focused INSTANTIATIONS in §3.3). We combine these two conditions into a prompt that we use for generation:

```
List types of [Focused-element] that [QUD-no-wh]:
```
(Prompt-I)

where the QUD has the wh-word removed. For example, the prompt for "CATS are cute" is "List types of cats that are cute:". We construct two prompts for INSTANTIATIONS: one for concept-focused and default readings (Prompt-$I_K$) and another for the property-focused reading (Prompt-$I_P$). See examples in Table 3.

*Focused Exceptions.* As with INSTANTIATIONS, we use the QUD for a generic to generate focused EXCEPTIONS. As discussed §3.3, focused EXCEPTIONS are alternative answers to the focused QUD that are in the exotype for the focused element (i.e., are not the generic) (see Property-Focused EXCEPTIONS and Concept-Focused EXCEPTIONS). However, the QUD alone in the prompt does not enforce the constraint that the EXCEPTIONS are in the exotype. Therefore, we supply the generic as the first answer to the QUD to indicate that the generated responses should be alternatives to the focused constituent. Specifically, given a generic, the corresponding prompt would be

```
List alternative answers to the question [QUD]:
- [GENERIC]
```
(Prompt-E)

with the model expected to produce a bulleted list of EXCEPTIONS as alternative answers. For example, concept-focused EXCEPTIONS to the generic "CATS are cute" might include "puppies are cute" or "rabbits are cute". For each generic, we have two prompts for the focused EXCEPTIONS: one for concept-focused (Prompt-$E_K$) and another for property-focused EXCEPTIONS (Prompt-$E_P$). See examples in Table 3.

*Default* Exceptions. The default EXCEPTIONS are individuals where the relation *does not hold* between the concept and property (see Eq.Default EXCEPTIONS in §3.3) Therefore, for default EXCEPTIONS we construct a prompt to generate such individuals. In particular, given a generic,

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| | EXEMPLAR Type | Source Prompt | Template |
|---|---|---|---|
| **(a)** | Concept-Focused INST. | Prompt-I$_K$ | [___]$_{\text{SUBTYPE}}$ [$r$] [$P$] |
| **(b)** | Property-Focused INST. | Prompt-I$_P$ | [$K$] [$r$] [___]$_{\text{SUBTYPE}}$ |
| **(c)** | Default EXCEP. | Prompt-D$_1$ | [___]$_{\text{SUBTYPE}}$ [$\neg r$] [$P$] |
| **(d)** | | Prompt-D$_2$ | [$K$] that [___]$_{\text{SITUATION}}$ [$\neg r$] [$P$] |

Table 4: Templates for constructing INSTANTIATIONS and default EXCEPTIONS using the generated components. $K$ (blue) is the concept, $r$ is the relation, and $P$ (pink) is the property.

the prompt would be

```
Generally, [GENERIC].                                              (Prompt-D1)
However, some [CONCEPT] [RELATION-negated] [PROPERTY]. These include:
```

where the negating the relation typically involves adding "not" with the verb.

While Prompt-D$_1$ produces specific subtypes of the concept (e.g., "penguins" and "ostriches" for the concept "birds") many default exceptions arise from temporary conditions. For example, "birds with a broken wing" cannot fly and are a valid default EXCEPTION to the generic "birds can fly", but this condition is temporary. Therefore, we use a second prompt to generate default EXCEPTIONS arising from temporary circumstances,

```
Generally, [GENERIC].                                              (Prompt-D2)
However, sometimes [CONCEPT] [RELATION-negated] [PROPERTY]. For
example when they:
```

We use both prompts to construct default EXCEPTIONS. Additionally, notice that for both prompts, we prefix the generic with the quantifying adverb "generally" (e.g., "generally, birds can fly"). This encourages the model to remember that the generic does not apply to all members of the concept. All of our prompts are constructed following our pragmatics-based *GenerIX* framework. We use our prompts for generation as described below.

### 4.2 Generation and Output Filtering

We use the prompt templates to compose prompts for EXEMPLARS candidate generation as described in §4.1. The prompts are used in a zero-shot open completion setting with the GPT-3 (Brown et al. 2020) *text-davinci-001* model (see A.2.1 for full details). We filter the output generations through two stages of filtering process.

**4.2.1 Completeness and Constraints Filtering.** We first process the generations so that all the candidates are complete sentences and fit the constraints of the corresponding EXEMPLAR's logical form. Focus EXCEPTIONS are generated as complete sentences. To ensure that they satisfy the constraints of the logical form, we remove candidates that do not meet the following requirements. For concept-focused EXCEPTIONS, the candidates must end with the generic's property; for property-focused EXCEPTIONS, candidates should begin with the generic's concept.

In contrast to the focus EXCEPTIONS generations, we observe that prompts for default EXCEPTIONS and INSTANTIATIONS produce either a list of subtypes (i.e., INST. Prompt-I's and default Prompt-D$_1$) or a list of situations (default Prompt-D$_2$). We process the generated lists into complete sentences that abide by the specified logical constraints. For each prompt and EXEMPLAR type, we craft a sentence template (see Table 4) that we deterministically fill using the generated lists. Using the information from the type of prompt and its corresponding EXEMPLAR type, we deterministically fill in sentence templates crafted for each exemplar type and source

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




prompt (see Table 4). If a candidate doesn't meet the requirements of the template, it is discarded. The details of the template filling are available in Appendix A.2.2.

**4.2.2 Filtering For Truthfulness.** Next, we use an LLM to identify and remove false candidates. 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 truth filtering step to the ranked output generations. Recently, LLMs (e.g., ChatGPT) have been successfully used to verify statements (Gilardi, Alizadeh, and Kubli 2023; Hoes, Altay, and Bermeo 2023)[19]. So, we use an LLM (GPT-3.5-Turbo) to check the veracity of generated candidate EXEMPLARS. To do this, we first convert each EXEMPLAR candidate into the singular and then ask the LLM whether the singular form is true. If the singular form of the EXEMPLAR is true, then we say the EXEMPLAR itself is true. The conversion to singular is done because the generated EXEMPLARS are often themselves generics with bare plurals. From initial explorations we found that the LLM struggles with determining the truthfulness of generic statements with bare plurals (further details in Appendix A.2.3).

To validate this filter, we evaluate GPT-3.5-Turbo on a set of 500 EXEMPLARS generated from *AnimalG* generics. The EXEMPLARS are human-annotated for truthfulness (see §6.2). The average precision for instances labeled true is 0.89 and the recall for the false instances is 0.79. This shows that most of the instances predicted as true by the LLM are in fact true (precision of true) and that most of the false instances are identified and predicted as false (recall of false).

**4.2.3 Validity Discrimination.** The filtering process provides us with a list of true EXEMPLARS. For true default EXCEPTIONS, they are fully valid because of how they are constructed. In particular, the default EXCEPTIONS are constructed by combining generated subtypes with the relation and property from the generic. Therefore, if they are true then they meet the constraints to be valid default EXCEPTIONS. This does not hold true for focused EXCEPTIONS and INSTANTIATIONS, where truthfulness does not necessarily indicate validity. For these cases we run an additional filtering step to ensure validity.

In the case of focused EXCEPTIONS, invalid statements can occur when the generated alternative concept is simply irrelevant (Ex.1) to the focused element. It can also occur in cases where the generated concept is a subset (or a superset) of the generic concept, and therefore it cannot be a valid *alternative* (Ex.2-Ex.4; see discussion §3.3.2). Additionally, true statements may possess alternatives for the wrong component of the generic (i.e., for some element other than the focused element; Ex.5). For example, given the concept-focused generic "BIRDS can fly", the following statements are all true but *invalid* concept-focused EXCEPTIONS:

Ex.1 ✗ "airplanes can fly" (irrelevant alternative),
Ex.2 ✗ "winged creatures can fly" (a superset of birds, *not* a valid alternative),
Ex.3 ✗ "non-flightless birds can fly" (a subset of birds, *not* a valid alternative),
Ex.4 ✗ "sparrows and pigeons can fly" (specific types of birds, *not* a valid alternative),
Ex.5 ✗ "birds can sing" (alternative to "fly" instead of to "birds").

Invalid cases for INSTANTIATIONS are less common, since they are constructed similarly to the default EXCEPTIONS. However, we still find that LLMs can generate invalid but true candidate INSTANTIATIONS and so benefit from an additional filtering step. For example, the statement

19 We note that some researchers have found tasks where LLMs do not perform well as evaluators (e.g., faithfulness in summarizing short stories; Subbiah et al. 2024). However, we use LLMs to validate general knowledge, which LLMs should be familiar with, and we find that the LLM filtering works well for our task (see above). See §7.1 for more discussion of the limitations of this approach.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| | | # EXCEPTIONS | | | | # INSTANTIATIONS | | | |
|---|---|---|---|---|---|---|---|---|---|
| | #Gens | Default | Concept$_F$ | Prop$_F$ | Total | Concept$_F$ | Prop$_F$ | Total | **Total** |
| *GGSmall* | 617 | 2718 | 2699 | 3522 | 8939 | 4831 | 3346 | 8177 | 17116 |
| *GGTest* | 1010 | 3392 | 6003 | 4864 | 14259 | 7999 | 4826 | 12825 | 27084 |
| *AnimalG* | 15028 | 38614 | 93587 | 75418 | 207619 | 61873 | 56143 | 118016 | 325635 |
| **All** | **16655** | 44724 | 102289 | 83804 | **230817** | 74703 | 64315 | **139018** | **369835** |

Table 5: Statistics of the dataset of EXEMPLARS generated by our *ExempliFI* system.

"binoculars are used to see things" is true but not a valid property-focused INSTANTIATION for the generic "binoculars are used to VIEW LOCATIONS" because it just paraphrases the generic.

To select the valid focus EXCEPTIONS, we train a discriminator to predict whether a statement is a valid focus EXCEPTION for a particular generic. We train a separate discriminator to predict whether a statement is a valid INSTANTIATION. Discriminator details are provided in Appendix A.2.4.

For the final output, focus EXCEPTIONS and INSTANTIATIONS are ranked using the relevant trained discriminator. For the default EXCEPTIONS, we follow Allaway et al. (2023) and rank generations by using the average of two ranks: perplexity and NLI. The NLI rank is determined by the probability that the default EXCEPTION candidate contradicts the generic. We keep all valid system generations in order to construct a large-scale dataset.

### 4.3 Generated EXEMPLARS Dataset

Using *ExempliFI*, we generate EXEMPLARS for the generics in the three datasets *GGSmall*, *GGTest*, and *AnimalG*, which we detail below.

- ***GGSmall*** is a set of **617 generics** sourced from Gen-Atomic (Bhagavatula et al. 2022). Gen-Atomic is a dataset of 14M generated generics. It encompasses a wide range of diverse everyday generalizations (e.g., "Bicycles have two wheels", "Hammers are used for construction"). For *GGSmall*, we use the same subset of the *human-verified* Gen-Atomic as used by Allaway et al. (2023). This subset excludes generics with human referents as the concept (e.g., nationalities, professions) due to social bias concerns.
- ***GGTest*** is sourced from Gen-Atomic's test set and consists of **1010 generics**. We obtain this subset by excluding generics with a human referent (as with *GGSmall*) and filtering using the discriminator published with *GGTest*. Specifically, we use the discriminator to select only statements that both humans and the model agree are generics.
- ***AnimalG*** is a set of 15028 **generics** with animal referents. These generics are sourced from the dataset of generics constructed by Ralethe and Buys (2022) for probing generic processing in LLMs. The original dataset was extracted from GenericsKB using a fixed list of animals (e.g., "reptiles", "fish", "birds").

In total we generate 369, 835 EXEMPLARS across 16, 655 generics (Table 5). In particular, we generate 230, 817 EXCEPTIONS and 139, 018 INSTANTIATIONS. Examples of the generated data are shown in Table 6. We validate in §6 that *ExempliFI* generates high-quality EXEMPLARS. Such a large and high quality dataset allows us to thoroughly probe the capabilities of LLMs to reason about generics (§5).

In the analysis of our dataset, we observe that among the different types of EXEMPLARS, *ExempliFI* is strongest at generating default EXCEPTIONS that go beyond knowledge-based counter-evidence. For example, "family chapels are not open to public" is an acceptable knowledge-based EXCEPTION to generics like "chapels are open to public" since it relies on general static

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530



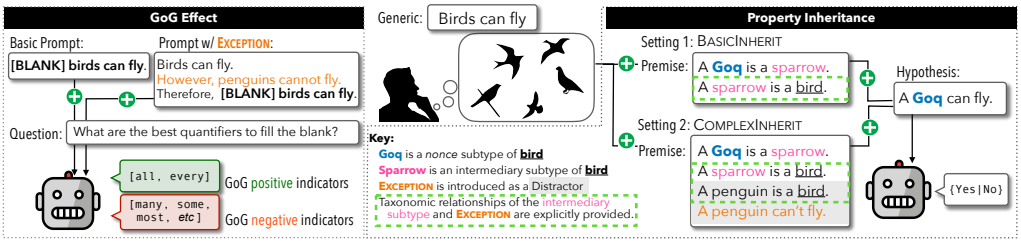




Figure 3: Using dataset generated by *ExempliFI*, we test LM's capabilities to reason with generics.

properties about a known kind (i.e., properties of family chapels). But *ExempliFI* is also able to generate reasoning-based EXCEPTIONS that require access to non-static and counterfactual knowledge about the generic. For example, for the same generic ("chapels are open to public"), *ExempliFI* generates the default EXCEPTION "chapels that are closed for renovation are not open to the public during regular hours"; this requires counterfactual reasoning about what might cause a chapel to be temporarily closed.

These reasoning-based default EXCEPTIONS are often more compelling than knowledge-based ones. This is because they do not simply enumerate factoids that may or may not be relevant to a user. Instead, the reasoning-based EXCEPTIONS provide additional relevant information that allows humans to contextualize and understand the generic. For example, "cats that live in apartments or homes do not sleep in trees" is a more useful EXCEPTION to the generic "cats sleep in trees" than "cheetahs do not sleep in trees" is; the average person may not know whether or not cheetahs generally sleep in trees[20].

Additionally, the reasoning-based EXCEPTIONS allow *ExempliFI* to produce default EXCEPTIONS for generics where the concept does not have any well-known subtypes (i.e., it is not an established kind). For example, "scavenger hunt" does not have well-known subtypes but *ExempliFI* can still generate default EXCEPTIONS using reasoning; it generates "a scavenger hunt that is too difficult is not a fun way to spend an afternoon with friends" as an EXCEPTION to "a scavenger hunt is a fun way to spend an afternoon with friends". We see this increased coverage reflected in how often the reasoning-based EXCEPTIONS are ranked highly. In particular, for the *AnimalG* data $\sim$63% of the top ten default EXCEPTIONS are reasoning based. For *GGSmall* and *GGTest* the proportions of reasoning-based EXCEPTIONS in the top ten default EXCEPTIONS are $\sim$73% and $\sim$17% respectively. The disparity between *GGTest* and the other two data sources may be due to the fact that *GGTest* contains a large number of definitional or tautological generics (e.g., "an outlet mall is a place") for which it is difficult to come up with reasoning-based default EXCEPTIONS. For example, while default EXCEPTIONS to "hot chocolates taste like cocoa" are difficult to construct, since all hot chocolate will taste like cocoa, valid EXCEPTIONS still exist (e.g., the property-focused EXCEPTION "hot chocolates taste like vanilla"). In the following section, we will use the EXEMPLARS generated by *ExempliFI*, especially the default EXCEPTIONS, to investigate how LLMs reason about generics.

## 5. Reasoning in Generics in LLMs

Theorists (Leslie 2008; Leslie, Khemlani, and Glucksberg 2011; Leslie and Gelman 2012; Sutherland et al. 2015; Gelman, Tapia, and Leslie 2016) have proposed that generics represent

20 Unlike many other large cats, cheetahs tend to sleep under, not in, trees. See https://cheetah.org/learn/about-cheetahs.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| | | |
|---|---|---|
| | ***AnimalG Data*** | |
| **(a)** | ***Generic:*** *Cats sleep in trees* | |
| | **Default EXCEPTIONS:** | **INSTANTIATIONS:** |
| | • Cats that live in apartments or homes do not sleep in trees.<br>• Cheetahs do not sleep in trees. | • Pumas sleep in trees.<br>• A lynx sleeps in trees.<br>• Cats sleep in the branches of a tree. |
| **(b)** | ***Generic:*** *Birds fly.* | |
| | **Default EXCEPTIONS:** | **INSTANTIATIONS:** |
| | • Ostriches are not able to fly.<br>• Birds that have a broken wing are not able to fly.<br>• Birds that have their wings clipped are not able to fly. | • A peregrine falcon is able to fly.<br>• A hummingbird is able to fly.<br>• Birds are able to glide for long periods of time.<br>• Birds are able to hover in the air. |
| **(c)** | ***Generic:*** *Moose have winter coats.* | |
| | **Concept-Focused EXCEPTIONS:** | **Property-Focused EXCEPTIONS:** |
| | • Rabbits have winter coats.<br>• Bears have winter coats. | • Moose have hooves.<br>• Moose have big antlers. |
| **(d)** | ***Generic:*** *Deer live in meadows.* | |
| | **Concept-Focused EXCEPTIONS:** | **Property-Focused EXCEPTIONS:** |
| | • Antelopes live in meadows.<br>• Rabbits live in meadows. | • Deer live in the forest.<br>• Deer live in the mountains. |
| | ***GGTest Data*** | |
| **(e)** | ***Generic:*** *a scavenger hunt is a fun way to spend an afternoon with friends.* | |
| | **Default EXCEPTIONS:** | **INSTANTIATIONS:** |
| | • A scavenger hunt that is too difficult is not a fun way to spend an afternoon with friends.<br>• A scavenger hunt that is in an unsafe location is not a fun way to spend an afternoon with friends. | • A food scavenger hunt is a fun way to spend an afternoon with friends.<br>• A historical scavenger hunt is a fun way to spend an afternoon with friends. |
| **(f)** | ***Generic:*** *Binoculars are used to view location.* | |
| | **Concept-Focused EXCEPTIONS:** | **Property-Focused EXCEPTIONS:** |
| | • A telescope is used to view location.<br>• A satellite is used to view location. | • Binoculars are used to view stars.<br>• Binoculars are used to magnify objects. |
| | ***GGSmall Data*** | |
| **(g)** | ***Generic:*** *A rose is placed in a container with water.* | |
| | **Default EXCEPTIONS:** | **INSTANTIATIONS:** |
| | • Dried roses are not placed in a container with water.<br>• Roses that are used as part of a garland are not placed in a container with water. | • A rose is placed in a vase.<br>• A rose is placed in a bowl.<br>• A cut roses are placed in a container with water. |
| **(h)** | ***Generic:*** *Cakes are made with a mix.* | |
| | **Concept-Focused EXCEPTIONS:** | **Property-Focused EXCEPTIONS:** |
| | • Pancakes are made with a mix.<br>• Waffles are made with a mix.<br>• Brownies are made with a mix. | • Cakes are made with eggs.<br>• Cakes are made with a cake pan.<br>• Cakes are made with an oven. |

Table 6: Examples of EXEMPLARS generated by *ExempliFI*.

a default way of thinking for humans. That is, generics are more cognitively fundamental than quantified statements. This **generics-as-default** hypothesis has been supported by cognitive science studies with both children and adults (cf. §2.1). Our generated EXEMPLARS allow us

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| | ***Top Quantifiers* Probe** | ***Psychology-based Questions* Probe** |
|---|---|---|
| Base | [blank] {birds can fly.}$_G$<br>What are the 5 best quantifiers to fill in the blank? | {All}$_Q$ {birds can fly.}$_G$ Yes or No? |
| + EXCEP. | {Birds can fly.}$_G$<br>However, {penguins cannot fly.}$_E$<br>Therefore, [blank] {birds can fly.}$_G$<br>What are the 5 best quantifiers to fill in the blank? | {Penguins cannot fly.}$_E$ Yes or No? |

Table 7: Prompts used to probe LMs for the GOG effect. The braces ({}) indicate slots filled by the generic (teal $G$), an EXCEPTION (orange $E$), or one of three quantifiers (pink $Q$). The three possible quantifiers are: "all", "some" or nothing (to obtain a generic statement).

to investigate whether LLMs exhibit similar behaviors by probing how they treat the semantic relationship between generics and EXEMPLARS.

