Evaluating Defeasible Reasoning in LLMs with DEFREASING

Anonymous ACL submission

Abstract

Defeasible inferences (i.e., inferences that are highly plausible but can be impacted by new information) are common in everyday life. We construct the DEFREASING dataset to evaluate defeasible reasoning about property inheritance (i.e., whether a subtype inherits a property from its parent type; Fig. 1). DEFREASING consists of $\sim 95k$ questions covering five patterns of reasoning and $\sim 8k$ inheritance rules. We use generics (i.e., generalizations without quantifiers) to represent the inheritance rules because their semantics includes exceptions. The semantics of generics, along with documented human reasoning behavior, is used to automatically construct the questions in DEFREASING. We evaluate 12 instruction-tuned LLMs on DEFREASING and find that not only does no model perform well across all pattern types, the best performing models only achieve ~ 0.64 overall $F1$. Further analysis highlights the challenges defeasible reasoning poses, as well as the inconsistencies in model performance depending on the type of reasoning involved and the availability of world-knowledge.

1 Introduction

In everyday life, a majority of the inferences we make (e.g., when making decisions or in argumentation) are *defeasible* (Chater et al., 2011). That is, they are highly plausible but are based on incomplete information and can therefore be impacted (e.g., defeated) by new information. Despite increasing studies into language model (LM) behavior (cf. Chang and Bergen, 2024), defeasible reasoning remains relatively understudied in LMs.

Defeasible inferences often arise from generalizations, which are readily captured in language by *generics* (i.e., generalizations without quantifiers). Generics can express general rules (e.g., "birds fly") while allowing for exceptions (e.g., "emus can't fly"). They also support inferences from only minimal evidence (Cimpian et al., 2010), making

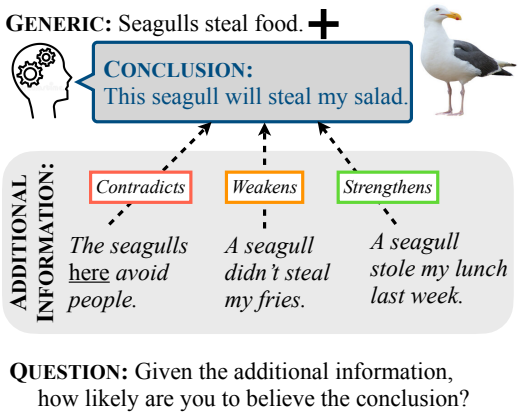

Figure 1: Overview of the defeasible reasoning task.

them a powerful inferential mechanism. Since generics occur frequently in texts (Herbelot and Copestake, 2011), and often occur unintentionally within existing reasoning datasets (e.g., Saparov and He, 2022), they are crucial to evaluating the defeasible reasoning abilities of LMs.

In this work, we construct a dataset DEFREASING (**Def**easible **Reas**oning about **In**heritance from **G**enerics) to evaluate defeasible reasoning in LMs through inferences drawn from generics. With defeasible inferences, new information can overturn a conclusion. However, new information can also strengthen or weaken our *confidence* in a conclusion *without changing* the conclusion itself (see Fig. 1). Generics are ideal for probing this latter type of impact, since their exceptions (i.e., examples where the generic doesn't apply) do not invalidate the generic itself (Allaway et al., 2023). Therefore, we use this semantics of generics (Allaway et al., 2024) to automatically construct instances that test whether models recognize the impact of new information on defeasible inferences.

Our DEFREASING dataset consists of $\sim 95k$ questions that probe defeasible reasoning. The questions concern property inheritance (i.e., whether an subtype inherits a property from its par-

ent type), since the concept-property rules used in inheritance can be naturally expressed with generics. We build the patterns and labels for our dataset using human studies on the relative strengthens of property arguments (e.g., Osherson et al., 1990). In total, DEFREASING includes 5 reasoning patterns for $\sim 8k$ property inheritance rules (generics). Variations of each pattern are also included to control for the effect world-knowledge (e.g., about the generic or its exceptions) has on performance.

We evaluate 12 instruction-tuned LLMs on DEFREASING. Despite models performing well on one type of reasoning, no model performs well across all types; the best performing models only achieve ~ 0.64 overall $F1$. We find that models generally perform better on examples where the additional information strengthens the conclusion, compared to examples that highlight the conclusion's defeasibility by weakening it. Models also struggle to recognize both diverse supporting evidence and irrelevant information. Finally, the design of DEFREASING facilitates analysis that shows that poorer performance often results from models relying on world-knowledge, rather than reasoning, for predictions. Our DEFREASING dataset highlights not only the difficulty of defeasible reasoning but also the inconsistencies in model performance depending on the type of reasoning involved and the availability of world knowledge.

Our contributions are: **(1)** we construct the first dataset to evaluate defeasible reasoning about property inheritance[1], **(2)** we leverage semantics of generics to automatically construct a large-scale dataset of $\sim 95k$ instances covering 5 patterns of reasoning, **(3)** we show that models struggle to perform consistently well across the patterns of reasoning in our dataset, highlighting the ongoing challenge of defeasible reasoning for LMs, and **(4)** the design of our dataset facilitates analysis into spurious factors affecting model performance, thus enabling further improvements in LMs.

2 Related Work

Defeasible and Nonmonotonic Reasoning Early work in AI on defeasible and nonmonotonic reasoning focused on formal logics (e.g., Reiter, 1978, 1980; Poole, 1988; Collins and Michalski, 1989) and recent NLP works have built upon these to construct datasets to evaluate nonmonotonic reasoning ability in LMs. Xiu et al. (2022) construct a dataset of proofs in nonmonotonic logic converted into natural language and Parmar et al. (2024) use patterns of nonmonotonic reasoning (Lifschitz, 1989) to build a QA dataset. Both of these works evaluate *what conclusions* can and cannot be drawn from premises. In contrast, our dataset tests whether models recognize the impact new information has on the *strength of an inference*, even when the conclusions are not changed.

Nonmonotonic reasoning has also been studied within the context of natural language inference (NLI) (Cooper et al., 1994; Yanaka et al., 2019b,a; Gubelmann et al., 2024). However, not only are instances of nonmonotonicity limited in these datasets but the task of NLI cannot account for additional premises. To address this, recent nonmonotonic reasoning tasks have been built on top of NLI (Bhagavatula et al., 2019; Rudinger et al., 2020). While we adopt a similar task formulation in this work, prior task datasets center around defeasible reasoning in social situations (Brahman et al., 2021; Ziems et al., 2023; Pyatkin et al., 2023; Rao et al., 2023), which conflates reasoning about commonsense knowledge and defeasibility[2]. In contrast, our work removes the influence of commonsense knowledge by focusing only on property inheritance inferences.

Property Inference Recent works have investigated LMs' reasoning about inheritance over implicit property (Misra et al., 2022, 2023, 2024) and taxonomic (Talmor et al., 2020) knowledge. However, the implicit knowledge makes it difficult to disentangle model ignorance and reasoning ability. Additionally, although property inheritance has been used to evaluate deductive reasoning over natural language rules (Saparov and He, 2022; Tafjord et al., 2021; Tian et al., 2021; Clark et al., 2021), these studies make use of if-then rules (e.g., if A is a cat then A purrs) that are more akin to first-order logic than to how humans often express rules in regular language. (i.e., through generalizations – "cats purr"). In contrast, our work studies property inheritance from generics (a specific type of linguistic generalization) and explicit knowledge.

Generics and Reasoning Resources for generics include methods to identify them (Friedrich and Pinkal, 2015; Friedrich et al., 2015, 2016;

[1] Our data will be made publicly available

[2] "Rob is late for work because he missed the bus" may be weakened by "Rob rides a bike to work" but not by "Rob rides a tricycle to work" because commonsense tells us that adults don't ride tricycles and children (who do) don't go to work.

Govindarajan et al., 2019) and datasets of both generics (Bhakthavatsalam et al., 2020; Bhagavatula et al., 2022) and their exemplars (i.e., cases where the generic does and does not hold) (Allaway et al., 2023). Additionally, studies have probed how LMs model the semantics of generics in relation to quantification (Ralethe and Buys, 2022; Collacciani et al., 2024). Recent studies have shown that LMs' reasoning about generics exemplars appears to be somewhat nonmonotonic (Allaway et al., 2024; Leidinger et al., 2024). That is, LMs often treat exemplars as not impacting their belief in the generic itself. In contrast to these works, we do not study quantification or the endorsement of a generic itself; rather, we focus on how reasoning operates when *generics are used* as inference rules.

Closely related to our work is the recent exploration from Allaway et al. (2024) of whether exemplars cause LMs to modify their behavior about property inheritance. They find that, similar to human pragmatic reasoning (e.g., Grice, 1975), LMs' behavior with generics does not align with general nonmonotonic logic patterns proposed in the AI literature (i.e., Lifschitz, 1989). Therefore, in this work we use more nuanced patterns of defeasible reasoning, proposed around generics (Asher and Morreau, 1990), to evaluate property inheritance.

3 DEFREASING Dataset

A defeasible inference is a plausible conclusion drawn from a set of premises. These inferences can be strengthened or weakened by new information. We construct DEFREASING to evaluate the capability of models to recognize and reason about defeasible inferences related to property inheritance. That is, the instances query whether a subtype (sparrows) of some concept (birds) inherits a property from the concept (can fly).

In the following, we first formally define the task and format for DEFREASING (§3.1). We then define the premises (§3.2) and additional information (§3.3) used to evaluate different patterns of reasoning. Finally, we describe how we use semantics of generics (Allaway et al., 2024) to automatically construct the questions in DEFREASING (§3.4).