Since LLMs have been trained on large quantities of human-written text, we hypothesize they will seem to treat generics as defaults, similarly to humans. Specifically, we investigate two behaviors: *overgeneralizing from generic statements to universals* (i.e., accepting false universal statements as true when the corresponding generic is true) and *treating generic statements as universals* in property inheritance.

First, humans have a cross-culturally documented tendency to treat universally quantified statements as generic (e.g., "all birds can fly" $\rightsquigarrow$ "birds can fly") (Hollander, Gelman, and Star 2002; Khemlani et al. 2007; Mannheim et al. 2010; Meyer, Gelman, and Stilwell 2011; Tardif et al. 2012). That is, universally quantified statements are deemed true despite the presence of known EXCEPTIONS. This has been termed the **G**eneric **O**ver**G**eneralization **(GOG)** effect and supports the generics-as-default hypothesis. In particular, if understanding generics is more basic (i.e., default behavior) than understanding quantified statements, humans should (and do) sometimes fall back on their interpretation of a generic when confronted with a universal (Leslie, Khemlani, and Glucksberg 2011).

Second, when drawing inferences (e.g., in syllogistic reasoning), generics are often treated as universally quantified (Khemlani, Leslie, and Glucksberg 2008, 2009). For example, the generic "birds can fly" is often treated as the default rule "in general, if X is a bird then X can fly" and so it can be inferred that a new bird, Y, can fly without knowing anything about Y. Such plausible inferences have been documented in human interactions (Collins and Michalski 1989) and support the hypothesis that generics are a default way of generalizing information.

In order to analyze whether LLMs show evidence of generics-as-defaults behavior, we first probe the GOG effect in LLMs (§5.1) and then investigate property inheritance via generics (§5.2).

### 5.1 Generics and Quantification

In humans, the GOG effect has primarily been demonstrated by asking study participants to agree or disagree with statements that are either quantified or generic (e.g. Leslie, Khemlani, and Glucksberg 2011). Specifically, in the first paper to investigate the effect, Leslie, Khemlani, and Glucksberg (2011) asked human participants to respond with either yes or no, indicating their agreement with various statements. Those statements were presented universally quantified (e.g., "all ducks lay eggs"), in generic form (e.g., "ducks lay eggs") or existentially quantified ("some ducks lay eggs"). Multiple participants responded to each statement. Of particular interest were generics that are intuitively true, despite only $\sim$50% of the concept satisfying the property. For example, "ducks lay eggs" is intuitively true, even though only mature, fertile, female ducks lay

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




egg. For such generics, accepting a universally quantified version (e.g., "all ducks lay eggs") clearly constitutes an error that an educated adult human is in a position to spot and avoid. Contrary to this, Leslie, Khemlani, and Glucksberg (2011) found evidence of the GOG effect across multiple studies. That is, they found that educated adult humans have a tendency to treat universals as though they are generics and will accept such universals a robust percentage of the time, despite being aware of the exceptions (e.g., male ducks).

To examine whether LLMs overgeneralize and exhibit a GOG effect, we probe which quantifiers are generated by LLMs for generic statements. We assess the quantifiers generated by the LLM and, if either of the universal quantifiers "all" or "every" is generated, we count this as evidence of the GOG effect. Note that there are many choices of quantifiers that do not constitute the GOG effect (e.g., "many", "most", "some", "few"), as well as adjectival modifiers, and so LLMs have many options other than universal quantifiers. By supplying a universal quantifier to modify the generic, the LLM provides evidence of a tendency to conflate universally quantified statements and generics (i.e., evidence of a GOG effect).

Furthermore, we investigate whether a GOG effect in LLMs persists when counter evidence to the generic is presented. To do this, we augment our probe to include EXCEPTIONS. This aims to control for the possibility that any observed GOG effect is due to missing knowledge about relevant exceptions in the LLM. Specifically, we follow Karczewski, Wajda, and Poniat (2020) and include automatically generated default EXCEPTIONS in the generic's context. In humans, knowledge of EXCEPTIONS may reduce the GOG effect (Leslie, Khemlani, and Glucksberg 2011) but it does not eliminate it; we expect LLMs to exhibit similar behavior.

#### 5.1.1 Methods.

*Probes.* Our main probe (**Top Quantifiers**) asks LLMs to answer a fill-in-the-blank question about how a generic statement should be quantified (see "Top Quantifiers Probe" column in Table 7). The LLM is asked to respond with the top five options (quantifiers) to fill in the blank. We do this to obtain variation in the models' responses. While human studies on the GOG effect can collect responses from multiple participants for each generic (described above; cf. Leslie, Khemlani, and Glucksberg 2011), multiple generations from LLMs exhibit very little variation. Therefore, a ranked list of quantifiers approximates multiple participant responses within the framework of a single LLM. To measure the GOG effect from this probe, we compute the frequency of the universal quantifiers "all" and "every" among the elicited quantifiers for each generic. For each generic, we run this probe with and without EXCEPTIONS to investigate the impact of EXCEPTIONS on the models' behavior.

Additionally, we run a supplementary probe (***Psychology-based Questions***) that directly follows psychology studies (e.g., Leslie, Khemlani, and Glucksberg 2011) on the GOG effect in humans. Specifically, this probe consists of four questions asked to the LLM separately (see "Psychology-based Questions Probe" column in Table 7). The model is asked to answer yes or no to each. The first two questions each contain a quantified version of the generic. The quantifier is either the universal "all" or the existential "some". The third question contains the unquantified generic (i.e., the generic as a generic). The fourth question asks whether the model endorses an EXCEPTION. We then measure how often LLMs endorse both the universally quantified generic (e.g., "all birds can fly") and an EXCEPTION to the generic (e.g., "penguins cannot fly").

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




*Details.* We run our probes on generics from the *AnimalG* dataset[21] for which our system generates valid default EXCEPTIONS. For our main *Top Quantifiers* probe we use all 10488 generics with valid default EXCEPTIONS. For the supplementary *Psychology-based Questions* probe we use a random sample of 1000 of the 10488 generics with valid default EXCEPTIONS; this sample is chosen such that each generic has at least three valid default EXCEPTIONS. For both probes we use the top three valid default EXCEPTIONS for each generic in the probe; we average the results across the EXCEPTIONS for each generic.

We investigate four LLMs[22]:

- **GPT-3**: A decoder-only transformer-based language model with 175B parameters and trained for causal language modeling (Brown et al. 2020).
- **GPT-3.5-Turbo** and **GPT-4**: Transformer-based language models that substantially improve performance over GPT-3 (OpenAI 2023). Both models are trained with reinforcement learning from human feedback (RLHF) (Christiano et al. 2017) and are optimized for chat purposes[23]. In RLHF, a reward function is learned from human preferences about generated text. The resulting reward function is then used with reinforcement learning to fine-tune the LLM.
- **LLAMA-2**: An open-source transformer-based language model trained using RLHF (Touvron et al. 2023). We use the version with 7B parameters optimized for dialogue use cases.

We choose these models to include both top-performing (GPT-4) and open-source (LLAMA-2) models, along with the LLMs used in our *ExempliFI* system (GPT-3 and GPT-3.5-Turbo).[24]

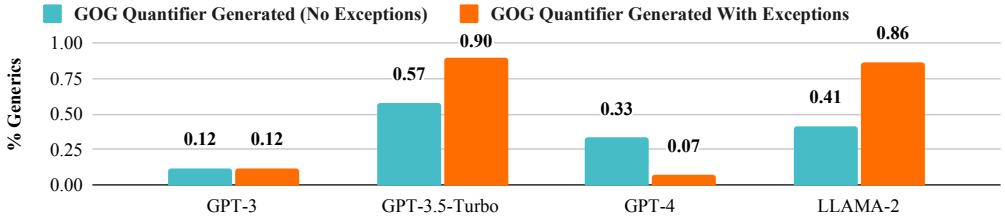
**5.1.2 Analysis.** To examine the main *Top Quantifiers* probe, we use as a metric the percentage of generics where a universal quantifier *is generated* by the LLM, both with and without default EXCEPTIONS probe. For the *Psychology-based Questions* probe, we examine four slices of the models' responses. These are the percentage of generics where: (i) the LLM endorses both the universally quantified generic and an EXCEPTION (i.e., a GOG response), (ii) the LLM *does not endorse* the universally quantified generic but does endorse an EXCEPTION (i.e., responds "correctly" and in a way that indicates knowledge of the generic's EXCEPTIONS), (iii) the LLM *does* endorse the universally quantified generic but does not endorse the EXCEPTIONS (i.e., the response may be attributable to ignorance about the generic' EXCEPTIONS), and (iv) *neither* the universally quantified generic nor the EXCEPTIONS are endorsed by the LLM (i.e., the model may be generally ignorant about the generic or it may be unable to respond to the prompt for some other reason). Note that we include full results for the other questions in *Psychology-based Questions* probe in Appendix B.1.2.

Our results from both probes show a non-zero GOG effect across all LLMs (Fig. 4) with the level ranging from fairly negligible (present on <10% of generics) to substantial (present on over 90% of the generics). With the *Top Quantifiers* probe, overgeneralization actually increases for half the LLMs (GPT-3.5-Turbo and LLAMA-2) when EXCEPTIONS are added to the prompt (Fig. 4a); for GPT-4 the effect does substantially decrease in the presence of EXCEPTIONS. The results of the *Psychology-based Questions* probe also show a GOG effect for all LLMs (Fig. 4b). For the

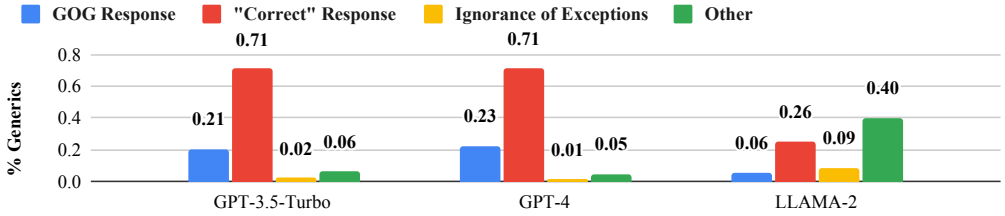
21 We use *AnimalG* data because it is the largest set of generics and because it consists of simple sentences that most closely follow linguistic definitions of generics.
22 GPT-3—`text-davinci-003`; GPT-3.5-Turbo—`gpt-3.5-turbo`; GPT-4—`gpt-4-0613`; LLAMA-2—`meta-llama/Llama-2-7b-chat-hf`.
23 https://platform.openai.com/docs/models
24 We do not use GPT-3 for the *Psychology-based Questions* probe experiments because GPT-3 has been deprecated by OpenAI.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




(a) Results of the *Top Quantifier Probe*. Percentage of generics where a GOG quantifier (i.e., one of the universals "all" or "every") is generated by the LLM to quantify the generic. *With Exceptions* indicates the prompts include default EXCEPTIONS. Higher values indicate a larger amount of overgeneralization by the LLM.




(b) Results of the *Psychology-based Questions Probe*. Percentage of generics with a particular response to the universally quantified generic and to the default EXCEPTIONS. *GOG Response* — both the universal statement and an EXCEPTION were endorsed, *"Correct" Response* – the universal statement was not endorsed but an EXCEPTION was endorsed, *Ignorance of* EXCEPTIONS– the universal statement was endorsed but the EXCEPTIONS were not endorsed, *Other* – neither the universal statement nor the EXCEPTIONS were endorsed.

Figure 4: Results of the GOG effect probes across LLMs.

GPT models, only ∼7.5 of the universal statement endorsements (∼2 of the total generics) can potentially be attributed to ignorance of the generics' EXCEPTIONS; the remaining portion indicate overgeneralization. Note that although the GOG response from LLAMA-2 is minimal, this may be attributable to the model failing to adequately process the probes' prompts. Specifically, a large portion (40%) of LLAMA-2's responses fall into the "Other" category (i.e., neither the universally quantified generic nor the EXCEPTIONS were endorsed by the model) and LLAMA-2 exhibits a substantially lower rate of correct responses than the GPT models (26% compared to 71% for the GPTs)[25]. Overall, regardless of whether we probe by asking the model a generation (i.e., free response – *Top Quantifiers*) or classification (i.e., yes or no – *Psychology-based Questions*) question, **a GOG effect is present across LLMs and the effect does not disappear in the presence of EXCEPTIONS.**

When comparing the two probes, recall that the way in which EXCEPTIONS are presented to the LLMs is distinct. Specifically, in the *Top Quantifiers* probe, the default EXCEPTION is presented together with the generic itself *in one prompt* (see Table 7). Hence the LLMs' response is generated conditioned on the EXCEPTION. In contrast, for the *Psychology-based Questions* probe, the universally quantified version of the generic and the default EXCEPTION are presented in *separate prompts*. This means that in the *Psychology-based Questions* probe the EXCEPTION does not explicitly impact the response to the universal statement, and vice versa. This probe,

25 An additional 19% of LLAMA-2's responses are not interpretable as a yes or no response (e.g., a string of only newline characters).

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




which does not explicitly require the model to reason about the relationship between the generic and EXCEPTION, serves as a baseline indicator of GOG effect in LLMs. **The observed increases in the GOG effect with the *Top Quantifiers* probe may then be due in part to difficulties in reasoning for the LLMs.**

To better understand the universal quantifiers elicited by the *Top Quantifiers*, we examine the LLMs' consistency across varying EXCEPTIONS. That is, for generics where an LLM produces a universal quantifier in the presence of at least one of the corresponding EXCEPTIONS, we compute the proportion where the LLM produces universals for *all* EXCEPTIONS (i.e., where the LLM is consistent in its decision to universally quantify the generic). We observe that the **models for which the GOG effect increases with EXCEPTIONS (i.e., GPT-3.5-Turbo and LLAMA-2) have substantially more consistency** than the models where the GOG effect decreases. That is, GPT-3.5-Turbo/LLAMA-2 are consistent for $60.3\%$ of cases on average while GPT-3/GPT-4 are consistent for only $24.4\%$ of cases on average. The low consistency with GPT-3/GPT-4 is likely indicative of a further decrease in GOG effect: the LLMs not only produce universals for less generics, they are also less confident even when they do.

Since the *Top Quantifiers* probe is essentially a free response question for the LLMs, the generated quantifiers may not actually be grammatical quantifiers. For example, the LLMs may generate adjectival modifiers (e.g., "old", "female"). When we compare the modifiers produced by the GPT models, we observe that both GPT-3 and GPT-4 produce nearly twice as many unique modifiers ($3633$ and $4077$ respectively) as GPT-3.5-Turbo and LLAMA-2 ($1351$ and $1589$ respectively). That is, GPT-3/GPT-4 are better able to come up with complex modifiers that qualify the generic without using a grammatical quantifier. In fact, GPT-4 produces $1505$ multi-word modifiers (compared to only $617$ from GPT-3.5-Turbo and $629$ for LLAMA-2[26]). For example, GPT-4 produces $146$ combinations of "only"+adjective (e.g., only matured, only female, only domestic, only fertile, etc.), compared to less than $50$ from GPT-3.5-Turbo/LLAMA-2. Such varied and appropriate modifiers allow the model to qualify the generic assertion while simultaneously accounting for the exceptions. Therefore, **the observed differences in GOG effect across LLMs may be partly attributable to differences in the LLMs' ability to produce appropriate, non-quantifier modifiers.**

Overall, these results show that LLMs do exhibit a GOG effect. In particular, like, humans, models conflate generics with universally quantified statements in multiple probe settings. However, unlike in humans, this effect is sometimes increased in the presence of default EXCEPTIONS. This could be due in part to difficulties for LLMs in reasoning over EXCEPTIONS and generics together or to varying abilities of LLMs to generate appropriate, non-quantifier modifiers for generics. We leave further discussion for §5.3 and now investigate whether the conflation of generics and universals extends to property inheritance in LLMs.