3.1 Task Definition

We formulate defeasible reasoning as a task adjacent to NLI, following Rudinger et al. (2020). Let $\mathcal{H}$ be a conclusion that is *entailed* by an initial set of premises $\mathcal{P}^i$. Then, given an additional set of premises $\mathcal{P}^x$, the task is to predict how $\mathcal{P}^x$ impacts the entailment relation: does $\mathcal{P}^x$ strengthen, weaken, or not impact that relation. For simplicity, we will refer to this impact on the entailment relation as an *impact on the conclusion*[3]. Therefore, each instance is a tuple $\{\mathcal{P}^i, \mathcal{P}^x, \mathcal{H}, \Delta\}$ where Δ is the impact of $\mathcal{P}^x$ on $\mathcal{H}$.

The instances in DEFREASING all center around a pattern of syllogistic reasoning. Specifically, given a rule about a concept K having a property A (e.g., birds have wings), it is inferred that a subtype of that concept (e.g., sparrow) also has the property. In DEFREASING, the rule is specified as a generic, making the conclusion defeasible. This is because the generic is a generalization and leaves room for unspecified exceptions. In each inference question, the initial premises $\mathcal{P}^i$ and conclusion $\mathcal{H}$ adhere to this reasoning pattern and the additional premises $\mathcal{P}^x$ are judged in relation to it.

In all instances, the initial premises $\mathcal{P}^i$ consist of statements indicating whether one or more concepts (e.g., birds, sparrows) possess a specified property A (e.g., have wings). Then, the conclusion $\mathcal{H}$ is a single statement indicating that $K^{\mathcal{H}}$, which is a subtype of K, also possess A. Finally, the additional premises $\mathcal{P}^x$ provide more information about whether or not concepts, either those already in $\mathcal{P}^i$ or new ones, have property A.

3.2 Initial Premises $\mathcal{P}^i$ and Conclusion $\mathcal{H}$

The initial premises $\mathcal{P}^i$ consist of statements indicating whether one or more concepts (e.g., birds, sparrows) possess a specified property A (e.g., can fly). Specifically, $\mathcal{P}^i$ contains two elements. The first is a rule (e.g., birds can fly) that specifies a base concept K (birds) and a property A (have wings) that K possesses. Each rule is provided as a generic. The second element in $\mathcal{P}^i$ is a set of statements specifying the taxonomic relationship between the concept K and one or more subtypes of K (e.g., sparrows are birds).

The instances in DEFREASING consist of either single or 2-step inheritance. For the single-step inheritance instances, the initial premise set $\mathcal{P}^i_1$ consists of two statements:

$$\mathcal{P}^i_1 = \{K \text{ have } A, \quad K^{\mathcal{H}} \text{ are } K\}$$

where the first premise is the generic expressing a rule and the second premise makes explicit the

[3]In reality, it is not the conclusion itself that is impacted but rather the belief in the conclusion.

taxonomic relationship between K and $K^{\mathcal{H}}$. The second premise is necessary because $K^{\mathcal{H}}$ is not a real concept.

For the 2-step instances, we add an intermediate type between K and $K^{\mathcal{H}}$ so the initial premises are

$$\mathcal{P}_2^i = \{K \text{ have } A, C \text{ are } K, K^{\mathcal{H}} \text{ are } C\}$$

where C is a subtype of K. Depending on whether or not the subtype has property A, C is either K^+ or K^- respectively (see §3.3). For example, for K="birds" and property A="have wings", C can either be a subtype with wings (e.g., "sparrows") or without wings (e.g., "Kiwi birds").

The conclusion $\mathcal{H}$ is the same across all instances. In particular, the conclusion has the form

$$\mathcal{H} = K^{\mathcal{H}} \text{ have } A.$$

Note that in all cases $K^{\mathcal{H}}$ is a nonsense type. This prevents the model from using world knowledge about $K^{\mathcal{H}}$ in its reasoning.

3.3 Additional Premises $\mathcal{P}^x$

We construct five categories of additional premises $\mathcal{P}^x$ which we group by the impact they have on the the conclusion $\mathcal{H}$: strengthening (§3.3.1), weakening (§3.3.2), or no impact (§3.3.3). Note, we will use K="birds" and A="have wings" as an illustrative example (additional examples in Table 1).

3.3.1 Strengthening: *S-case* and *S-alt*

Given initial premises $\mathcal{P}^i$, the conclusion can be strengthened with two kinds of examples. Firstly, we have subtypes of K that have property A. These strengthen the conclusion by providing confirmation of the property among subtypes of K (***S-case***). In the case of single-step inheritance, these subtypes provide indirect support for inheritance of the property to $K^{\mathcal{H}}$. We construct the additional premises for these examples

$$\mathcal{P}_{1+}^x = \{K^+ \text{ are } K, K^+ \text{ have } A\}.$$

Alternatively, for 2-step inheritance, the additional premises directly confirm the presence of the property in the intermediate type. Recall that first statement in $\mathcal{P}_{1+}^x$ is part of the initial premises $\mathcal{P}_2^i$ in this case and $K^{\mathcal{H}}$ is a subtype of K^+, rather than K The additional premises are then

$$\mathcal{P}_{2+}^x = \{K^+ \text{ have } A\}.$$

In other words, $\mathcal{P}^x$ strengthens the conclusion by ensuring that the property *again* only needs to be inherited one step (i.e., from K^+ to $K^{\mathcal{H}}$).

The second kind of strengthening example is based on the phenomena that arguments with more diverse evidence are stronger (Osherson et al., 1990). Therefore, these examples use concepts *other than* K that have property A (***S-alt***). For our example K and A, we might choose K^+="bats" since bats are not birds but do have wings. We only construct these for single-step inheritance and the premises have the form

$$\mathcal{P}_{1\oplus}^x = \{K^{\oplus} \text{ are not } K, K^{\oplus} \text{ have } A\}$$

where $K^{\oplus}$ is the alternate concept to K.

3.3.2 Weakening: *W-case*

Weakening instances follow a similar pattern as described above. That is, the conclusion is weakened by providing examples of K without the property A (***W-case***). This nonmonotonicity arises from the fact that in reasoning with generics, information about subtypes takes precedence over information about the super type (Asher and Morreau, 1990). Therefore, these counterexamples provide evidence that the information about K possessing A may not always apply.

For single-step inheritance, the additional premises that weaken the conclusion have the form

$$\mathcal{P}_{1-}^x = \{K^- \text{ are } K, K^- \text{ do not have } A\}.$$

Additionally, we construct 2-step instances with

$$\mathcal{P}_{2-}^x = \{K^- \text{ do not have } A\}.$$

Note that $\mathcal{P}_{2-}^x$ *directly* contradicts the conclusion by indicating that the narrowest mentioned supertype of $K^{\mathcal{H}}$ (here K^-) does *not* have property A.

3.3.3 No Impact: *N-alt* and *N-prop*

Finally, we construct instances where the additional premises $\mathcal{P}^x$ have no impact on the conclusion. Such instances contain information that is irrelevant to the conclusion. Firstly, we use as irrelevant information concepts other than K that *do not have* property A (e.g., "cats" for our illustrative example) (***N-alt***). The additional premises for these are

$$\mathcal{P}_{1\ominus}^x = \{K^{\ominus} \text{ are not } K, K^{\ominus} \text{ do not have } A\}.$$

Secondly, irrelevant information can be other properties that K possess (e.g., "have beaks" for K ="birds") (***N-prop***). These have the form

$$\mathcal{P}_{1\diamond}^x = \{K \text{ have } A^{\diamond}\}$$

where $A^{\diamond}$ is a property other than A that K possess.

Initial Premises $\mathcal{P}^i$	Additional Premises $\mathcal{P}^x$	Name
$\mathcal{P}^i_1$: Cats sleep in trees. A Wumox is a cat.	$\mathcal{P}^x_{1+}$: Leopards are cats. Leopards sleep in trees.	*S-case*
	$\mathcal{P}^x_{1\oplus}$: Koalas are not cats. Koalas sleep in trees.	*S-alt*
	$\mathcal{P}^x_{1-}$: Cheetahs are cats. Cheetahs do not sleep in trees.	*W-case*
	$\mathcal{P}^x_{1\ominus}$: Dogs are not cats. Dogs do not sleep in trees.	*N-alt*
	$\mathcal{P}^x_{1\diamond}$: Cats sleep in beds.	*N-prop*
$\mathcal{P}^i_2$: Cats sleep in trees. Leopards are cats. A Wumox is a leopard.	$\mathcal{P}^x_{2+}$: Leopards sleep in trees.	*S-case*
$\mathcal{P}^i_2$: Cats sleep in trees. Cheetahs are cats. A Wumox is a cheetah.	$\mathcal{P}^x_{2-}$: Cheetahs do not sleep in trees.	*W-case*
$\mathcal{H}$: Wumoxes sleep in trees.		

Table 1: Examples of the initial ($\mathcal{P}^i$) and additional ($\mathcal{P}^x$) premises for the patterns in DEFREASING. Examples are based on the generic "cats sleep in trees". The hypothesis $\mathcal{H}$ is the same for all tuples $\langle \mathcal{P}^i, \mathcal{P}^x, \mathcal{H}, \cdot \rangle$ represented.