### 5.2 Generics and Inheritance

Human reasoning about property inheritance often confounds generics with universally quantified statements. For example, based on the generic "birds can fly" humans tend to infer that if Polly is a bird then Polly can fly, unless they are provided with evidence to the contrary. In other words, humans treat the generic "birds can fly" as universally quantified ("all birds can fly") and therefore applicable to all individual birds. However, when presented with counterexamples (e.g., "Bob is a penguin and Bob cannot fly"), humans sometimes deviate from applying the default rule (i.e., not

26 We remove complete sentences that do not contain the concept from consideration as modifiers. For most models the number of such instances is very small (16-22), although LLAMA-2 produces $475$ instances.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| Subsest | 2-Step? | Distractor? | Premises |
|---|---|---|---|
| *BasicInherit* | | | {Sheep have horns.}$_G$ A Yeb is a sheep. |
| | ✓ | | {Sheep have horns.}$_G$ A Yeb is {a bighorn sheep}$_S$. {A bighorn sheep}$_S$ is a sheep. |
| *ComplexInherit* | | ✓ | {Sheep have horns.}$_G$ A Yeb is a sheep. Sheep that have their horns removed for safety reasons are sheep. {Sheep that have their horns removed for safety reasons do not have horns. }$_E$ |
| | ✓ | ✓ | {Sheep have horns.}$_G$ A Yeb is {a bighorn sheep}$_S$ {A bighorn sheep}$_S$ is a sheep. Sheep that have their horns removed for safety reasons are sheep. {Sheep that have their horns removed for safety reasons do not have horns. }$_E$ |
| **Conclusions** | | | |
| *Condition*$^{[+]}$: A Yeb has horns. | | | |
| *Condition*$^{[-]}$: A Yeb does not have horns. | | | |

Table 8: Prompts for both property inheritance probe sets. Premises are shown in the top portion and the conclusions for both conditions are provided in the bottom portion of the table. *2-Step* indicates the prompt question involves 2-step inheritance, *Distractor* indicates the prompt question contains an EXCEPTION. The braces ({}) indicate slots filled by the generic (teal $G$), an EXCEPTION distractor (orange $E$), or an INSTANTIATION's subtype (magenta $S$).

inferring that Polly can fly). In this work, we probe how LLMs reason about property inheritance with generics and how EXCEPTIONS impact this reasoning.

Property inheritance with generics has primarily been studied in formal methods for nonmonotonic reasoning[27]. Early work in artificial intelligence investigated inheritance from generalizations with exceptions (Hanks and McDermott 1986; Brewka 1987; Horty and Thomason 1988) and a number of formal logics have been proposed to facilitate reasoning with such sentences (McCarthy 1980, 1986; Reiter 1978, 1980; Poole 1988; Delgrande 1988; Veltman 1996; Collins and Michalski 1989). In an attempt to benchmark the success of nonmonotonic reasoning systems, Lifschitz (1989) compiled a set of challenge problems. We use the inheritance reasoning subset of these problems as inspiration for constructing probe questions for LLMs[28].

To investigate inheritance reasoning in LLMs, we construct a set of probe questions that ask the model whether an assertion about property inheritance is valid. For example, an LLM is asked whether the assertion "A Goq can fly" is valid given the premises "birds can fly" (a generic) and "a Goq is a bird" (a statement that connects the queried subtype, Goq, to the concept in the generic). By using generics rather than explicitly quantified statements (e.g., "all birds can fly"), we probe whether LLMs treat consider generics as default inference rules. Additionally, we probe how LLMs reason about inheritance in the presence of EXCEPTIONS to the generic.

### 5.2.1 Methods.

27 In deductive logic, it would be invalid to conclude that the hypothesis below follows from the premises:

Premises: Polly is a bird. Birds can fly
Hypothesis: Polly can fly.

However, humans and machines tend to employ more flexible reasoning and formal methods for nonmonotonic reasoning attempt to capture some of this flexibility (cf. Ginsberg 1987). Following nonmonotonic logics, it is valid to infer the hypothesis from the premises.

28 Specifically, problems B1 and B2.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




*Setup.* We construct two probe sets of inheritance questions using generics and the EXEMPLARS generated by *ExempliFI*. Each question is centered around a property generalization (e.g., "sheep have horns"). The model is then asked to determine the validity of a conclusion about property inheritance: whether the specified property is inherited by a subtype (e.g., "a Yeb"). We use nonsense words for the subtype in the conclusion in order to ensure that the model does not rely on prior knowledge in determining validity. We construct questions with both positive (e.g., "A Yeb has horns") (***Condition***$^{[+]}$) and negative (e.g., "A Yeb does not have horns") (***Condition***$^{[\neg]}$) conclusions. In contrast to prior work, which compares the log-likelihood of sentence pairs to measure LLM's ability to do property inheritance reasoning (Misra, Ettinger, and Rayz 2021), we use direct questions for evaluation. This formulation is better suited to decoder-only LLMs (e.g., GPT series) which are trained to follow task instructions.

Our first probe set, ***BasicInherit***, contains questions with two components. The first is a generic (e.g., "sheep have horns"), which provides a generalization that a concept (e.g., "sheep") has a property (e.g., "have horns"); the second component is a set of premises consisting of the relevant taxonomic relations between the subtype and the concept (e.g., "A Yeb is a sheep"). In contrast to prior work (Talmor et al. 2020; Misra, Ettinger, and Rayz 2021), we explicitly include both the property generalization (i.e., the generic) and taxonomic relations. First, by including the property generalization in the question, we ensure that decisions by the LLM cannot be attributed to a lack of knowledge (e.g., not knowing that sheep have horns). Secondly, the taxonomic relations enable us to use nonsense words as the subtype in the conclusion. Furthermore, the taxonomic relations also mean that the model is actually being evaluated on whether it endorses a property inheritance assertion, and not on whether it can also connect the nonsense word to the concept.

Our second probe set, ***ComplexInherit***, probes property inheritance behavior in the presence of potentially conflicting information. Namely, each question includes a distractor—an example of the concept that does *not* have the property (e.g., "sheep that have had their horns removed for safety reasons"). These questions additionally include the same components as the questions in *BasicInherit*. The distractor is included as part of the premises in the question, along with the taxonomic relations. As distractors, we use generated default EXCEPTIONS to the generic in each question.

For both probe sets, we construct both single and 2-step inheritance questions. The single-step inheritance questions probe inheritance to a direct subtype of the concept. For example, given "sheep have horns" and "a Yeb is a sheep", the model is asked to determine whether "a Yeb has horns" is valid (in *Condition*$^{[+]}$). In the 2-step inheritance questions, an intermediate subtype is introduced between the conclusion and the concept. For example, for the *Condition*$^{[+]}$ conclusion "a Yeb has horns", the premises "sheep have horns", "bighorn sheep are sheep", and "a Yeb is a bighorn sheep" illustrate 2-step inheritance (i.e., sheep $\rightarrow$ bighorn sheep $\rightarrow$ Yeb). We use generated INSTANTIATIONS to a question's generic as the intermediate subtypes.

*Details.* We use a subset of 1000 randomly selected generics from the *AnimalG* dataset[29] to construct our evaluation questions. Each question has the following format:

```
Premises:
[premises]
Conclusion: Therefore, [conclusion].
Does the conclusion logically follow from the premises?
(yes/no)
```

29 We use *AnimalG* because it was created to contain only generics about animals. So it is well suited for investigating property inheritance to subtypes. In contrast, the generics in *GGSmall* and *GGTest* describe concepts that do not necessarily have subtypes and would not make sense in an inheritance setting.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| | *Condition*$^{[+]}$ | | *Condition*$^{[\neg]}$ | | All | |
|---|---|---|---|---|---|---|
| | Inherits | Doesn't | Inherits | Doesn't | Inherits | Doesn't |
| GPT-3.5 | 0.885 | 0.115 | 0.889 | 0.107 | 0.887 | 0.113 |
| GPT-3 | 0.944 | 0.057 | 0.969 | 0.032 | 0.956 | 0.044 |
| GPT-4 | 0.985 | 0.016 | 0.999 | 0.001 | 0.992 | 0.008 |

(a) Results on the *BasicInherit* probe questions.

| | *Condition*$^{[+]}$ | | *Condition*$^{[\neg]}$ | | All | |
|---|---|---|---|---|---|---|
| | Inherits | Doesn't | Inherits | Doesn't | Inherits | Doesn't |
| GPT-3.5 | 0.116 | 0.884 | 0.417 | 0.582 | 0.267 | 0.733 |
| GPT-3 | 0.164 | 0.837 | 0.554 | 0.447 | 0.359 | 0.642 |
| GPT-4 | 0.727 | 0.274 | 0.964 | 0.037 | 0.845 | 0.155 |

(b) Results on the *ComplexInherit* probe questions.

Table 9: Percentage of property inheritance probe questions where the model did or did not endorse the property inheritance conclusion. In *Condition*$^{[+]}$, "yes" indicates endorsement while in *Condition*$^{[\neg]}$ "no" indicates endorsement.

The premises consist of the property generalization (the generic), relevant taxonomic relations, and potentially a distractor (an EXCEPTION) as discussed above (see examples in Table 8). We use the wording "logically follow" to instruct the model to do deductive inference[30]. As a result, if the model endorses property inheritance (e.g., responding "yes" to the *Condition*$^{[+]}$ conclusion in Table 8) we know that it has treated the generic in the premises (e.g., "sheep have horns") as if it were universally quantified (e.g., "all sheep have horns"), since that is the only way for such a conclusion to be deductively valid.

Given a generic, we construct premises for both *BasicInherit* and *ComplexInherit*. In particular, we first construct single and 2-step inheritance premises for *BasicInherit*, using the top ranked concept-focused INSTANTIATION as the intermediate subtype in 2-step inheritance. Then, we add a distractor and connecting taxonomic information to each set of premises in *BasicInherit*, making the premises for *ComplexInherit*. We use the top ranked default EXCEPTION as the distractor. This results in four sets of premises. Note that the nonsense subtypes are randomly chosen from a set of five nonsense words and the same nonsense word is used across all premises for a generic.

The probe questions for a generic are constructed by combining each set of premises with the conclusions for both conditions. Namely, each set of premises produces two questions, one for *Condition*$^{[+]}$ and one for *Condition*$^{[\neg]}$. Therefore, for each generic we have eight questions and so our final probe sets consist of 4000 questions each.

We probe property inheritance in three LLMs[31]: GPT-3, GPT-3.5-Turbo, and GPT-4 (see §5.1.1 for descriptions of the models).

**5.2.2 Results and Analysis.** As a metric, we measure the percentage of questions on which each LLM endorses property inheritance (Table 9).

30 We validate that the models behave correctly in response to this wording with a manually constructed set of sample questions. See Appendix B.2 for details.

31 We do not include LLAMA-2 in this analysis because the model generates almost exclusively a single response ("no" for 91.5% of the instances) and so its behavior corresponds to randomly assigning yes or no.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




First, we observe that with *BasicInherit*, all three LLMs endorse property inheritance with very high frequency (94.8% on average, see Table 9a). Furthermore, we observe minimal variation between *Condition*$^{[+]}$ and *Condition*$^{[\neg]}$. **This indicates that the models *do* tend to treat generics as default inference rules** for straightforward property inheritance, and that negation does not impact the models' behavior. Since endorsing property inheritance here means judging that the argument is valid, we take this as further evidence that LLMs, like humans, treat generics as akin to universally quantified statements in a reasoning context.

In contrast, on the *ComplexInherit* probe questions, endorsement of the property inheritance is substantially lower. In other words, **the presence of distractors (EXCEPTIONS) has a substantial impact on how often models support property inheritance**. For example, an LLM would likely assert that some bird X cannot fly because an example is provided of a bird that cannot fly (e.g., penguins). Unlike with *BasicInherit*, there is a clear difference between the two conditions. In particular, property endorsement is more frequent with *ComplexInherit* in *Condition*$^{[\neg]}$ across all models. By responding "no" to the conclusion in *Condition*$^{[\neg]}$ (e.g., "a Yeb does not have horns"), one interpretation is that the model is implicitly endorsing the opposite (i.e., that a Yeb does have horns). This means the models are more likely to implicitly endorse property inheritance, rather than explicitly. An alternative interpretation is that the models do a better job in this condition of recognizing that, in fact, neither conclusion follows deductively from the premises.

Looking more closely at *ComplexInherit*, we observe that GPT-4 endorses property inheritance substantially more than GPT-3.5-Turbo and GPT-3; in *Condition*$^{[+]}$ the latter two models support property inheritance in only $\sim$300 instances. This difference may be due in part to substantial increases in GPT-4's reasoning performance (e.g., as demonstrated by large improvements on academic exams; OpenAI 2023), which may allow the LLM to better reason about the semantic relationship between generics and their EXCEPTIONS.

If we set aside questions of deductive validity and focus on non-monotonic patterns of inference, in human discourse, **considering EXCEPTIONS relevant to possible property inheritance is not unreasonable**. Following Gricean maxims (Grice 1975), if a speaker asserts something, then other participants may assume that that thing is relevant. So in this case, the model might reasonably assume that the default EXCEPTION is relevant to whether the property is inherited to the subtype; the assertion of an EXCEPTION may imply the final subtype is also an EXCEPTION. Although this contrasts with patterns of formal nonmonotonic reasoning (Lifschitz 1989), it is supported by studies with humans. In particular, human studies on default reasoning have found evidence that factors such as the choice of EXCEPTION and the number of EXCEPTIONS influence whether humans deem that a property is inherited from a generic to a subtype (Elio and Pelletier 1996; Pelletier and Elio 2005). Given that LLMs are trained from vast quantities of human language, we hypothesize that as with quantification (cf. §5.1.2) certain LLMs (e.g., GPT-3.5 Turbo) may have adopted some of the pyschologism that Pelletier and Elio (2005) argue impacts human inheritance reasoning. Our large generated dataset of generics EXCEPTIONS facilitates further investigations into this question.

Overall, our results show the LLMs do treat generics as default inference rules in simple property inheritance scenarios. LLMs also seem to exhibit similar inference patterns as humans in complex inheritance reasoning. In particular, when EXCEPTIONS are present, LLMs may consider them relevant to potential inheritance.

### 5.3 Discussion

Based on our results, we offer the following insights into how LLMs reason about generics.

***Do LLMs show evidence of overgeneralization with generics?*** Our analyses using generics and EXEMPLARS show that LLMs *do* exhibit evidence of the GOG effect. First, we find that across

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




LLMs, universal quantifiers are generated to modify generic statements, even when EXCEPTIONS to the generic are present. This indicates that quantified statements and generics are conflated by LLMs (§5.1.2). Additionally, LLMs treat generic statements as default rules about property inheritance (§5.2.2). That is, LLMs treat generics as universally quantified by reasoning that, as a matter of deductive logic, properties are inherited from them.

***What differences do we observe between how LLMs and humans reason about generics?*** Unlike humans, some LLMs demonstrate a greater degree of conflation between generics and universally quantified statements when EXCEPTIONS are presented (§5.1.2). That is, some LLMs (GPT-3.5-Turbo and LLAMA-2) produce *more* universal quantifiers for generics that are presented along with their EXCEPTIONS.

On the one hand, the GOG effect (see above) is evidence that LLMs do *not* process the universal quantifier "all" in a strictly logical sense, and therefore are not expected to always adjust the quantifiers in their response (e.g., no longer generate "all") when EXCEPTIONS are presented. On the other hand, the *increase* in GOG effect with some models suggests that factors other than quantifier treatment (i.e., non-logical handling of "all") are impacting the models' responses to EXCEPTIONS. One such factor may be that EXCEPTIONS require more complex reasoning from the LLMs. Specifically, EXCEPTIONS require LLMs to relate multiple sentences (the generic and the EXCEPTION) and further require that the internal semantic representations of concepts account for EXCEPTIONS. Determining whether this is indeed the case in LLMs is beyond the scope of this work. Therefore, further investigations are needed to better understand the source of LLMs' behavior in response to generic EXCEPTIONS.

Additionally, LLM behavior may differ from humans' reasoning due to non-human-like errors. For one, LLMs are known to be sensitive to the contents of their prompts and so differences in length between prompts with and without EXCEPTIONS may impact how the LLMs respond. Additionally, humans typically understand the type of response required of them when answering a prompt. In contrast, LLM responses do not always adhere to basic syntactic or commonsense constraints. For example, LLMs may generate an entire sentence with a new concept as a modifier for a generic (see discussion §5.1). Finally, synthetic data may result in a small number of malformed inputs (e.g., with incorrectly conjugated verbs). While humans would likely be able to understand what the input should have been and then respond accordingly, LLM behavior in response to such inputs is unpredictable.

***What are our insights into how LLMs reason about generics?*** Our investigations with generics and EXEMPLARS show that LLMs exhibit patterns of non-logical (i.e., non-deductive) reasoning with similarities to how humans actually reason. Specifically, LLMs show evidence of conflating generics and universally quantified statements. This behavior aligns with the human tendency to treat generic statements as a default mechanism for generalization, a behavior that is argued to be cognitively fundamental (Leslie 2007). Although LLMs are not humans, they *are* trained on massive corpora of human language. Therefore, it remains to be seen how GOG-effect-influenced behavior benefits (or harms) LLM performance in various downstream tasks (e.g., question-answering). The implications for bias and stereotyping are particularly important. That is, the GOG effect may be indicative of a bias towards associating a property with all members of a particular group, which could potentially be harmful.

It should be noted that LLM behavior does not entirely align with humans. Specifically, LLMs do not consistently adjust their responses when presented with contradictory evidence (i.e., obvious counterexamples) and further studies are needed to identify the source of this behavior. For example, we observe discrepancies between models trained with (e.g., GPT-4) and without (GPT-3) RLHF but further investigations are needed to determine the specific impact of RLHF. Such investigations are important because EXCEPTIONS are crucial for many applications that

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




reason with generics (e.g., robot object retrieval using generic rules about where objects are normally located; Sridharan et al. 2015). Studies from philosophy and psychology on how humans actually use and acquire generics, and their EXCEPTIONS, may help in identifying ways to improve LLM reasoning about EXCEPTIONS. Finally, different approaches to prompting (e.g., chain-of-thought; Wei et al. 2022) may help clarify LLMs responses when reasoning about generics and EXEMPLARS.

## 6. Validation and Details of the *ExempliFI* System

In this section we discuss details of our *ExempliFI* system and validation experiments: the data sources (§6.1), annotation procedures (§6.2), and baselines used for comparison (§6.3). We also present the results of a human evaluation of the data generated by *ExempliFI* (§6.4). We show that *ExempliFI* generates high-quality EXEMPLARS.

### 6.1 Data

We source generics from three datasets in our experiments, as described in §4.3. Additionally, we use the EXEMPLARS annotated by Allaway et al. (2023), *GGSmall-Exemplars*, as additional training data for our discriminators (§4.2.3). Full details of the datasets used and their processing can be found in Appendix A.1.

### 6.2 Annotations

We collect annotations for: (i) training and evaluating the quality of our validity discriminators used in §4.2.3, (ii) evaluating the truthfulness filter in §4.2, and (iii) conducting human evaluation. All annotations are done using Amazon Mechanical Turk with three annotators per HIT (paid at $15/hour on average). For each annotation tasks, annotators must first pass a corresponding qualification task consisting of five questions. We report the full agreement measures across all tasks in Appendix A.3.