3.4 Dataset Construction

We construct the **5 categories (*S-case*, *S-alt*, *W-case*, *N-alt*, *N-prop*)** of instances in DEFREASING automatically. This is possible because we use generics to specify the rule in each instance, allowing us to exploit the semantic relationships between generics and their exemplars (i.e., examples where the generic does and does not hold) to obtain the subtypes necessary for each category of additional premises (§3.3). Specifically, we know that there exist not only subtypes where the generic holds but also subtypes where it does not. We extract and use these subtypes to construct the examples in our dataset. We use the semantic relationships between generics and exemplars defined by Allaway et al. (2024) to identify appropriate subtypes ($K^+, K^-, K^\oplus, K^\ominus$) and relevant properties ($A^\diamond$) for the additional premises (§3.3). For a more detailed discussion of the semantics from Allaway et al. (2024), see Appendix B.

3.4.1 Type Extraction

Subtypes: K^+ and K^- For both *S-case* and *W-case* examples we need subtypes of the base concept K. In particular, for strengthening instances we need subtypes K^+ that also have the property A. Since our base rule is expressed by a generic, these supporting subtypes will occur in instantiations of the generic. In contrast, the weakening instances require subtypes K^- that do not have the property A; these will occur in *exceptions* to the generic[4]. In 2-step inheritance, we use the same subtypes for the corresponding single-step instances.

Alternates: $K^\oplus, K^\ominus$, and $A^\diamond$ For *S-alt* instances, we need $K^\oplus$ that are concepts distinct from K that also have property A. For a generic, the concept-alternate examples from Allaway et al. (2024) have this specific type of concept. Since these exemplars also require that their alternate concepts are relevant to the generic (e.g., "airplanes have wings" is not relevant to "birds have wings" but "bats have wings" is), we can use these to obtain $K^\oplus$ that strengthen inheritance conclusions.

Similar to the concept alternates, Allaway et al. (2024) define property-alternate exemplars that specify other properties possessed by the concept K. We use these property-alternate exemplars to obtain $A^\diamond$ for the *N-prop* examples.

Finally, $K^\ominus$ in *N-alt* examples should be a concept that *does not* have property A. However, statements with such $K^\ominus$ are not relevant to the generic "K have A" and therefore cannot be obtained from exemplars (e.g., "cats don't have wings" is not an exemplar for "birds have wings"). Furthermore, we still want $K^\ominus$ to be related to K so that it is not obviously irrelevant (e.g., "cats" are related to "birds" whereas "boats" are not). Therefore, to obtain $K^\ominus$, we first categorize the concept in a generic (e.g., "bird" has the category "animal"). Then, we randomly select a related second category (e.g., "fish") from which we sample candidate $K^\ominus$. Finally, to ensure that $K^\ominus$ does not have property A, we pass the candidates through a truth filter.

3.4.2 Real vs. Nonsense Type Variations

In order to account for the impact of world-knowledge about the various types included in the additional premises, we construct two kinds of additional premises $\mathcal{P}^x$ for each of the five categories described in §3.3. One kind uses a real type while the other uses a nonsense type. For example, for $\mathcal{P}^x_{1+}$ we construct instances using a real subtype of K for K^+ (e.g., K^+=sparrows for K=bird) and

[4]For clarity, we use our own names for exemplar types. See Appendix B for the mapping to Allaway et al. (2024).

using a nonsense type for K^+ (e.g., a Wumox). Note that because the $\mathcal{P}^x$ for *N-prop* examples do not contain a subtype or alternate type to K (see §3.3), we do not create two kinds of instances for this category. Overall, DEFREASING includes **13 kinds of instances** with real and nonsense types for 4 of the 5 categories, 2 of which also have single and 2-step inheritance questions.

3.4.3 Implementation Details

As a source of generics and exemplars we use a subset of 8726 generics and accompanying exemplars (*AnimalG-Ex*) from Allaway et al. (2024). Specifically, we use generics concerning animals which have valid exceptions. For each generic, we take the top ranked exemplar for each relevant type (§3.4.1) in constructing our examples. Namely, we use the top-ranked concept instantiation for K^+, exception for K^-, concept-alternate exemplar for $K^\oplus$, and property-alternate exemplar for $A^\diamond$.

To obtain $K^\ominus$, we use the categories and property annotations from the XCSLB dataset (Devereux et al., 2014; Misra et al., 2022), a dataset of human-annotated property norms for 521 concepts. Specifically, we categorize each concept K and randomly select a second category to sample $K^\ominus$ from. Then, we compute the similarity between the concept K and each concept in the second category and sample the five most similar concepts from this second category as candidate $K^\ominus$. We pass the resulting property statements (i.e., "$K^\ominus$ do not have A") to a truth filter and the final $K^\ominus$ is the most similar candidate that the truth filter validates does *not* have property A.

For a pair of concepts, we follow Misra et al. (2023) and measure similarity as the Jaccard index between two feature vectors that represent the properties each concept in the pair does and does not have. The properties considered are those in XCSLB. The feature values combine the XCSLB annotations with concept-property information from the generics in *AnimalG-Ex*. Before sampling candidate $K^\ominus$, we remove any examples from the secondary category that are marked as *having* the property A in question. Following Allaway et al. (2024) we use GPT-3.5 for the filter. See Appendix A for full data construction details.

3.4.4 Dataset Statistics

DEFREASING contains 94671 instances covering 8726 generics (see Table 1 for examples). Note that due to the small size of XCSLB, we only obtained viable $K^\ominus$, and *N-alt* examples, for 200 generics (we denote this subset **DEFREASING∗*N-alt***).

4 Experiments

We describe the models (§4.1) and prompting set-up (§4.2) used to evaluate models on DEFREASING (full implementation details in Appendix C).

4.1 Models

We evaluate 12 instruction-tuned LLMs on our DEFREASING dataset. The models include LLMs both from the top of evaluation leaderboards[5] and from prior studies on nonmonotonic reasoning (Leidinger et al., 2024). We use 9 open-source models: *Mistral* (Jiang et al., 2023), *Mixtral* (Jiang et al., 2024), *Hermes* (NousResearch, 2023), *Starling* (Zhu et al.), *Zephyr* (Tunstall et al., 2023) *Llama2* (Touvron et al., 2023), *Llama3* (Meta, 2024b), *Llama3.1* (Meta, 2024a), and *Wizard* (Xu et al., 2023). We also evaluate *GPT-3.5* (Ouyang et al., 2022), *GPT-4* (Achiam et al., 2023), and *GPT-4o* (OpenAi, 2024).

4.2 Prompt Format

We format the examples in DEFREASING using a chat set-up which includes a system prompt (see Fig. 1). Although the exact format of the chat template is dependent on the model (see Appendix C), the instruction wording is the same across models. For models that do not support a system prompt, we append the system instructions to the beginning of the user input. To convert the model output into labels, we check whether the response contains the label phrases from the system prompt (e.g., "more likely" or "less likely").

Since LLM behavior can vary depending on the wording in the prompt (Webson and Pavlick, 2022; Leidinger et al., 2023), we experiment with three different prompts. The first uses more natural and colloquial language (see Table 2). The third uses Chain-of-Thought (Wei et al., 2022) prompting by adding "Lets think step by step" to the first type of prompt. Our results are averaged across prompts (see Appendix D for per-prompt results).

5 Results and Analysis

We report the overall accuracy and macro-averaged F1, along with the accuracy for each question, for each model on DEFREASING in Table 3. Note

[5] AlpacaEval and ChatBot Arena.

Please answer with only "more likely", "less likely", or "it has no impact".

Consider the following information: [$\mathcal{P}^i$]
From this information we can draw a conclusion about [$K^{\mathcal{H}}$].
Conclusion: [$\mathcal{H}$].
Now suppose we are given additional information.
Additional information: [$\mathcal{P}^x$]
Given the additional information,
how likely are you to believe the conclusion?

Table 2: Example prompt format. The system instruction is in *italics* above the dashed line.

that due to computational costs we only evaluate *GPT-3.5/4/4o* on DEFREASING∗*N-alt*.

Overall, the best performing models only achieve ∼0.64 F1 across all question categories, leaving substantial room for improvement. Furthermore, there is large variation in the performance per category, suggesting that models may correctly recognize only certain types of reasoning. We discuss the results in more detail below.

We observe that while models may perform very well on one type of example (e.g., identifying strengthening evidence in *S-case* examples), **no model performs well across all categories of examples**. For example, while *Starling* and *Llama3* both perform close to perfect on *S-case* instances, both models perform very poorly (close to 0 $F1$) at identifying information that does not impact the conclusion (*N-alt* and *N-prop* instances).

Generally, the best performing models are those that exhibit the least variation in performance across categories. In particular, *Mixtral* and *Zephyr* have the highest overall $F1$ scores, although their performance is not the highest for any individual question type. It should be noted that low cross-category variation can also be due to consistently low performance (e.g., *Llama2*); likewise, high overall $F1$ can result from near perfect performance on several types (e.g., *Llama3*). However, the latter indicates that in order for models to achieve better results on DEFREASING they must demonstrate some grasp of each of the three ways (strengthen, weaken, no impact) that new evidence can affect reasoning.

Monotonicitiy is easier than nonmonotonicity Model performance is generally higher on the strengthening evidence examples (*S-case* and *S-alt*), which have a monotonic impact on the conclusion, than on the weakening evidence examples (*W-case*), which have a *non*monotonic impact[6]. Note that *Llama3.1* (in addition to *Zephyr*, *GPT-4/4o*) is an exception to this; in fact, it achieves the *highest* performance on the weakening examples while having some of the *lowest* performance on strengthening examples, emphasizing the large variations in performance across categories.

Models struggle to recognize diverse support Models generally perform much better on the *S-case* examples compared to the *S-alt* examples. Recall that the *S-case* examples include a subtype of the concept where the property *is inherited* while the *S-alt* examples include an *alternate* concept that has the property. These latter cases support the conclusion by providing *diverse* supporting examples. While a preference for diverse property inheritance arguments has been documented in humans (Osherson et al., 1990), shallow heuristics may be impacting how models perform on *S-alt* examples.