**6.2.1 Validity Annotations.** We use three separate annotation tasks to annotate the validity of candidate EXEMPLARS, one each for the INSTANTIATIONS, the default, and the focused EXCEPTIONS. All three tasks are framed as a debate between two students, where one students (Student A) asserts the generic and the other (Student B) replies with an EXEMPLAR. We use this framing since, as discussed in §3.3, EXEMPLARS provide alternative answers to the same QUD that the generic answers. Concretely, in the INSTANTIATION task, the annotators are asked to assess whether the responses are valid corroborating evidence for the corresponding generic. For EXCEPTIONS task, the annotators are asked to assess whether the responses are countering evidence for the corresponding generic, taking into account the focus. Full details, instructions, and examples are provided in Appendix A.3.1.

We use this annotation process for gathering data for discriminator training and evaluation (§4.2.3), and for evaluating *ExempliFI* generations (§6.4). For discriminator use, we have annotators label 4100 randomly selected INSTANTIATIONS and focused EXCEPTIONS across 613 generics selected from both the *GGTest* and *AnimalG* datasets. The average Fleiss' $\kappa$ is 0.2903.

For conducting a human evaluation of system generations (§6.4), we collect annotations for EXEMPLARS from 96 generics from *GGSmall* for which our system and both baseline systems (i.e., the system proposed by Allaway et al. (2023) and the corresponding GPT-3 baseline; see §6.3 for more details) each produce five EXCEPTIONS and five INSTANTIATIONS. That is, we have

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




annotators label a total of 1440 EXCEPTIONS and 1440 INSTANTIATIONS. The average Fleiss' $\kappa$ is 0.3056.

We note that while all the validity annotation tasks achieve moderate inter-annotator agreement, these tasks are difficult for annotators. Firstly, annotators must determine the validity of an EXEMPLAR in relation to a specific QUD. This requires detailed and complex instructions (see Appendix A.3). Additionally, determining EXEMPLARS validity requires nuanced judgements about object category boundaries. In particular, for the focused EXCEPTIONS annotators are asked two questions: whether a candidate answers a specific QUD, and whether it includes a relevant alternative (i.e., whether the concept or property is in the exotype; §3.3.2). The second question (determining relevance of alternatives) is substantially more difficult than the first question with an average Fleiss' $\kappa$ of 0.259, compared to 0.405 for the first question. As an example, consider whether "a phone" is a valid alternative to the concept "cameras" for the generic "cameras are used to take pictures"; since many phones *have* cameras the two categories overlap, which makes the validity of the alternative ambiguous. Our annotation procedures are a substantial improvement compared to prior work (see Appendix B.3 for analysis), but the continuing challenges of annotating EXEMPLARS highlight the importance of future work in this area.

**6.2.2 Truthfulness Annotations.** To evaluate the truthfulness filter (§4.2), we collect annotations on the truthfulness of generated EXEMPLARS. We use one annotation task, in which annotators are provided with four sentences (EXEMPLARS) and asked to judge whether each is either "generally true" or "generally false". Annotators are asked to mark nonsensical statements as false. Full instructions and examples are provided in Appendix A.3.2.

We have annotators label a set of 500 EXEMPLARS for generics from the *AnimalG* dataset: 100 randomly selected from each type (concept- and property-focused EXCEPTIONS, default EXCEPTIONS, concept- and property-focused INSTANTIATIONS). The Fleiss' $\kappa$ is 0.4407.

### 6.3 Generation Baselines

As generation baselines, we use the constrained decoding system (*ConstraintDec*) and a prompt-based GPT-3 baseline from Allaway et al. (2023). Both systems assign generics to categories based on their semantic behavior (e.g., "birds can fly" is categorized as *principled* because there is a strong association between "birds" and "fly") and use templates to control the output form and content. In order to evaluate the baseline generations using our proposed annotation setup (§6.2.1), we deterministically map (see Table 10) baseline templates to our five types of EXEMPLARS (default EXCEPTIONS, concept- and property-focused EXCEPTIONS, and concept- and property-focused INSTANTIATIONS). Note that the baselines systems do not generate concept-focused EXCEPTIONS. Full details of the systems are given in Appendix A.4.

### 6.4 Human Evaluation

To quantitatively evaluate *ExempliFI*, we conduct a human evaluation by collecting validity annotations on a subset of generated EXEMPLARS (§6.2) and computing precision at $k$ (for $k = 1$ and $k = 5$).

*ExempliFI* substantially outperforms both baselines by a large gap (average of 20.63 points) for EXCEPTIONS (Table 11). For the INSTANTIATIONS, *ExempliFI* performs the same as *ConstraintDec* while outperforming the GPT-3 baseline by an average of 12.93 points. Comparing the three systems, we observe that the INSTANTIATIONS are less difficult to generate than the EXCEPTIONS; the baseline performance is 17.08 points higher on average for the INSTANTIATIONS

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| EXEMPLAR Type | Input Template | Output Template | In Ours? | Mapped Type | |
|---|---|---|---|---|---|
| INST. | $[K_{\text{SUBTYPE}} + r]$ | [___]$_P$ | ✓ | Concept$_F$ | **(i)** |
| | $[K + r]$ | [___]$_{\text{SUBTYPE}_P}$ | ✓ | Property$_F$ | **(ii)** |
| | $[K_{\text{SUBTYPE}} + r]$ | [___]$_{\text{SUBTYPE}_P}$ | ✗ | Property$_F$ | **(iii)** |
| EXCEP. | $[K_{\text{SUBTYPE}} + \neg r]$ | [___]$_P$ | ✓ | Default | **(iv)** |
| | $[K + \neg r]$ | [___]$_{\text{SUBTYPE}_P}$ | ✗ | Default | **(v)** |
| | $[K + r]$ | [___]$_{\sim P}$ | ✓ | Property$_F$ | **(vi)** |
| | $[K_{\text{SUBTYPE}} + r]$ | [___]$_{\sim P}$ | ✗ | Property$_F$ | **(vii)** |

Table 10: Templates used for generation by the baseline systems. The *Input Template* is used to construct the prompt and the *Output Template* is used to control the output. $K$ is the concept, $P$ is the property, $r$ is the relation, and $\sim$ indicates the exotype. ✓indicates *ExempliFI* uses an analogous template for generation while ✗indicates it does not. *Mapped Type* is the mapping used for evaluating the baseline generations under our annotation setup.

| | EXCEPTIONS | | INSTANTIATIONS | |
|---|---|---|---|---|
| | P@1 | P@5 | P@1 | P@5 |
| *GPT-3-baseline* | 0.6250 | 0.6437 | 0.7812 | 0.7437 |
| *ConstraintDec* | 0.6667 | 0.7062 | **0.9271** | **0.8729** |
| *ExempliFI* | **0.8750** | **0.8583** | 0.9167 | 0.8667 |

Table 11: Precision at $k$ ($P@k$) results from human evaluation.

| | EXCEPTIONS | | | INSTANTIATIONS | |
|---|---|---|---|---|---|
| | Concept$_F$ | Property$_F$ | Default | Concept$_F$ | Property$_F$ |
| *GPT-3-baseline* | $0.6389_{288}$ | $0.6584_{442}$ | $0.4915_{59}$ | $0.7678_{211}$ | $0.7249_{269}$ |
| *ConstraintDec* | – | $0.7325_{456}$ | $0.2083_{24}$ | $\mathbf{0.8922}_{232}$ | $\mathbf{0.8548}_{248}$ |
| *ExempliFI* | $0.8701_{254}$ | $\mathbf{0.8190}_{326}$ | $\mathbf{0.9416}_{154}$ | $0.8845_{251}$ | $0.8472_{249}$ |

Table 12: Precision of generated EXEMPLARS by type.

than the EXCEPTIONS. Therefore, the large improvement on EXCEPTIONS highlights the quality and usefulness of our system.

We further examine precision by EXEMPLAR type. Note that *ConstraintDec* does not generate concept-focused EXCEPTIONS (see Table 10 for a summary of the baseline generation patterns). For all types of EXCEPTIONS, *ExempliFI* outperforms the baselines (Table 12). This improvement is particularly large for the default EXCEPTIONS. Not only does *ExempliFI* increase the precision of default EXCEPTIONS by an average of 59.15 points, it also generates around 4.5 times as many valid default EXCEPTIONS. Since these EXCEPTIONS address the default interpretation of the generic they are particularly important to generate.

The ability of *ExempliFI* to produce reasoning-based default EXCEPTIONS (§4.3) also means that it produces default EXCEPTIONS for a wider range of generics than the baselines. For example, consider the output generations from our system and from *ConstraintDec*:

**Generic:** Chapels are open to the public during regular hours.
***ConstraintDec* candidates:**
• {wedding chapels, a wedding chapel, funeral chapels, chapels} are open to members of the clergy.
***ExempliFI* candidates:**

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




• Chapels are open to the public by appointment.
• Chapels that are closed for renovation are not open to the public during regular hours.
• Chapels that are closed for repairs are not open to the public during regular hours.
• Chapels that are being used for a funeral are not open to the public during regular hours.
• Chapels are open to the public by special arrangement.

While all of the top generations from *ConstraintDec* are property-focused EXCEPTIONS, *ExempliFI* produces three valid default EXCEPTIONS in the top five generations.

When examining the property-focused EXCEPTIONS, we observe that *ExempliFI* is able to produce better alternatives than the baselines. Consider the following example

**Generic:** a coyote should be considered a wild animal.
***ConstraintDec* candidates:**
• a coyote pup should be kept in a cage.
• a coyote pup should be kept in a home.
***ExempliFI* candidates:**
• a coyote is considered a member of the dog family.
• a coyote is considered a danger to people and pets.

Here, the alternatives generated by *ConstraintDec* ("kept in a cage" and "kept in a home") are not relevant to the generic. They are physical rather than abstract considerations about coyotes. In contrast, *ExempliFI* generates alternative thoughts to have about coyotes, better addressing the original generic. This is likely due to the way the focused EXCEPTIONS are generated in *ExempliFI*. Specifically, the LLM is prompted to not only address the QUD arising from the generic (here, "what coyotes should be considered") but also to provide an alternative to the property given in the generic. The alternative is encouraged by providing the actual generic as the first answer to the QUD. The example illustrates the strength of this approach.

Overall, our results show that *ExempliFI* generates high-quality EXEMPLARS. When evaluated by humans, *ExempliFI* substantially outperforms baselines from prior work. The improvement is particularly large for default EXCEPTIONS. Not only are default EXCEPTIONS the most natural EXCEPTIONS to a generic, they are also challenging for LLMs to reason about.

## 7. Conclusion

In this work, we investigate how LLMs process and reason about generics by automatically generating EXEMPLARS and using them to probe specific capabilities in LLMs. To generate EXEMPLARS, we propose a computational framework *GenerIX* that uses the pragmatic phenomenon of focus to capture a range of interpretations for a generic across contexts of use. Our *GenerIX* framework provides precise logical-form based definitions for EXEMPLARS that we operationalize in a generation system *ExempliFI*.

We use *ExempliFI* to automatically generate a dataset of $\sim 370k$ EXEMPLARS across $\sim 17k$ generics. Human validation of our dataset shows that, in comparison to prior work, *ExempliFI* generates substantially higher quality EXCEPTIONS (with an average improvement of 20.6 precision points). While the INSTANTIATIONS generated by *ExempliFI* are comparable to prior work, it should be noted that generating EXCEPTIONS is more difficult than generating INSTANTIATIONS. Despite this, *ExempliFI* improves the generation of EXCEPTIONS such that they are of comparable quality to the INSTANTIATIONS. Additionally, *ExempliFI* generates more diverse EXCEPTIONS than in prior work by including not only knowledge-based examples but

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




also examples based on reasoning about temporary situations. Our large, high-quality dataset of generated EXEMPLARS allows us to effectively probe how LLMs reason about generics.

We use our validated dataset to probe how LLMs process and reason about generics. Specifically, we investigate the GOG effect (i.e., Generic Overgeneralization; cf. §5) in relation to LLMs. In humans, the GOG effect supports the *generics-as-default* hypothesis: that generics are a default way of thinking (e.g., Leslie, Khemlani, and Glucksberg 2011; Khemlani, Leslie, and Glucksberg 2008). By probing LLMs for the GOG effect, we examine how LLM reasoning is similar to human reasoning when processing a simple but fundamental type of statement (generics). We find that LLMs *do* show evidence of a GOG effect when reasoning about both quantifiers and property inheritance. This behavior is similar to how humans exhibit the GOG effect. For example, LLMs have similar patterns of non-logical reasoning as humans when considering property inheritance. However, we also find that LLMs struggle to reason about the relationship between generics and EXEMPLARS. This indicates the challenges and importance of further studies into reasoning about generics and EXEMPLARS.

### 7.1 Limitations

*GenerIX Framework.* Our *GenerIX* framework makes simplifying assumptions about generics. First, our framework operates only with generics in the active voice. Secondly, we assume that all three interpretations (default, concept-focused, and property-focused) have valid EXEMPLARS for each generic. However, for certain generics this may not hold but our system will still attempt to generate all types of EXEMPLARS. For example, the generic "squares have four sides" has no default EXCEPTIONS since there are no squares without four sides. Third, *GenerIX* relies on a restricted set of potential foci within a generic. In particular, the focus is either on the entire subject of the sentence (i.e., the concept) or the entire predicate without the verb (i.e., the property). Finally, we assume that the focus is given. Future work should investigate how to determine the focus from the generic's context.

*Data.* In this work, we source generics from three existing datasets, all exclusively in English. Therefore, our approach may not be suited to all generics in all languages. For example, our system requires that generics do not have a complex syntactic structure (e.g., as in nested generics). Additionally, the generic statements we use are not guaranteed to be linguistically generics (e.g., "young gazelle break vertebrae" is in *AnimalG* but is questionably a generic).

As in Allaway et al. (2023), we do not generate EXEMPLARS for generics involving human referents (e.g., professions, nationalities). While our *ExempliFI* system could be used with generics involving human referents, additional care should be taken in such cases to check that stereotypes or social biased statements are not deemed valid output. Checking for this is beyond the scope of this work but is an important future step.

*ExempliFI System.* Our *ExempliFI* system uses an LLM to evaluate the truthfulness of the system generated candidates (§4.2). Although LLMs have been successfully used to verify statements in recent works (e.g., Gilardi, Alizadeh, and Kubli 2023; Hoes, Altay, and Bermeo 2023), they have also been shown to hallucinate (Rohrbach et al. 2018). Therefore, using an LLM to check for veracity may result in errors. Additionally, the LLM may only be able to determine the veracity of statements that appeared in (or are similar to) their training data. As a result, the generated data that passes the truthfulness filter may be limited in coverage, with true but unseen statements being deemed false. We discuss the implications of potential overlap between LLMs' training data and generated data below.

We also note that both our *ExempliFI* system itself and our system validation experiments rely on human annotations of EXEMPLARS validity (§6.2). While all the validity annotation tasks

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




achieve moderate inter-annotator agreement, these tasks are difficult for annotators and we discuss reasons for the difficulty in §6.2. The difficulty of the annotation tasks, and resulting moderate inter-annotator agreement, means that noise may be introduced. Specifically, the precision of both our system and the baselines (§6.4) may be inflated by noise in the annotations. For example, if annotators are biased towards marking default EXCEPTIONS as valid, this would inflate the precision of *ExempliFI* for generating EXCEPTIONS, since our system generates nearly three times as many default EXCEPTIONS as the baselines. Work using our system to generate additional data could consider applying additional filters (e.g., NLI as in Allaway et al. 2023) to ensure data quality. Additionally, future work should investigate improvements to the annotation procedures for validating EXEMPLARS.

*Experiments.* In our probing experiments into how LLMs reason about generics (§5), we do not do extensive prompt tuning. In particular, we use only two prompts for our main probes on the GOG effect (§5.1) and a single prompt for our probes on property inheritance (§5.2). However, recent works have shown that LLMs can be sensitive to features of the prompts, including formatting (Sclar et al. 2023), how the task is presented (Hu and Levy 2023), and the provided context for the model (Kassner and Schütze 2020; Lampinen 2023; Misra, Ettinger, and Mahowald 2024). Changes to the prompts used for probing may impact the scale of the results we observe. However, we note that the goal of our probing experiments is to determine whether *any* GOG effect is ever exhibited by LLMs. While measuring the scale of the GOG effect (as opposed to whether or not it is present) is interesting, constructing such prompts would require careful interdisciplinary crafting of prompts (e.g., incorporating both cognitive psychology and linguistics) and is a compelling direction for future investigations.

Additionally, our probing experiments all use only zero-shot evaluation. We choose to do this so as not to bias the models' outputs. In particular, we do not want to inadvertently prime the models, through few-shot examples, to generate specific quantifiers (e.g., "all") or to exhibit certain inheritance reasoning behavior. However, we acknowledge that the LLMs may behave differently under few-shot evaluation, compared to zero-shot evaluation. We think that few-shot overgeneralization analysis is a promising future direction. Furthermore, differences in how models behave in the two settings may help clarify what kinds of generalization information models learn from text.

As mentioned above, we use automatically generated EXEMPLARS for our probing experiments in §5. This data is generated using generics sourced in part from ConceptNet (Speer, Chin, and Havasi 2017) and GenericsKB (Bhakthavatsalam, Anastasiades, and Clark 2020). Since the LLMs used in our experiments were trained on data scraped from the internet through at least 2021[32], it is likely that many of our generics appeared in some way in the LLMs' pre-training data. Consequently, our probing experiments use examples that the model has seen in some capacity. This could result in noise in the results we observe. In particular, we may observe an artificially lower GOG effect with probes that contain only a generic (i.e., only examples the model has already seen), compared to the probes containing EXCEPTIONS. Furthermore, the model may be biased towards a particular response (e.g., only generating "most" as a quantifier for a particular generic) depending on what data was present in pre-training. Further experiments should be done to understand what data the LLMs have seen, especially which EXEMPLARS, and how this impacts their responses.

Finally, our probe experiments only examine generics in isolation. In particular, they do not evaluate how LLMs reason about generics within real texts. Although this aligns closely with human studies on the GOG effect (e.g., Leslie, Khemlani, and Glucksberg 2011), the

32 See https://platform.openai.com/docs/models

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




observable impacts of how LLMs treat generics will likely be in downstream tasks (i.e., texts with contextualized generics). Therefore, future work should explore additional methods to probe LLMs' reasoning with generics within texts.