For example, recall that both *S-case* and *S-alt* examples explicitly state the taxonomic link (or lack thereof) to the concept. For the *S-alt* examples, this link includes a negation (e.g., "Cows are not sheep"). We observe a statistically significant positive correlation (Pearson's r)[7] for most models between the presence of "not" in the additional premises and more predictions of the "weakens" label. We note that although inclusion of the taxonomic statement may impact the performance on these cases negation cannot be simply avoided when studying LLMs; this is especially true in defeasible reasoning, where weakening evidence often explicitly contradicts a premise using negation. Therefore, the performance on these alternate cases serves both to further document the struggles of LMs at handling negation (cf. Chang and Bergen, 2024) and to underscore the limitations in LM ability that arise as a result.

Irrelevant evidence is not irrelevant It has been previously documented that models struggle to handle irrelevant information (Shi et al., 2023). Our results show that, for most models, this is also the case in reasoning about property inheritance. That is, model performance on the "no impact" (*N-alt* and *N-prop*) examples is substantially lower than performance on the strengthening and weak-

[6]Only the 2-step *W-case*'s formally result in a nonmonotonic inference (i.e., the conclusion should be withdrawn); the single-step *W-case*'s indicate a *potential* nonmonotonicity.

[7]Using a two-sided t-test with $p < 0.0001$.

	S-case		*S-alt*	*W-case*		*N-alt*	*N-prop*	Overall	
	1step	2step		1step	2step			Acc.	$F1$
Mistral	0.8665	0.9317	0.03752	0.2758	0.4929	**0.9883**	**0.7943**	0.5501	0.4940
Mixtral	0.9523	0.9547	0.2577	0.5092	0.6890	0.6642	0.5000	0.6788	0.5912
Starling	**0.9987**	**0.9990**	0.4904	0.6663	0.6743	0.0001	0.0420	0.7045	0.4687
Zephyr	0.7168	0.7468	0.1130	0.8658	0.9887	0.4875	0.4993	0.6737	0.5894
Hermes	0.9873	0.9913	0.5880	0.9712	0.9826	0.0	0.0110	0.8327	0.5885
Llama2	0.6291	0.5998	0.4643	0.2317	0.4102	0.3475	0.3509	0.4575	0.3216
Llama3	0.9986	0.9986	**0.9357**	0.9423	0.9997	0.0001	0.0210	**0.8904**	**0.6353**
Llama3.1	0.6280	0.6573	0.3883	**0.9886**	**0.9998**	0.0617	0.0420	0.6747	0.4678
Wizard	0.8063	0.6927	0.3322	0.7950	0.8892	0.0808	0.1667	0.6589	0.5256
*GPT-3.5**	0.9908	0.9867	0.6299	0.8342	0.9517	0.1992	0.1296	0.7207	0.6023
*GPT-4**	0.0100	0.1308	0.0	0.7750	0.9725	1.0	0.9983	0.4415	0.4214
*GPT-4o**	0.0075	0.1083	0.0	0.9858	0.9942	1.0	0.9427	0.4135	0.3959

Table 3: Accuracy on DEFREASING across reasoning types (§3.3) and macro-$F1$. Results are averaged across prompts (§4.2) and across real and nonsense types (§3.4.2). Best model is **bolded**, worst model is underlined. * indicates the evaluation is done only on DEFREASING∗*N-alt*. *1step* and *2step* indicate single and 2-step inheritance.

	S-case		*S-alt*	*W-case*		*N-alt*
	1step	2step		1step	2step	
Mistral	+.049	-.009	+.042	-.111	-.084	+.007
Mixtral	-.052	-.028	-.108	-.018	+.274	+.202
Starling	-.003	+.001	-.004	<.001	+.011	+.002
Zephyr	+.162	-.019	+.025	-.137	-.014	-.068
Hermes	-.003	+.001	-.055	-.021	-.018	0.0
Llama2	-.024	+.020	-.009	-.004	-.009	-.005
Llama3	-.003	<.001	-.031	+.045	<.001	-.002
Llama3.1	+.011	-.011	+.119	+.011	<.001	+.027
Wizard	+.017	+.145	+.105	-.088	+.005	+.015
GPT-3.5	+.005	-.003	+.045	-.165	-.020	+.052
GPT-4	+.020	+.162	0.0	-.343	-.005	0.0
GPT-4o	+.008	+.067	0.0	-.028	+.005	0.0
AVG	+.016	+.027	+.011	-.081	+.012	+.019

Table 4: Difference between accuracy on real and nonsense types (§3.4.2) for each category. *1step* and *2step* indicate single and 2-step inheritance respectively.

ening examples. We note that although both *GPT-4* and *GPT-4o* perform at or close to 100% accuracy on the "no impact" examples, they have a strong bias towards predicting "no impact" (predicted on $\sim$60% of examples). This leads to poor performance for *S-case*, *S-alt*, and *W-case* examples. Additionally, although *Mistral* performs relatively well on the *N-alt* and *N-prop* examples, its performance on *W-case* and *S-alt* instances is very poor. Therefore, models that appear to recognize irrelevant information have substantial limitations in their general defeasible reasoning ability.