## A. Experimental Details

### A.1 Data

*GGSmall and GGTest.* We use two sets of generics sourced from Gen-Atomic (Bhagavatula et al. 2022). The set *GGSmall* contains 617 generics that are human-verified. These generics are the generics used by Allaway et al. (2023) for which their system generates both valid INSTANTIATIONS and EXCEPTIONS.

The set *GGTest* contains 1010 generics sourced from Gen-Atomic's test set. The full Gen-Atomic test set consists of 2254 generics deemed true by humans. From these, we remove temporal generics (i.e., beginning with "before", "after", "while"), generics relating to necessity (i.e., beginning with 'in order to"), and generics with verbs of consideration (i.e., consider, posit, suppose, suspect, think). We then run the discriminator published with Gen-Atomic and keep only generics that the discriminator predicts as true with confidence at least 0.7. Finally, we removed generics with human referents (e.g., professions, nationalities) using a manually compiled list.

We preprocess both *GGSmall* and *GGTest* by removing adverbs of quantification (i.e., usually, typically, generally). We also convert hedging statements to more explicit forms (e.g., "may have to be" to "must be").

*AnimalG Data.* These generics are sourced from GenericsKB by Ralethe and Buys (2022) for a fixed list of animals. The generics are separated into two categories: majority characteristic generics (i.e., true about the majority of the kind) and minority characteristic generics (i.e., true for only a minority of the kind). We combine and use both categories.

We preprocess these generics by removing modifier clauses (e.g., "frogs are active at night, which is when the air is more humid" → "frogs are active at night"). This removes unnecessary information from the generics, including potential EXCEPTIONS (e.g., "lizards have legs, but some are legless" → "lizards have legs").

### A.2 *ExempliFI* System Details

We describe here the implementation details of our *ExempliFI* system and the tools used. We first use a *spacy* dependency parser to identify text spans for the concept, relation, and property in a generic. We use *inflect* to obtain plural and singular word forms and *mlconjug3*[33] to conjugate verbs[34].

**A.2.1 Generation.** To construct the QUDs used in the generation prompts (§4) we follow 3.2.2. Specifically, we use three templates to construct the QUDs, one each for concept-focused, property-

33 Additional conjugations were added to increase coverage and fix errors. These will be available with the data.
34 https://spacy.io/; https://pypi.org/project/inflect/; https://pypi.org/project/mlconjug3/

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




focused, and default. The templates are

| | |
|---|---|
| [wh-word] [*relation*] [PROPERTY] | (Concept-Focused QUD Template) |
| [wh-word] [CONCEPT] [*relation*] | (Property-Focused QUD Template) |
| What is true about [CONCEPT] | (Default QUD Template) |

For the concept-focused QUDs, we use "what" as the wh-word for all generics, since the replacement for the wh-word (i.e., the focused element) is a kind. For example, for the generic "cats are cute" the concept-focused QUD is"*what* is cute". However, in the property-focused QUDs, the replacement for the wh-word can be a location (e.g., "tigers live in *the jungle*"). Therefore, we use Wordnet (Fellbaum 2000) to identify location-related (i.e., needing "where") properties. In particular, we check for specific keywords in the set of hypernyms[35] for each property, which we obtain with *nltk*[36]. Since properties my be multiple words, we extract hypernyms for the root word of the property.

We generate EXEMPLARS using GPT-3 (Brown et al. 2020). Specifically, we use the *text-davinci-001* model with temperatue 0.9 and a max length of 100 tokens in the output. We use the best of 5 sequences for all generations.

**A.2.2 Output Processing.** While the prompts for focused EXCEPTIONS are designed to produce full sentences (i.e., complete EXEMPLARS), the prompts for default EXCEPTIONS and INSTANTIATIONS produce either a list of subtypes (i.e., INST. Prompt-I's and default Prompt-$D_1$) or a list of situations (default Prompt-$D_2$). Therefore, to obtain the complete set of generated candidate EXEMPLARS we use the generations from the INSTANTIATION and default EXCEPTION prompts to fill templates.

For the INSTANTIATIONS, we have two INSTANTIATION templates. The first uses the generated subtypes from Prompt-$I_K$ to construct concept-focused and default INSTANTIATIONS (Table 4(a)) while the second uses generations from Prompt-$I_P$ to construct property-focused INSTANTIATIONS (Table 4(b)). For the default EXCEPTIONS we also have two templates, one that directly uses subtypes generated from Prompt-$D_1$ (Table 4(c)) and another that converts generated situations from Prompt-$D_2$ into subtypes (Table 4(d)).

Since the focus EXCEPTIONS are generated as complete sentences, we need to ensure separately that they satisfy the constraints of the logical form. This means that for concept-focused EXCEPTIONS, the candidates end with the generic's property; for property-focused EXCEPTIONS, candidates should begin with the generic's concept. We remove any candidate EXCEPTIONS that do not fit these requirements.

**A.2.3 Filtering for Truthfulness.** We use GPT-3.5-Turbo (Ouyang et al. 2022) to predict whether each candidate generation is true or false. To do this, we first convert each EXEMPLAR into the singular, using the following prompt,

```
Put the following sentence into the singular: [EXEMPLAR].
```

We then ask the LLM whether the singular form is true with the following prompt

```
True or false: [EXEMPLAR-singular]?
```

For converting candidates to singular, we take only 1 generation from the LLM. For determining the truth of the singular candidates we take the majority vote of 5 responses. A response indicates

35 The hypernym set consists of the hypernyms for the primary synset of the target word and for three levels up. Keywords for location-related properties: "building", "geographical region".
36 https://www.nltk.org/

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| | ***GGSmall*** | | | ***GGTest*** | | | ***AnimalG*** | | |
|---|---|---|---|---|---|---|---|---|---|
| | Train | Dev | Test | Train | Dev | Test | Train | Dev | Test |
| **Focus EXCEPTIONS** | 485 | 57 | 51 | 916 | 123 | 129 | 172 | 113 | 128 |
| **INSTANTIATIONS** | 840 | 76 | 62 | 1258 | 123 | 182 | 162 | 113 | 122 |

Table 13: Statistics for the data used to train the validity discriminators.

the input is true if the text "true" is in the first non-empty, lowercased string of the response. Otherwise, the input candidate is predicted to be false.

**A.2.4 Validity Filtering.** After removing false candidate EXEMPLARS, we conduct a final filtering step to obtain a final set of valid EXEMPLARS. To filter the focused EXCEPTIONS, we train a discriminator to predict whether a statement is a valid focused EXCEPTION; we train an analogous discriminator for the INSTANTIATIONS. For each discriminator we fine-tune a *RoBERTA-large* model using training data annotated using the procedures described below (§A.3). The training data consists of EXEMPLARS generated for each of the three sources of generics used in our work (see §6.1 and §A.1). Dataset statistics are shown in Table 13.

For each discriminator, we conduct a hyperparameter search to obtain the best performing model. The selected hyperparameters, along with the respective model's accuracy and precision are shown in Table 14. Since the discriminators are used as filters, we prioritize minimizing the number of statements that are incorrectly predicted as valid when they are not (i.e., the precision of the "valid" class).

| | | **Hyperp.** | | ***GGSmall*** | | ***GGTest*** | | ***AnimalG*** | |
|---|---|---|---|---|---|---|---|---|---|
| | | B | LR | Acc. | $\text{Prec}^{(1)}$ | Acc. | $\text{Prec}^{(1)}$ | Acc. | $\text{Prec}^{(1)}$ |
| **Focus EXCEP.** | Dev | 24 | 1e-5 | 0.5614 | 0.5208 | 0.8374 | 0.8446 | 0.8407 | 0.8710 |
| | Test | | | 0.6667 | 0.7180 | 0.8372 | 0.8571 | 0.7890 | 0.8119 |
| **INST.** | Dev | 32 | 3e-5 | 0.7895 | 0.8438 | 0.7840 | 0.8167 | 0.8350 | 0.8866 |
| | Test | | | 0.7419 | 0.7097 | 0.7857 | 0.8099 | 0.7623 | 0.8046 |

Table 14: Hyperparameters and evaluation results on the development and test sets for the validity discriminators.

We remove all focused EXCEPTION and INSTANTIATION candidates predicted invalid by their respective discriminators. Finally, we rank the valid candidates. For the focused EXCEPTIONS and INSTANTIATIONS we use the trained discriminators to rank candidates. For the default EXCEPTIONS, we use a combination of perplexity and NLI contradiction probability to rank the candidates. We use GPT2-XL (Radford et al. 2019) to obtain perplexity ranking and a RoBERTa (Liu et al. 2019) model fine-tuned on MNLI[37] to obtain the NLI ranking.

### A.3 Annotations

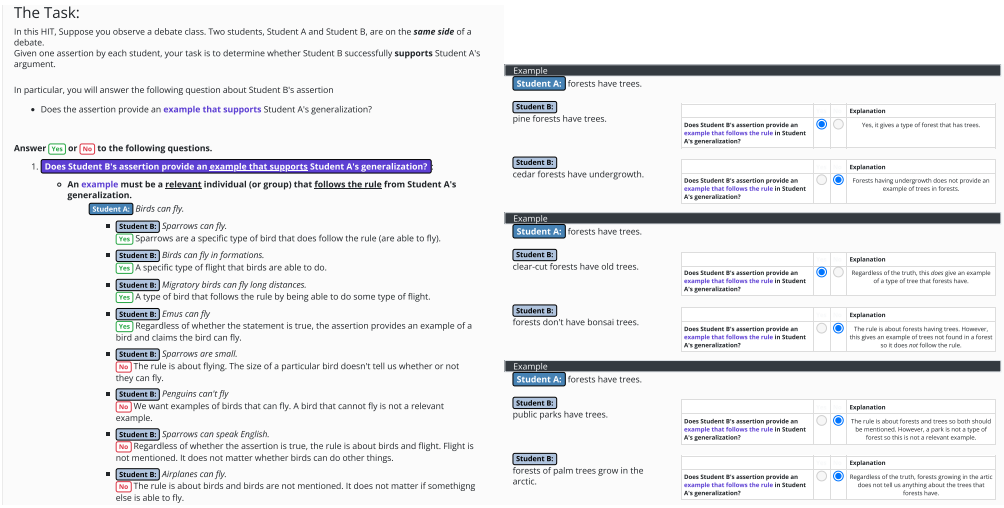
#### A.3.1 Validity Annotation.

*Annotation Task.* We use three separate annotation tasks to annotate the validity of candidate EXEMPLARS, one each for the INSTANTIATIONS (Fig. 5) and for the default (Fig. 6) and the focused (Fig. 7) EXCEPTIONS. All three tasks are framed as a debate between two students, where one students (Student A) asserts the generic and the other (Student B) replies with an EXEMPLAR.

37 https://huggingface.co/roberta-large-mnli

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




(a) Instructions (b) Examples

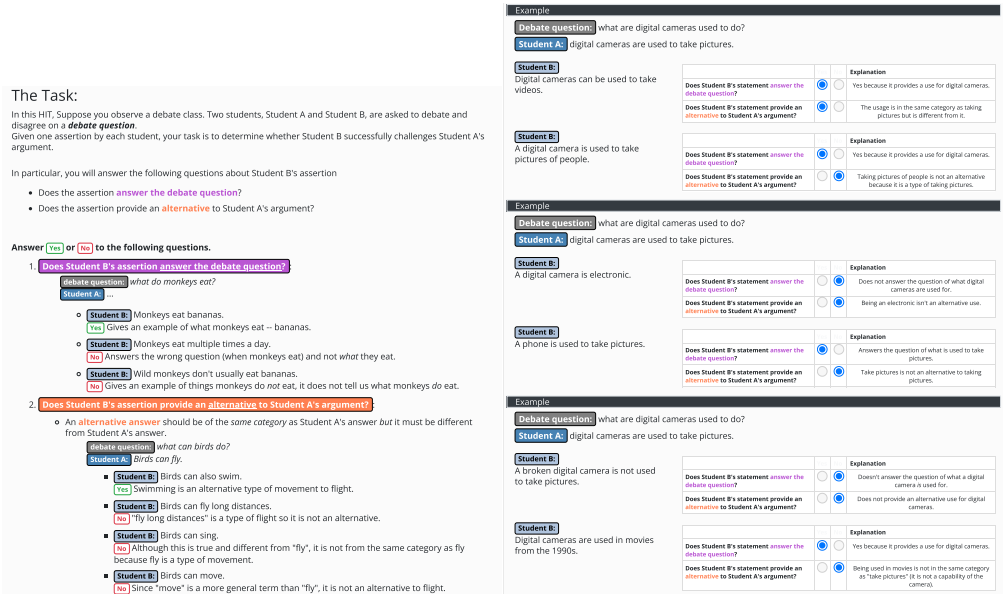
Figure 5: Annotation task for INSTANTIATIONS.


(a) Instructions (b) Examples

Figure 6: Annotation task for default EXCEPTIONS.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




(a) Instructions (b) Examples

Figure 7: Annotation task for focused EXCEPTIONS.

For the INSTANTIATION task, the two students are on the *same side* of the debate and the annotator's task is to determine whether Student B, in asserting the EXEMPLARS, provides an example supports Student A's assertion (i.e., an example where the generic applies). Full instructions and examples are shown in Fig. 5a and Fig. 5b respectively.

In contrast, when annotating EXCEPTIONS the two students are on opposing sides of the debate. The debate question provided as context to annotators is the QUD for the corresponding generic. For default EXCEPTIONS, annotators must decide if Student B provides an example that conflicts with follow Student A's assertion (i.e., the generic). See Fig. 6a and Fig. 6b for full instructions and examples. For the focused EXCEPTIONS, we check that the alternative provided is actually in the exotype (see §3.3) of the generic's focused element. In particular, as discussed in §4.2.3, for a concept-focused generic the alternative is valid if is is neither irrelevant nor a supertype or subtype of the corresponding generic's concept. Furthermore, the alternative must be for the focused element (e.g., the concept) of the generic. Therefore, annotators are asked to determine whether (1) the statement by Student B is actually related to the debate question (i.e., whether it answers the same QUD as the generic) and (2) alternative is valid (see Fig. 7a and Fig. 7b for full instructions and examples). Annotators must answer "yes" to both questions for the candidate to be deemed a valid focused EXCEPTION. The same procedure can be applied analogously for annotating property-focused EXCEPTIONS.

*Collected Data.* For training the INSTANTIATION and focused EXCEPTION discriminators we collect annotations using the validity tasks for a sample of our system's generated outputs generated. In particular, we have **annotators label 3055 generated EXEMPLARS** for 207 generics **from the *GGTest*** data, as well as **1045 generated EXEMPLARS** for 406 generics **from the *AnimalG*** data. The Fleiss' $\kappa$ and percentage agreement for each dataset and task are shown in Table 16.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| | Default EXCEPTIONS | | | Focus EXCEPTIONS | | | INSTANTIATIONS | | |
|---|---|---|---|---|---|---|---|---|---|
| | #Ex | $\kappa$ | % Agr. | #Ex | $\kappa$ | % Agr. | #Ex | $\kappa$ | % Agr. |
| Our Setup | 237 | 0.3238 | 0.7300 | 1224 | 0.3465 | 0.7685 | 1440 | 0.2464 | 0.7296 |
| *PenguinsSetup* | 532 | 0.3367 | 0.7392 | 2912 | 0.3001 | 0.6758 | 2946 | 0.1863 | 0.6650 |

Table 15: Inter-annotator agreement for the annotations used in human evaluation. *Our Setup* indicates evaluation from the main portion of the paper (§6), *PenguinsSetup* indicates annotations used for the comparison of annotation procedures (see §A.3).

| | Default EXCEPTIONS | | | Focus EXCEPTIONS | | | INSTANTIATIONS | | |
|---|---|---|---|---|---|---|---|---|---|
| | #Ex | $\kappa$ | % Agr. | #Ex | $\kappa$ | % Agr. | #Ex | $\kappa$ | % Agr. |
| *GGTest* | 294 | 0.3584 | 0.7120 | 1168 | 0.2935 | 0.7146 | 1593 | 0.2836 | 0.6874 |
| *AnimalG* | 221 | 0.0812 | 0.6561 | 413 | 0.3019 | 0.8309 | 411 | 0.4231 | 0.7568 |

Table 16: Inter-annotator agreement for the validity annotations used to train and evaluate the discriminators.

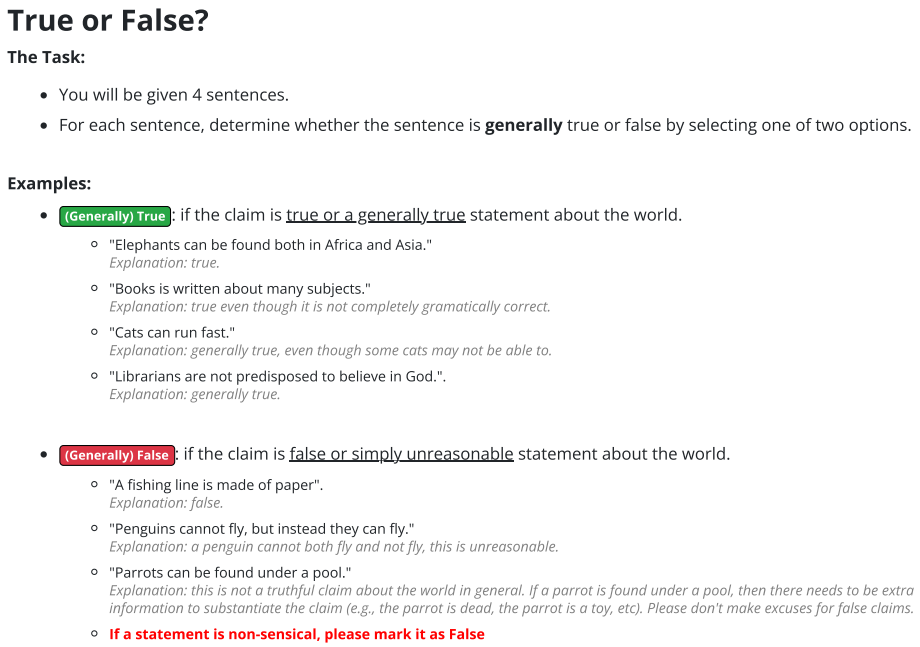
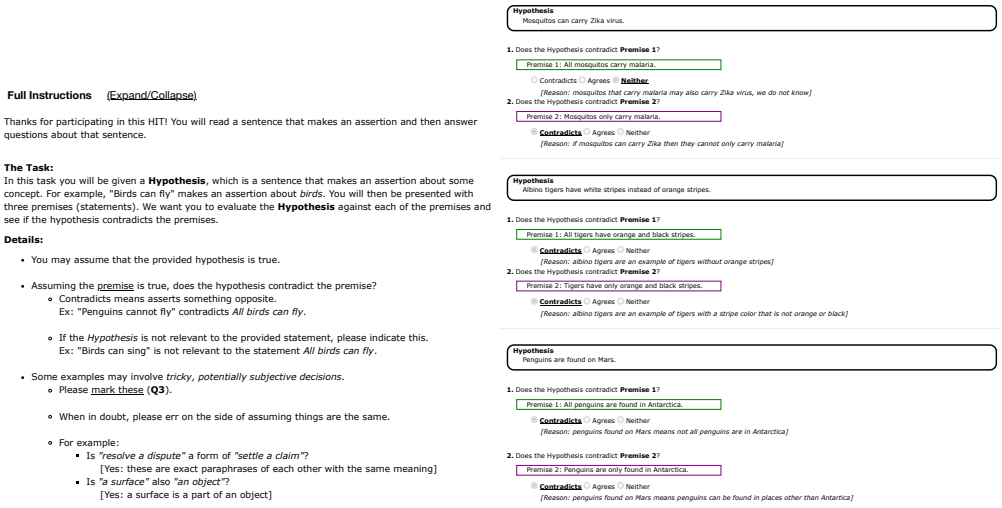



Figure 8: Annotation task for truthfullness of generated EXEMPLAR candidates.

For conducting a human evaluation of system generations (§6.4), we collect annotations for EXEMPLARS from 96 generics from *GGSmall* for which our system and both baseline systems (i.e., the system proposed by Allaway et al. (2023) and the corresponding GPT-3 baseline; see §6.3 for more details) each produce five EXCEPTIONS and five INSTANTIATIONS. That is, we have annotators label **1440 EXCEPTIONS and 1440 INSTANTIATIONS** using the annotation tasks described in §A.3. The Fleiss' $\kappa$ and percentage agreement for each dataset and task are shown in the first row of Table 15.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




(a) Instructions (b) Examples

Figure 9: Annotation task for *PenguinsSetup* (§A.3.3) from Allaway et al. (2023).

**A.3.2 Truthfulness Annotation.** To annotate generated candidate EXEMPLARS for truthfulness, we ask annotators to determine whether a candidate is generally true or generally false. Annotators are instructed to consider nonsensical statements as false. Full instructions are given in Figure 8.

We collect truthfulness annotation in order to validate the quality of the truthfulness filter used in *ExempliFI* (i.e., GPT-3.5-Turbo). Specifically, we collect annotations for a set of 500 randomly sampled EXEMPLARS generated from *AnimalG* generics, 100 EXEMPLARS of each type (default EXCEPTIONS, concept- and property-focused EXCEPTIONS, concept- and property-focused INSTANTIATIONS). The Fleiss' $\kappa$ is $0.4407$ and the percent agreement is $0.7507$.

**A.3.3 Penguins Annotation.** We validate the quality of generations from our *ExempliFI* system by using the annotation tasks from Allaway et al. (2023) (henceforth called *PenguinsSetup*). In particular, we use the *PenguinsSetup* to collect validity annotations for our generated EXEMPLARS, as well as for the EXEMPLARS from both the baseline systems. This is to check that our system outperforms the baselines in the setting they were designed for.

In *PenguinsSetup*, INSTANTIATIONS are annotated by asking annotators are asked whether an INSTANTIATION contradicts the original generic; valid INSTANTIATIONS will agree with the generic. In contrast, for EXCEPTIONS annotators are asked whether an EXCEPTION contradicts two modified forms of the generic. Specifically, whether the EXCEPTION contradicts (i) the generic prefixed with "all" or (ii) the generic with "only" added as a modifier on the property. For example, the two modified forms of the generic "birds can fly" are (i) "all birds can fly" and (ii) "birds can fly only". Allaway et al. (2023) posit that default EXCEPTIONS will contradict form (i) and property-focused EXCEPTIONS will contradict form (ii). Note that because they do not generate concept-focused EXCEPTIONS there is no condition to check the validity of such EXCEPTIONS. Full annotation instructions and examples are shown in Figure 9.

Using *PenguinsSetup*, for a random subset of 200 generics from *GGSmall*, we collect annotations on the validity of the top five INSTANTIATIONS and top five EXCEPTIONS generated by our system and both baseline systems. Agreement measures are shown in the second row of Table 15.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| | Baseline Template | Prompt In-Context Examples |
|---|---|---|
| **(i)** | $[K_{\text{SUBTYPE}}]$ $[r]$ $[P]$ | Birds can fly. For example, seagulls can fly.<br>Dogs protect buildings from intruders. For example, pitbulls protect buildings from intruders.<br>Ducks lay eggs. For example, female ducks lay eggs. |
| **(ii)** | $[K]$ $[r]$ $[P_{\text{SUBTYPE}}]$ | Viruses are spread through body fluids. For example, viruses are spread through saliva.<br>Dogs protect buildings from intruders. For example, dogs protect some private homes from intruders.<br>Cowsheds are found on farms. For example, cowsheds are found on dairy farms. |
| **(iii)** | $[K_{\text{SUBTYPE}}]$ $[r]$ $[P_{\text{SUBTYPE}}]$ | Birds can fly. For example, Canadian geese fly long distances to migrate.<br>Ostriches lay eggs. For example, female ostriches lay large spotted eggs.<br>Elephants are found in zoos. For example, African elephants are found in most large zoos. |
| **(iv)** | $[K_{\text{SUBTYPE}}]$ $[\neg r]$ $[P]$ | Birds can fly. But also penguins cannot fly.<br>Ducks lay eggs. But also male ducks do not lay eggs.<br>Dogs protect buildings from intruders. But also very small dogs do not protect buildings from intruders. |
| **(v)** | $[K]$ $[\neg r]$ $[P_{\text{SUBTYPE}}]$ | Dogs protect buildings from intruders. But also dogs do not protect apartment buildings from intruders.<br>Cowsheds are found on farms. But also cowsheds are not found in orchards.<br>The sun produces radiation. But also the sun does not produce x-rays. |
| **(vi)** | $[K]$ $[r]$ $[\sim P]$ | Elephants are found in zoos. But also elephants are found in the wild in Africa.<br>Viruses are spread through body fluids. But also viruses are spread in the air.<br>A hair dryer is used to dry hair. But a hair dryer can also be used to dry clothes. |
| **(vii)** | $[K_{\text{SUBTYPE}}]$ $[r]$ $[\sim P]$ | Elephants are found in zoos. But also African elephants are found in the wild in Africa.<br>Viruses are spread through body fluids. But also coronaviruses are spread in the air.<br>A hair dryer is used to dry hair. But also an electric hair dryer can be used to dry clothes. |
| **(viii)** | $[\sim K]$ $[r]$ $[P]$ | Elephants are found in zoos. But also giraffes are found in zoos.<br>Dogs protect buildings from intruders. But also security cameras protect buildings from intruders.<br>A hair dryer is used to dry hair. But also a towel can be used to dry hair. |

Table 17: Prompts for GPT-3 Baseline from Allaway et al. (2023). Note that (viii) is added in this work to adapt the GPT-3 baseline to generate concept-focused EXCEPTIONS.

### A.4 Generation Baselines

To generate EXEMPLARS, both baseline systems use seven templates, four for EXCEPTIONS and three for INSTANTIATIONS (see Table 10). Of the three INSTANTIATION templates used by the baselines, (i) and (ii) are the same as the templates we use to construct INSTANTIATIONS (see Table 4 (a) and (b)). Similarly, template (iv) is the same as the template (c) in our system (Table 4) for generating default EXCEPTIONS. Additionally, template (vi) follows the logical form for property-focused EXCEPTIONS used in our system (Table 2).

In order to evaluate the baseline generations using our proposed annotation setup (§A.3) we treat generations from templates (iv) and (v) as default EXCEPTIONS and the generations from (vi) and (vii) as property-focused EXCEPTIONS. Additionally, we treat generations from

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




template (i) as concept-focused INSTANTIATIONS and generations from templates (ii) and (iii) as property-focused INSTANTIATIONS[38].

*Constrained Decoding (ConstraintDec).* The system proposed by Allaway et al. (2023) (***ConstraintDec***) uses the NeuroLogic A⋆esque (NeuroLogic⋆) (Lu et al. 2022) constrained decoding algorithm to generate EXEMPLARS from GPT2-XL. Following the templates in Table 10, this system constructs a generation prompt (from the input template) and a set of lexical constraints (from the output template) that should be satisfied during decoding. The lexical constraints specify $n$-grams that should be included in or excluded from the generated output (e.g., exclude "fly", "flying", "flew", etc.). The NeuroLogic⋆ algorithm outputs a sequence that has both high likelihood and high satisfaction of the specified constraints. The generated candidates are then filtered for truthfulness and validity using trained discriminators.

*GPT-3 Baseline.* The GPT-3 baseline used by Allaway et al. (2023) uses few-shot prompting to illustrate the desired template for generation. Each prompt consists of three in-context examples (see Table 17). Note that we add an additional prompt to generate concept-focused EXCEPTIONS (see (viii) Table 17). This baseline uses the *davinci* model with top-p sampling 1.0, temperature 0.8, maximum length 50 tokens and top 5 sequences. As with the constrained decoding system, the generated candidates are filtered for truthfulness.

### A.5 LLMs and Generics Probe Details

For the quantification probe, we slightly modify the decoder-only prompts from Table 7 for LLAMA-2. Specifically, we use the following prompts with LLAMA-2. For the probe without default EXCEPTIONS we use

> What are the 5 best quantifiers to fill in the [blank] in the sentence?
> Sentence: [blank] {birds can fly}$_G$.
> Answer (list 5 quantifiers):
> 1.

and for the probe with default EXCEPTIONS we use

> What are the 5 best quantifiers to fill in the [blank] in the sentence?
> Sentence: {birds can fly}$_G$. However, {penguins cannot fly}$_E$. Therefore, [blank] {birds can fly}$_G$.
> Answer (list 5 quantifiers):
> 1.

where $G$ indicates the generic and $E$ the default EXCEPTION.

For the generations from LLAMA-2, we use a maximum token length of 120. For both the quantification and inheritance the generations from GPT-3, GPT-3.5-Turbo, and GPT-4, we use: a maximum token length of 100, temperature 0.9, presence penalty 0.0, frequency penalty 0.0, and top-$p$ of 1.0.

## B. Supplementary Results

In this section we discuss supplementary analyses and results. Specifically, we discuss the results of our quantifier probe using metrics from prior work (§B.1) and the validation experiments for

38 Since template (iii) contains subtypes of both the concept and the property, these generations could be considered either property or concept-focused INSTANTIATIONS. We arbitrarily choose to consider them property-focused.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




the wording used in our property inheritance probes (§B.2). We also include analyses comparing the annotation procedures proposed in this work to those from prior work (§B.3).

### B.1 GOG Effect and Quantifiers

We include here additional results, including exploration of an alternative prompt strategy for examining the GOG effect through quantifiers.

**B.1.1 *Top Quantifiers* Probe.** Following prior work (Ralethe and Buys 2022), we include here analysis of the GOG effect in multiple encoder-only LMs. Specifically, we use mask infilling to determine the associations between quantifiers and generics. That is, we insert a mask token before the generic and then take the top five tokens predicted to replace the mask. For example, for the generic "birds can fly" the input would be "`<mask> birds can fly`", where `<mask>` is a special token of the LM. As in §5.1, we run this probe with and without default EXCEPTIONS (e.g., "`Birds can fly. However, penguins cannot fly. Therefore, <mask> birds can fly`"). Similarly, as with our main *Top Quantifiers* probe in §5.1, we measure the frequency of the universal quantifiers "all" and "every" among the elicited quantifiers.

The encoder-only models we use are[39]:

- **BERT**: A bidirectional transformer-based language model trained with two objectives: MLM and next sentence prediction (Devlin et al. 2018).
- **RoBERTa**: A BERT model trained without the next-sentence prediction task as well as longer sequences and more data (Liu et al. 2019).
- **ALBERT**: A BERT model trained with parameter reduction techniques and an additional loss component to increase inter-sentence coherence (Lan et al. 2019). The model has 11M parameters, compared to BERT's 340M.
- **ELECTRA**: A bidirectional transformer-based language model trained with an alternative to MLM (Clark et al. 2020). In particular, the input is corrupted by replacing random tokens with a sampled alternative (rather than a mask). Then during training, the objective is to predict whether or not a token has been replaced.

These models are chosen to cover a sample of popularly used bidirectional LMs.

We report our results for these models in Figure 10. We observe a GOG effect across all the LMs. That is, universal quantifiers are supplied in the top five infilling tokens for a non-negligible percentage of generics for all models. Furthermore, we also observe that the GOG effect increases when EXCEPTIONS are included in the prompt. Recall that we also observe an increased GOG effect with GPT-3.5-Turbo and LLAMA-2 (see §5.1.2), which we posit is in part due to the substantially lower variety in unique modifiers, particularly multi-word modifiers, generated by GPT-3.5-Turbo/LLAMA-2 LLMs. Since mask infilling does not *allow* the bidirectional models to produce multi-word modifiers (e.g., it cannot produce "not all"), a similar explanation may apply here.

Note that, in prior work, Ralethe and Buys (2022) measure the GOG effect with a similar infilling probe to our *Top Quantifiers* probe. As metrics they compute precision at 5 (P@5) and Mean Reciprocal Rank (MRR). P@5 measures the proportion of universal quantifiers that occur in the quantifiers returned by the probe. In contrast, MRR measures how highly ranked the universal

39 BERT—`bert-large-uncased`; RoBERTa—`roberta-large`; ALBERT—`albert-large-v2`; ELECTRA—`google/electra-large-generator`.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530



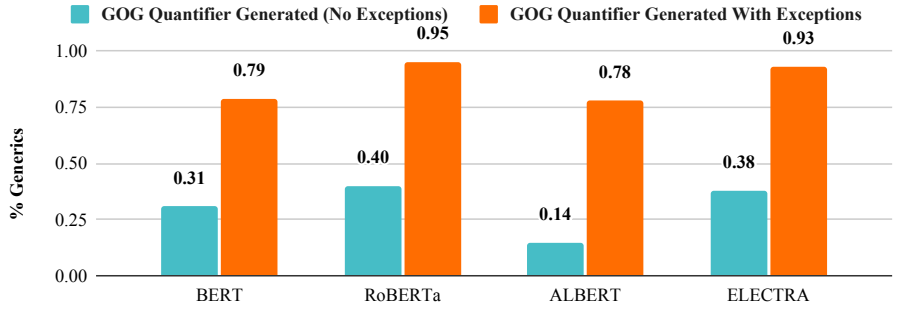
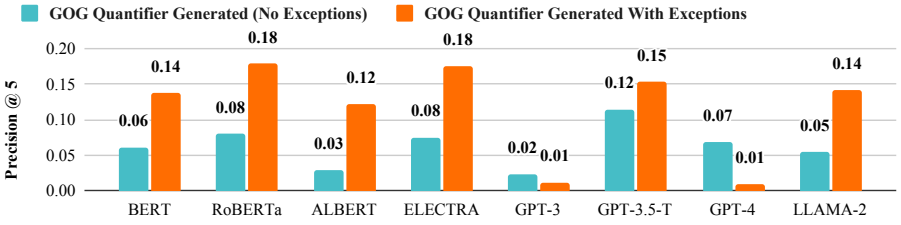
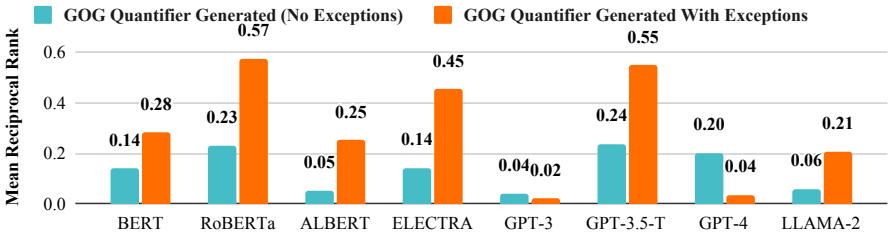




Figure 10: Results of the *Top Quantifiers* probe from §5.1 adapted to bidirectional LMs with infilling. Percentage of generics where a GOG quantifier (i.e., one of the universals "all" or "every") is in the top 5 tokens supplied by the LM to quantify the generic. *With Exceptions* indicates the prompts include default EXCEPTIONS. Higher values indicate a larger amount of overgeneralization by the LM.




(a) Precision at 5 ($P$@5). Note that since two quantifiers are used ("all" and "every"), the maximum $P$@5 is 0.4).




(b) Mean Reciprocal Rank (MRR).

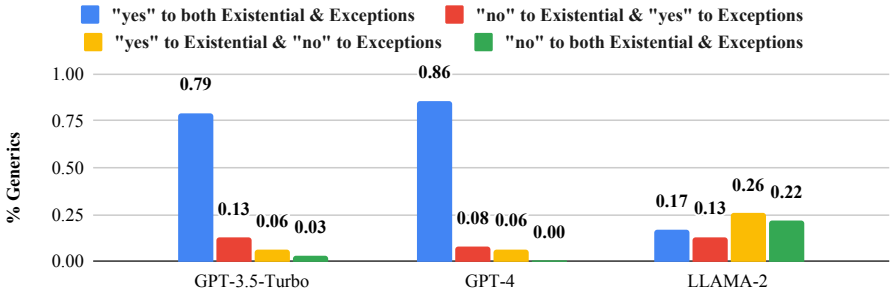
Figure 11: Results of the *Top Quantifiers* probe evaluated using the metrics from Ralethe and Buys (2022). GOG quantifiers are the universals "all" and "every". *With Exceptions* indicates the prompts include default EXCEPTIONS. Higher values indicate a larger amount of overgeneralization by the LM.

quantifiers are, if they occur. We report the results from our investigations using these metrics in Figure 11a and Figure 11b.

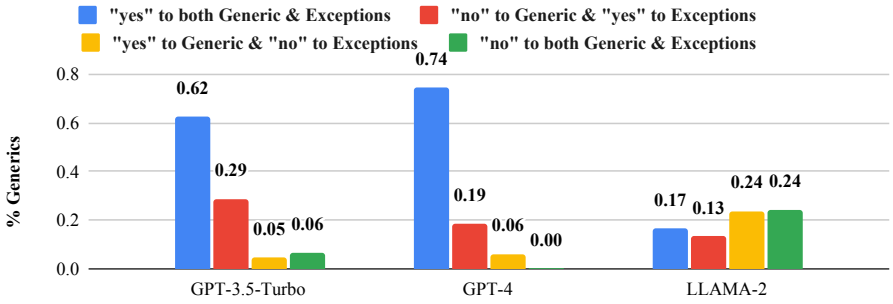
Across the LLMs for which the GOG effect increases in the presence of EXCEPTIONS, we see that the increases in MRR are generally greater than in P@5. Therefore, universal quantifiers are not only being produced more frequently they are also being ranked higher in the generated modifiers.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




(a) Responses for the generic presented in its original form (i.e., as a generic).




(b) Responses for the generic presented as a universally quantified statement (i.e., quantified with "some").

Figure 12: Supplementary results from the *Leslie Questions* probe. Percentage of generics with a particular response to the generic in some form and to the default EXCEPTIONS.

**B.1.2 *Psychology-based Questions* Probe.** We present here (Fig. 12) the results for statements in their generic form statements and in their existential form from the *Psychology-based Questions* probe in §5.1. We observe with both GPT-3.5-Turbo and GPT-4 a relatively high level of endorsement of both the generic and the existentially quantified statements. For LLAMA-2, we observe that only a small portion of responses fall into each slice, suggesting that the model is simply unable to respond appropriately, similar to what we find with other probes.

**B.1.3 Alternative Prompting Strategy.** To explore the impact of prompt choice on observed GOG effect in LMs, we experiment with an additional prompt for a sample of generics. Specifically, we reformulate the generic as a question and then ask the models to answer it. As with the *Leslie Quesitons* probe in §5.1 we experiment with three versions of the generic: the generic in its base form, a universally quantified version of the generic (i.e., quantified with "all"), and an existentially quantified version of the generic (i.e., with "some"). Additionally, we provide the (potentially quantified) question form of the generic to the model with and without default EXCEPTIONS. For example, for the generic "birds can fly", without EXCEPTIONS we query the model with three separate prompts of the form "`Can [quantifier] birds fly?`", where `[quantifier]` is either nothing (for the generic itself), or one of "all" and "some" (e.g., "`Can all birds fly?`"); including EXCEPTIONS we again have three prompts of the form "`[EXCEPTION]. Can [quantifier] birds fly?`", where `[EXCEPTION]` is a

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




default EXCEPTION to the generic (e.g., "penguins cannot fly") and `[quantifier]` is again one of the three quantification options.

As with the *Psychology-based Questions* probe, we run this prompting strategy on a sample of 1000 generics from the *AnimalG* dataset for which each has at least three valid default EXCEPTIONS. Specifically, we use the same 1000 generics and accompanying EXCEPTIONS as with the *Psychology-based Questions* probe. Note that querying with the generic as a generic does not probe the GOG effect, even if EXCEPTIONS are included. This is because, if the generic is felicitous, the answer should be "yes" regardless of whether EXCEPTIONS are provided. If the LM respondes "no" to the generic than it is possible that the generic is not felicitous (assuming a perfectly correct LM); more likely, the model is either ignorant of the generic or simply unable to respond appropriately to the input stimulus.

We present the results of the probe in Figure 13. We observe a GOG effect across models (Fig. 13a). For GPT-3.5-Turbo and GPT-4 we observe a decrease when EXCEPTIONS are included in the prompt, while for LLAMA-2 we observe an increase in the effect. Note however that LLAMA-2's behavior in response to the prompt is far from sensible (Fig. 13b). In particularly, LLAMA-2 responds "yes" to only $10\%$ of the generics when presented in their generic form (compared to $62\%$ and $72\%$ for GPT-3.5-Turbo and GPT-4) and only $40\%$ of the generics when they are existentially quantified (compared to $83\%$ and $91\%$ for GPT-3.5-Turbo/GPT-4). This suggests that LLAMA-2's behavior is likely indicative of a broad failure by the model to respond appropriately to the prompt, regardless of the generic it contains.

We note that this prompting strategy measures the GOG effect differently from the main *Top Quantifiers* probe in §5.1. In particular, our *Top Quantifiers* probe aims to capture potential variation in the LMs' responses by asking them to generate multiple quantifiers. This can be seen as evaluating how the GOG effect might impact models when used for generation tasks. In contrast, this prompt captures instead how the model might behave in a classification-like scenario, akin to the supplementary *Psychology-based Questions* probe. Regardless, as with our other probes we observe that LMs do in fact exhibit a GOG effect.

### B.2 Property Inheritance Validation

To validate the wording used in our property inheritance probe, we manually construct a set of questions to test the LLMs. Specifically, the questions evaluate whether the models recognize that the wording of the prompt asks them to consider deductive reasoning (see templates in Table 18). We use the following five nonsense words in all property inheritance evaluations: "Dofik", "Yeb", "Wumox", "Bafu", "Goq". For the validation questions we use the following five properties: "has three sides", "is red", "eats oranges", "lives in New Zealand", and "swims quickly".

The overall accuracy of the models LLMs evaluated on these questions is $0.820$ (GPT-3.5-Turbo), $0.932$ (GPT-4), and $0.864$ (GPT-3).

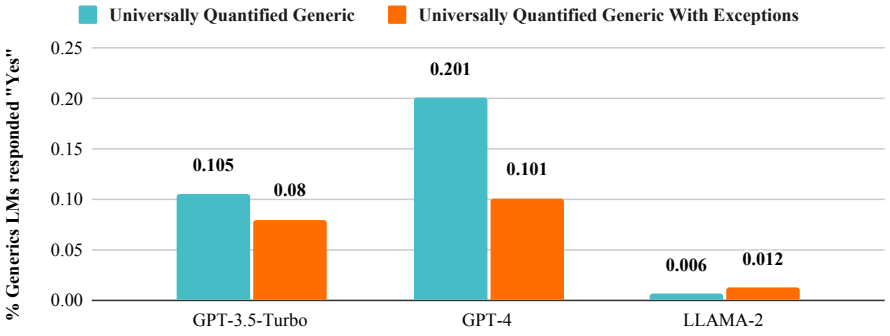
### B.3 Comparison of Annotation Procedures

We note that the precision scores obtained by our collected human annotations for *GPT3-baseline* and *ConstraintDec* are around $10$ points higher than those reported by Allaway et al. (2023). These annotations were collected using the new annotation setup we developed (§6.2) that better aligns with the logical forms for the EXEMPLARS. Therefore, we also conduct a human evaluation using the annotation setup from Allaway et al. (2023) (***All-Only Setup***). As before, we compute precision at $k$.

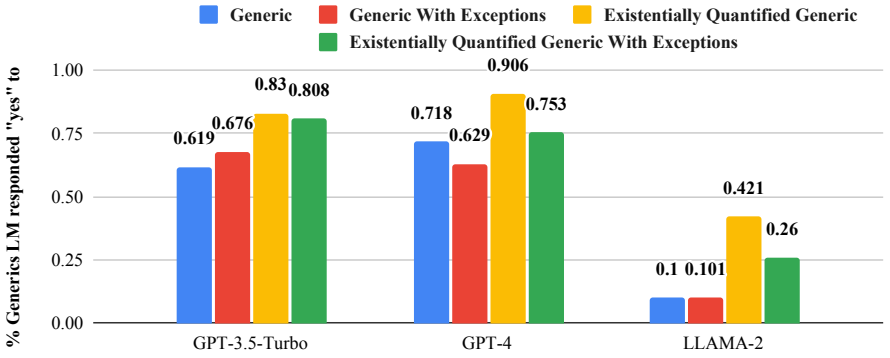
We observe first that the precision across systems decreases substantially in the *All-Only Setup*. However, across both EXCEPTIONS and INSTANTIATIONS our system outperforms both baselines. This further highlights the strength of our system.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




(a) Percentage of generics where the models responded "yes" to the universally quantified form.




(b) Percentage of generics where the model responded "yes" to the generic form of the generic or to the existentially quantified form of the generic.

Figure 13: Responses to the alternative prompt formulating generics as questions. *With Exceptions* indicates the prompt included a default EXCEPTION. Higher values indicate larger amounts of overgeneralization.

*Are the procedures comparable?*. We note that for EXCEPTIONS, the precision we find in the *All-Only Setup* is similar to that reported in Allaway et al. (2023); they report ∼0.624 for *ConstraintDec*, 0.54 for *GPT-3-baseline*. However, for INSTANTIATIONS the precision is substantially lower (∼0.897 for *ConstraintDec* and ∼0.724 for *GPT-3-baseline*). We hypothesize that this is due to the difficulty of the annotation task. Even after removing annotators with low competence, the Fleiss' $\kappa$ for agreement is only 0.1863 for INSTANTIATIONS in the *All-Only Setup*. In contrast, the $\kappa$ is approximately 0.32 for the EXCEPTIONS in the same annotation setup. Despite the increased annotation difficulty, our system still outperforms both baselines.

*How are incorrectly formatted candidates handled?*. Our new annotation setup enforces that generations fit the desired template in order to be valid, while the *All-Only Setup* does not. As noted by Allaway et al. (2023), the *GPT-3-baseline* generations often do not adhere to the required template. Therefore, generations deemed invalid in our annotation setup due to having the incorrect format could be marked valid in the *All-Only Setup*, thereby increasing precision. This is likely why the precision of *GPT-3-baseline* for EXCEPTIONS actually increases under the *All-Only Setup*.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| | **Question** | **Label** |
|---|---|---|
| (1) | Premises: All $t_1$ [property]. A $t_2$ is a $t_1$. Conclusion: Therefore, all $t_2$ [property]. | Yes |
| (2) | Premises: All $t_1$ [property]. A $t_2$ is not a $t_1$. Conclusion: Therefore, all $t_2$ [property] | No |
| (3) | Premises: All $t_1$ [property]. A $t_2$ is a $t_1$. Conclusion: Therefore, all $t_2$ [not-property]. | No |
| (4) | Premises: All $t_1$ [property]. A $t_2$ is not a $t_1$. Conclusion: Therefore, all $t_2$ [not-property]. | No |
| (5) | Premises: Some $t_1$ property. A $t_2$ is a $t_1$. Conclusion: Therefore, all $t_2$ property. | No |
| (6) | Premises: Some $t_1$ [property]. A $t_2$ is a $t_1$. Conclusion: Therefore, some $t_2$ [property]. | Yes |
| (7) | Premises: Some $t_1$ [property]. A $t_2$ is a $t_1$. Conclusion: Therefore, all $t_2$ [not-property]. | No |
| (8) | Premises: Some $t_1$ [property]. A $t_2$ is not a $t_1$. Conclusion: Therefore, all $t_2$ [property]. | No |
| (9) | Premises: Some $t_1$ [property]. A $t_2$ is not a $t_1$. Conclusion: Therefore, some $t_2$ [property]. | No |
| (10) | Premises: Some $t_1$ [property]. A $t_2$ is not a $t_1$. Conclusion: Therefore, all $t_2$ [not-property]. | No |

Table 18: Templates for constructing the questions to validate the wording of the property inheritance probe. $t_1$ and $t_2$ are placeholders for nonce types, [property] is a placeholder for a property to be inherited and [not-property] is the negation of that property.

| | EXCEPTIONS | | INSTANTIATIONS | |
|---|---|---|---|---|
| | P@1 | P@5 | P@1 | P@5 |
| *GPT-3-baseline* | 0.6610 | 0.5814 | 0.5329 | 0.4695 |
| *ConstraintDec* | 0.5847 | 0.6237 | 0.5449 | 0.5246 |
| *ExempliFI* | **0.6695** | **0.6763** | **0.6168** | **0.6072** |

Table 19: Precision at $k$ ($P@k$) results for human evaluation using the annotation procedure from Allaway et al. (2023).

To see why this is the case, consider the following default EXCEPTION generated by *GPT-3-baseline*:

**Generic:** a jungle gym can be a great place to work out.
***GPT-3-baseline* candidate:** a jungle gym could be a great place to get hurt.

In our setup, the format is enforced by having separate annotation tasks (with different instructions and examples) for the default and focused EXCEPTIONS (§6.2). Therefore, this example would be marked invalid because it does not provide a counterexample to the generic (i.e., does not provide a type of jungle gym that is not good for working out; see annotation instructions in Fig. 6). In contrast, the *All-Only Setup* uses two questions, asked together, for the default and focused EXCEPTIONS. In particular, the *All-Only Setup* asks whether the EXEMPLAR contradicts: (1) the generic prefixed with "all" (i.e., "*All* jungle gyms are a great place to work out") and (2) the generic with "only" (e.g., "jungle gyms can *only* be a great place to work out"). Therefore, in

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




the *All-Only Setup*, the example candidate would be marked valid because it contradicts question (2). This occurs despite the example *not* contradicting question (1), which was intended for to identify valid default EXCEPTIONS[40].

*How successfully are alternatives identified?*. In addition to allowing incorrectly formatted EXCEPTIONS to be valid, the *All-Only Setup* places less restrictions on the focused generics. Consider the following example from *ConstraintDec*:

> **Generic:** hockey is a game.
> ***ConstraintDec* candidate:** hockey can be a lot more than that.

This was annotated as a valid EXCEPTION in the *All-Only Setup* because it contradicts "hockey is *only* a game" (i.e., annotation question (2)). However, this candidate does not actually contain a valid alternative to "game". In other words, it does not answer the property-focused QUD for the generic (i.e., "what hockey is") and is therefore an invalid EXCEPTION. Since our annotation setup directly queries the relationship between the candidate and the QUD, it produces the correct label (invalid).

Even in cases where an alternative property is included, the *All-Only Setup* can fail to catch irrelevant alternatives. For example, consider the candidate from *ConstraintDec* and our system:

> **Generic:** drummers are taught to play with their hands.
> ***ConstraintDec* candidate:** a drumming instructor is taught to work with the student.
> **Our system candidate:** drummers are taught to use a combination of their hands and feet.

Both candidates would be marked valid in the *All-Only Setup* because they contradict "drummers are *only* taught to play with their hands". However, the *ConstraintDec* candidate is invalid because "work with the student" is not an alternative to "play with their hands". While it *is* something drummers are taught, the generic is about how drummers are taught to play and so the alternatives should be other ways of playing (e.g., with their feet). Although the candidate from our system does include an alternative way of playing ("a combination of their hands and feet"), this is still not a valid alternative; "hands and feet" overlaps with "hands" and so is not fully different. Both candidates are correctly marked as invalid EXCEPTIONS in our annotation setup.

### B.4 Additional Comparison Baseline for *ExempliFI*

We include here a proof-of-concept for using GPT-3.5-Turbo as a baseline for evaluating our *ExempliFI* system in §6.4. For this baseline, we use *gpt-3.5-turbo* with a temperature of 0.9 and max length of 100 tokens. We use the prompts from the GPT-3 baseline (Table 17) that align with the EXEMPLARS in this work. In particular, we use the following prompt-EXEMPLAR type pairs: (i) for concept-focused INSTANTIATIONS, (ii) for property-focused INSTANTIATIONS, (iv) for default EXCEPTIONS, (vi) for property-focused EXCEPTIONS, and (viii) for concept-focused EXCEPTIONS.

We evaluate this baseline on the sample of eight generics shown in §4.3. Note that because "Birds fly" is one of these generics, we must adjust two of the prompts to not include this example. In particular, we replace "Birds can fly. For example seagulls can fly" in (i) with "Bats can fly. For example, fruit bats can fly"; in (iv) we replace the relevant example with "Tigers are orange and black. But also albino tigers are not orange and black".

40 Recall that the generations from templates (iv) and (v) in Table 10 are considered default EXCEPTIONS, while the generations from templates (vi) and (vii) are considered property-focused EXCEPTIONS.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| | ***AnimalG Data*** | |
|---|---|---|
| **(a)** | ***Generic:*** *Cats sleep in trees* | |
| | **Concept-Focused INSTANTIATIONS:**<br>• Serval cats sleep in trees.<br>• Tabby cats sleep in trees.<br>• Persian cats sleep in trees.<br><br>**Default EXCEPTIONS:**<br>• Cats prefer to sleep indoors.<br>• Some cats do not like climbing trees and prefer to sleep on the ground. | **Property-Focused INSTANTIATIONS:**<br>• Cats sleep in branches of tall trees.<br>• Cats sleep in tall trees in the forest.<br><br>**Property-Focused EXCEPTIONS:**<br>• Cats sleep in cozy beds indoors.<br>• Cats sleep in sunny spots by the window.<br><br>**Concept-Focused EXCEPTIONS:**<br>• Birds sleep in trees.<br>• Squirrels also sleep in trees. |
| **(b)** | ***Generic:*** *Birds fly.* | |
| | **Concept-Focused INSTANTIATIONS:**<br>• Eagles can fly.<br>• Hawks can fly.<br><br>**Property-Focused INSTANTIATIONS:**<br>• Birds fly south for the winter.<br><br>**Concept-Focused EXCEPTIONS:**<br>• Bats are able to fly.<br>• Airplanes are able to fly. | **Property-Focused EXCEPTIONS:**<br>• Birds are able to swim.<br>• Birds are able to migrate thousands of miles across continents.<br><br>**Default EXCEPTIONS:**<br>• Penguins are birds that cannot fly.<br>• Some birds, such as ostriches and emus, are flightless. |
| **(c)** | ***Generic:*** *Moose have winter coats.* | |
| | **Concept-Focused INSTANTIATIONS:**<br>• Moose have thick winter coats to keep them warm in winter.<br><br>**Property-Focused INSTANTIATIONS:**<br>• Moose have thick fur coats to keep them warm in cold climates.<br>• Moose have thick, insulated fur that keeps them warm during the cold winter months.<br><br>**Property-Focused EXCEPTIONS:**<br>• Moose have summer coats. | **Default EXCEPTIONS:**<br>• Moose shed their winter coats in the summer.<br>• Some moose living in warmer climates may not develop a thick winter coat.<br><br>**Concept-Focused EXCEPTIONS:**<br>• Bears have thick fur to keep them warm in winter.<br>• Reindeer have thick fur to keep them warm in winter. |
| **(d)** | ***Generic:*** *Deer live in meadows.* | |
| | **Concept-Focused INSTANTIATIONS:**<br>• White-tailed deer live in meadows.<br><br>**Property-Focused INSTANTIATIONS:**<br>• Deer live in forests and grasslands.<br>• Deer can often be seen grazing in meadows.<br><br>**Default EXCEPTIONS:**<br>• Some deer can live in forests or mountains instead of meadows. | **Concept-Focused EXCEPTIONS:**<br>• Rabbits live in meadows.<br>• Foxes live in meadows.<br>• Horses live in meadows.<br><br>**Property-Focused EXCEPTIONS:**<br>• Deer can be found in forests and other types of habitats. |

Table 20: Examples of EXEMPLARS generated by the proof-of-concept baseline using GPT-3.5-Turbo (§B.4) for generics from *AnimalG*.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




| | |
|---|---|
| | ***GGTest Data*** |
| **(e)** | ***Generic:*** *a scavenger hunt is a fun way to spend an afternoon with friends.*<br>**Concept-Focused INSTANTIATIONS:**<br>• A scavenger hunt is a fun way to spend an afternoon with friends in a park.<br>**Property-Focused INSTANTIATIONS:**<br>• A scavenger hunt is a fun way to spend a birthday party with children.<br>• A scavenger hunt is a fun way to celebrate a birthday with friends.<br>**Default EXCEPTIONS:**<br>• Scavenger hunts can be done alone for a solo adventure.<br>• Some people may not find scavenger hunts enjoyable.<br>**Concept-Focused EXCEPTIONS:**<br>• A movie marathon is a fun way to spend an afternoon with friends.<br>• A picnic in the park is a fun way to spend an afternoon with friends.<br>• Playing board games is a fund way to spend an afternoon with friends.<br>**Property-Focused EXCEPTIONS:**<br>• A scavenger hunt can be a creative educational activity for students to learn about their surroundings.<br>• A scavenger hunt can be a team-building exercise for corporate events. |
| **(f)** | ***Generic:*** *Binoculars are used to view location.*<br>**Concept-Focused INSTANTIATIONS:**<br>• Birdwatchers use binoculars to view distant birds.<br>**Property-Focused INSTANTIATIONS:**<br>• Binoculars are used to view wildlife in national parks.<br>• Binoculars are used to view wildlife in their natural habitat.<br>**Default EXCEPTIONS:**<br>• Binoculars are used for stargazing.<br>• Binoculars are used for surveillance, wildlife observation, or marine activities.<br>**Concept-Focused EXCEPTIONS:**<br>• Telescopes are used to view celestial bodies.<br>• Microscopes are used to view tiny objects.<br>**Property-Focused EXCEPTIONS:**<br>• Binoculars can be use for stargazing or birdwatching. |
| | ***GGSmall Data*** |
| **(g)** | ***Generic:*** *A rose is placed in a container with water.*<br>**Concept-Focused INSTANTIATIONS:**<br>• A red rose is placed in a vase with water.<br>• A yellow rose is placed in a glass jar with water.<br>**Property-Focused INSTANTIATIONS:**<br>• A rose is placed in a vase with water.<br>**Default EXCEPTIONS:**<br>• A dried rose cannot be placed in a container with water.<br>• A fake rose made of plastic or silk cannot be placed in a container with water.<br>**Concept-Focused EXCEPTIONS:**<br>• A bouquet of flowers can be placed in a vase with water.<br>• A single flower can be placed in a bud vase with water.<br>**Property-Focused EXCEPTIONS:**<br>• A rose can be planted in soil to grow into a bush or a tree.<br>• A rose can be dried and preserved for decorative purposes. |
| **(h)** | ***Generic:*** *Cakes are made with a mix.*<br>**Concept-Focused INSTANTIATIONS:**<br>• Birthday cakes are made with a mix.<br>• Pancakes are made with a mix.<br>• Chocolate cakes are made with a mix.<br>**Property-Focused INSTANTIATIONS:**<br>• Cakes are made with a mix of flour, sugar, eggs, and other ingredients.<br>**Default EXCEPTIONS:**<br>• Some cakes are made from scratch.<br>**Property-Focused EXCEPTIONS:**<br>• Cakes can be made from scratch using individual ingredients.<br>**Concept-Focused EXCEPTIONS:**<br>• They can be made by professional bakers or homemade using recipes and various ingredients. |

Table 21: Examples of EXEMPLARS generated by the proof-of-concept baseline using GPT-3.5-Turbo (§B.4) for generics from *GGTest* and *GGSmall*.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




Sample outputs for each of the five types of EXEMPLARS are shown in Table 20 for *AnimalG* generics and Table 21 for generics from *GGTest* and *GGSmall*. We observe that while the this baseline is substantially more controllable than the GPT-3 baseline, it is still less controllable than our *ExempliFI* system. For one, in cases where only a portion of the generic should change (e.g., the concept in concept-focused EXCEPTIONS) this is not consistently the case. For example, for the generic "binoculars are used to view location" the concept-focused EXCEPTIONS include "telescopes are used to view celestial bodies" (see (f) in Table 21); here both the concept and property have been changed.

Additionally, this baseline fails to generate default EXCEPTIONS in most cases (more details below). Instead it generates candidates that are either focused EXCEPTIONS (e.g., (d) in Table 20 and (f) in Table 21) or alternatives for the generic as a whole (e.g., "some people may not find scavenger hunts enjoyable" for the generic "a scavenger hunt is a fun way to spend an afternoon with friends"; (e) Table 21). While these latter cases are interesting and worth investigating in future work, they are not valid EXCEPTIONS under our framework. Note that even when the model successfully generates default EXCEPTIONS (e.g., "penguins are birds than cannot fly") the EXCEPTIONS are primarily knowledge-based, not reasoning-based. Further exploration of the prompts used for the baseline may improve the default EXCEPTIONS. Overall, GPT-3.5-Turbo appears to be a reasonable baseline and should be further explored in future work.

## References

Allaway, Emily, Jena D. Hwang, Chandra Bhagavatula, Kathleen McKeown, Doug Downey, and Yejin Choi. 2023. Penguins don't fly: Reasoning about generics through instantiations and exceptions. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 2618–2635.

Allaway, Emily, Nina Taneja, Sarah-Jane Leslie, and Maarten Sap. 2022. Towards countering essentialism through social bias reasoning. In *Workshop on NLP for Positive Impact at EMNLP*.

Asher, Nicholas and Michael Morreau. 1995. What some generic sentences mean. *The generic book*, pages 300–338.

Bhagavatula, Chandra, Jena D. Hwang, Doug Downey, Ronan Le Bras, Ximing Lu, Keisuke Sakaguchi, Swabha Swayamdipta, Peter West, and Yejin Choi. 2022. I2d2: Inductive knowledge distillation with neurologic and self-imitation. In *Annual Meeting of the Association for Computational Linguistics*, pages 9614–9630.

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

Brewka, Gerhard. 1987. The logic of inheritance in frame systems. In *IJCAI*, pages 483–488.

Brown, Tom B, 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*.

Carlson, Greg N. 1977. *Reference to kinds in English*. Ph.D. thesis, University of Massachusetts, Amherst.

Carlson, Greg N. 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, pages 167–192.

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

Christiano, Paul F, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30:1–9.

Clark, Kevin, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *ICLR*, pages 1–18.

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

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

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

Collins, Allan and Ryszard Michalski. 1989. The logic of plausible reasoning: A core theory. *cognitive science*, 13(1):1–49.

Delgrande, James P. 1988. An approach to default reasoning based on a first-order conditional logic: Revised report. *Artificial intelligence*,

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




36(1):63–90.
Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
Ehrlich, Paul R., David S. Dobkin, and Darryl Wheye. 1988. How fast and high do birds fly? https://web.stanford.edu/group/stanfordbirds/text/essays/How_Fast.html.
Elio, Renée 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.
Fellbaum, Christiane D. 2000. Wordnet : an electronic lexical database. *Language*, 76:706.
Friedrich, Annemarie, Alexis Palmer, and Manfred Pinkal. 2016. Situation entity types: automatic classification of clause-level aspect. In *Annual Meeting of the Association for Computational Linguistics*, pages 1757–1768.
Friedrich, Annemarie, Alexis Palmer, Melissa Peate Sørensen, and Manfred Pinkal. 2015. Annotating genericity: a survey, a scheme, and a corpus. In *LAW@NAACL-HLT*, pages 21–30.
Friedrich, Annemarie and Manfred Pinkal. 2015. Discourse-sensitive automatic identification of generic expressions. In *Annual Meeting of the Association for Computational Linguistics*, pages 1272–1281.
Gelman, Susan A, Ingrid Sanchez Tapia, and Sarah-Jane Leslie. 2016. Memory for generic and quantified sentences in spanish-speaking children and adults. *Journal of Child Language*, 43(6):1231–1244.
Gilardi, Fabrizio, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences of the United States of America*, 120:3.
Ginsberg, M. 1987. Introduction. In M. Ginsberg, editor, *Readings in Nonmonotonic Reasoning*. Morgan Kaufmann, Los Altos, CA, page 481.
Govindarajan, Venkata S, Benjamin Van Durme, and Aaron Steven White. 2019. Decomposing generalization: Models of generic, habitual, and episodic statements. *Transactions of the Association for Computational Linguistics*, 7:501–517.
Greenberg, Yael. 2007. Exceptions to generics: Where vagueness, context dependence and modality interact. *Journal of Semantics*, 24(2):131–167.
Grice, Herbert P. 1975. Logic and conversation. In *Speech acts*. Brill, pages 41–58.
Grosz, Barbara, Aravind Joshi, and Scott Weinstein. 1983. Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st Annual meeting of the Association for Computational Linguistics*, pages 44–50, Association for Computational Linguistics.
Grosz, Barbara J, Aravind K Joshi, and Scott Weinstein. 1995. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.
Grosz, Barbara Jean. 1977. The representation and use of focus in dialogue understanding. *Technical Report 151*, page 186.
Hamblin, C. L. 1973. Questions in montague english. *Foundations of Language*, 10(1):41–53.
Hanks, Steve and Drew McDermott. 1986. Default reasoning, nonmonotonic logics, and the frame problem. In *AAAI*, pages 328–333.
Haward, Paul, Laura Wagner, Susan Carey, and Sandeep Prasada. 2018. The development of principled connections and kind representations. *Cognition*, 176:255–268.
Hoes, Emma, Sacha Altay, and Juan Bermeo. 2023. Leveraging chatgpt for efficient fact-checking. *PsyArXiv*.
Hollander, Michelle A, Susan A Gelman, and Jon Star. 2002. Children's interpretation of generic noun phrases. *Developmental psychology*, 38(6):883.
Horty, John F. and Richmond H. Thomason. 1988. Mixing strict and defeasible inheritance. In *AAAI*, pages 427–432.
Hu, Jennifer and Roger Levy. 2023. Prompting is not a substitute for probability measurements in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5040–5060.
Jackendoff, Ray S. 1972. *Semantic interpretation in generative grammar.* MIT Press.
Kadmon, Nirit. 2001. *Formal Pragmatics: Semantics, Pragmatics, Presupposition, and Focus*. Wiley-Blackwell.
Kadmon, Nirit and Fred Landman. 1993. Any. *Linguistics and philosophy*, 16(4):353–422.
Karczewski, Daniel, Edyta Wajda, and Radosław Poniat. 2020. Do all storks fly to africa? universal statements and the generic overgeneralization effect. *Lingua*, 246:102855.
Kassner, Nora and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Association for Computational Linguistics, Online.
Khemlani, Sangeet, Sarah-Jane Leslie, and Sam Glucksberg. 2008. Syllogistic reasoning with generic premises: The generic

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




overgeneralization effect. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 30, pages 1–6.

Khemlani, Sangeet, Sarah-Jane Leslie, and Sam Glucksberg. 2009. Generics, prevalence, and default inferences. *Proceedings of the 31st annual cognitive science society*, pages 443–448.

Khemlani, Sangeet S., Sarah-Jane Leslie, Sam Glucksberg, and Paula Rubio-Fernández. 2007. Do ducks lay eggs? how people interpret generic assertions. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 29, pages 1–6.

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

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

Krifka, Manfred. 2008. Basic notions of information structure. *Acta Linguistica Hungarica*, 55(3-4):243 – 276.

Krifka, Manfred, Francis Jeffry Pelletier, Gregory Carlson, Alice Ter Meulen, Gennaro Chierchia, and Godehard Link. 1995. Genericity: an introduction. In Gregory Carlson and Francis Jeffry Pelletier, editors, *The generic book*. The University of Chicago Press, pages 1–125.

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

Lampinen, Andrew Kyle. 2023. Can language models handle recursively nested grammatical structures. *arXiv preprint arXiv:2210.15303*.

Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Leshin, Rachel A, Sarah-Jane Leslie, and Marjorie Rhodes. 2021. Does it matter how we speak about social kinds? a large, preregistered, online experimental study of how language shapes the development of essentialist beliefs. *Child development*, 92(4):531–547.

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

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

Leslie, Sarah-Jane. 2014. Carving up the social world with generics. *Oxford studies in experimental philosophy*, 1:208–231.

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

Leslie, Sarah-Jane and Susan A Gelman. 2012. Quantified statements are recalled as generics: Evidence from preschool children and adults. *Cognitive psychology*, 64(3):186–214.

Leslie, Sarah-Jane, Sangeet Khemlani, and Sam Glucksberg. 2011. Do all ducks lay eggs? the generic overgeneralization effect. *Journal of Memory and Language*, 65(1):15–31.

Leslie, Sarah-Jane and Adam Lerner. 2016. Generic Generalizations. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Winter 2016 edition. Metaphysics Research Lab, Stanford University.

Lewis, David. 1975. Adverbs of quantification. In Edward L. Keenan, editor, *Formal Semantics of Natural Language*. Cambridge University Press, page 3–15.

Lifschitz, Vladimir. 1989. Benchmark problems for nonmonotonic reasoning. In *Proceedings of the Second international Workshop on Non-monotonic Reasoning*, pages 202–219.

Liu, Yinhan, 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*.

Lu, Ximing, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. Neurologic a* esque decoding: Constrained text generation with lookahead heuristics. In *NAACL*, pages 780–799.

MAAREC. 2011. A field guide to honey bees and their maladies. https://www.nj.gov/agriculture/divisions/pi/pdf/fieldguidetohoneybees.pdf. Accessed: 2023-10-14.

Mannheim, Bruce, Susan A Gelman, Carmen Escalante, Margarita Huayhua, and Rosalía Puma. 2010. A developmental analysis of generic nouns in southern peruvian quechua. *Language Learning and Development*, 7(1):1–23.

McCarthy, John. 1980. Circumscription—a form of non-monotonic reasoning. *Artificial intelligence*, 13(1-2):27–39.

McCarthy, John. 1986. Applications of circumscription to formalizing common-sense knowledge. *Artificial intelligence*, 28(1):89–116.

Meyer, Meredith, Susan A Gelman, and Sarah M Stilwell. 2011. Generics are a cognitive default: Evidence from sentence processing. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, pages 913–918.

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




Misra, Kanishka, Allyson Ettinger, and Kyle Mahowald. 2024. Experimental contexts can facilitate robust semantic property inference in language models, but inconsistently. *arXiv preprint arXiv:2401.06640*.

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

Mun, Jimin, Emily Allaway, Akhila Yerukola, Laura Vianna, Sarah-Jane Leslie, and Maarten Sap. 2023. Beyond denouncing hate: Strategies for countering implied biases and stereotypes in language. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 9759–9777.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*, pages 1–15.

Partee, Barbara. 1991. Topic, focus and quantification. In *Semantics and Linguistic Theory*, volume 1, pages 159–188.

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

Poesio, Massimo. 2004. Discourse annotation and semantic annotation in the GNOME corpus. In *Proceedings of the Workshop on Discourse Annotation*, pages 72–79, Association for Computational Linguistics, Barcelona, Spain.

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

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

Prasada, Sandeep 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.

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

Ralethe, Sello and Jan Buys. 2022. Generic overgeneralization in pre-trained language models. In *International Conference on Computational Linguistics*, pages 3187–3196.

Reiter, Nils and Anette Frank. 2010. Identifying generic noun phrases. In *Annual Meeting of the Association for Computational Linguistics*.

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

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

Rhodes, Marjorie, Sarah-Jane Leslie, and Christina M Tworek. 2012. Cultural transmission of social essentialism. *Proceedings of the National Academy of Sciences*, 109(34):13526–13531.

Roberts, Craige. 1996. Information structure in discourse: Towards an integrated formal theory of pragmatics. In J. Hak Yoon and A. Kathol, editors, *Ohio State University Working Papers in Linguistics*, volume 49. Ohio State University, pages 1–69.

Rohrbach, Anna, 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*, pages 4035–4045, Association for Computational Linguistics, Brussels, Belgium.

van Rooij, Robert and Katrin Schulz. 2019. A causal power semantics for generic sentences. *Topoi*, 40:131 – 146.

Rooth, Mats. 1992. A theory of focus interpretation. *Natural language semantics*, 1(1):75–116.

Sclar, Melanie, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*.

Sidner, Candace Lee. 1979. *Towards a computational theory of definite anaphora comprehension in English discourse*. Ph.D. thesis, Massachusetts Institute of Technology.

Speer, Robyn, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*, pages 4444–4451.

Sridharan, Mohan, Michael Gelfond, Shiqi Zhang, and Jeremy L. Wyatt. 2015. A refinement-based architecture for knowledge representation and reasoning in robotics. *ArXiv*, abs/1508.03891.

Subbiah, Melanie, Sean Zhang, Lydia B Chilton, and Kathleen McKeown. 2024. Reading subtext: Evaluating large language models on short story summarization with writers. *arXiv preprint arXiv:2403.01061*.

Suh, Sangweon. 2006. Extracting generic statements for the semantic web. *Master's*

*Computational Linguistics* Just Accepted MS.
https://doi.org/10.1162/coli_a_00530




*thesis, University of Edinburgh*.
Sutherland, Shelbie L, Andrei Cimpian, Sarah-Jane Leslie, and Susan A Gelman. 2015. Memory errors reveal a bias to spontaneously generalize to categories. *Cognitive science*, 39(5):1021–1046.
Talmor, Alon, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. *Advances in Neural Information Processing Systems*, 33:20227–20237.
Tardif, Twila, Susan A Gelman, Xiaolan Fu, and Liqi Zhu. 2012. Acquisition of generic noun phrases in chinese: Learning about lions without an '-s'. *Journal of child language*, 39(1):130–161.
Tessler, Michael Henry and Noah D Goodman. 2019. The language of generalization. *Psychological review*, 126(3):395.
Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
Vallduví, Enric and Elisabet Engdahl. 1996. The linguistic realization of information packaging. *Linguistics*, pages 459–519.
Veltman, Frank. 1996. Defaults in update semantics. *Journal of philosophical logic*, 25:221–261.
Von Fintel, Kai-Uwe. 1994. *Restrictions on quantifier domains*. Ph.D. dissertation, University of Massachusetts Amherst.
Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.