Reliance on World-Knowledge Spurious associations are well documented in models for reasoning in natural language (Poliak et al., 2018; McKenna et al., 2023). Therefore, DEFREASING has examples using both real and nonsense types in the additional premises (§3.4.2). In this way we can observe whether predictions are impacted by world-knowledge about the real types.

We observe (Table 4) that generally the performance differences between the real and nonsense types are quite small. However, some models do exhibit discrepancies that indicate that they may be relying on world-knowledge to make predictions. For example, *Zephyr*'s performance on *S-case* instances drops $\sim$16 points with the nonsense types and for *W-case* instances it increases $\sim$14 points with the nonsense types. Additionally, we observe that models with mediocre performance (e.g., *Mistral*) are more sensitivity to the nonsense types. This highlights the importance of considering the impact of world knowledge when evaluating defeasible reasoning.

6 Conclusion

We present the DEFREASING dataset to evaluate defeasible reasoning in LLMs. It consists of $\sim 95k$ questions that probe how models reason about property inheritance from generics and covers five patterns of reasoning for $\sim 8k$ inheritance rules. The questions in DEFREASING are constructed automatically based on documented human property-reasoning behavior and semantics of generics. We evaluate 12 instruction-tuned LLMs on DEFREASING and find that no model performs well across all reasoning types. Our analysis highlights inconsistencies in performance depending on the type of reasoning and the availability of world-knowledge. These results indicate that there is more work to be done in order for LMs to handle defeasible reasoning. We hope that DEFREASING will stimulate progress in this area.

7 Limitations

Due to computational limitations, we run *Mixtral* quantized. Note that there is evidence that quantization does not substantially impact *Mistral* models' performance on numerous tasks (Badshah and Sajjad, 2024).

Additionally, the dataset of generics and exemplars from Allaway et al. (2024) used to create DEFREASING is synthetic. Although Allaway et al. (2024) conduct human evaluation on a portion, and find very high precision, they also note that it is likely that the generics appeared in some way in the training data of LMs. Since our DEFREASING dataset uses this data, it is likely that the training data for LMs includes the generics (i.e., inheritance rules) included in DEFREASING. However, the actual reasoning questions in DEFREASING have been newly constructed and should therefore not be part of any LM training dataset. Note also that our dataset only contains English examples.

We construct the labels for DEFREASING based on documented behavior in how humans reason about property inheritance arguments. Many studies have been done on such behavior and there are numerous factors that affect behavior. For example, whether or not a property is anatomical or behavioral can impact the strength of inferences (Heit and Rubinstein, 1994). Further work should be done to examine the impact that these factors have on the reasoning examples in our dataset. We also note that human studies were done with English-speaking participants and it is possible that biases (e.g., cultural or linguistic) may have impacted their results.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Emily Allaway, Chandra Bhagavatula, Jena D. Hwang, Kathleen McKeown, and Sarah-Jane Leslie. 2024. Exceptions, Instantiations, and Overgeneralization: Insights into How Language Models Process Generics. *Computational Linguistics*, pages 1–60.

Emily Allaway, 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 *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2618–2635, Dubrovnik, Croatia. Association for Computational Linguistics.

Nicholas Asher and Michael Morreau. 1990. Commonsense entailment: A modal theory of nonmonotonic reasoning. In *European Workshop on Logics in Artificial Intelligence*, pages 1–30. Springer.

Sher Badshah and Hassan Sajjad. 2024. Quantifying the capabilities of llms across scale and precision. *arXiv preprint arXiv:2405.03146*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. *arXiv preprint arXiv:1908.05739*.

Chandra Bhagavatula, 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. *ArXiv*, abs/2212.09246.

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

Faeze Brahman, Vered Shwartz, Rachel Rudinger, and Yejin Choi. 2021. Learning to rationalize for nonmonotonic reasoning with distant supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12592–12601.

Tyler A Chang and Benjamin K Bergen. 2024. Language model behavior: A comprehensive survey. *Computational Linguistics*, 50(1):293–350.

Nick Chater, Mike Oaksford, Ulrike Hahn, and Evan Heit. 2011. Inductive logic and empirical psychology. In *Handbook of the History of Logic*, volume 10, pages 553–624. Elsevier.

Andrei Cimpian, Amanda C Brandone, and Susan A Gelman. 2010. Generic statements require little evidence for acceptance but have powerful implications. *Cognitive science*, 34(8):1452–1482.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Claudia Collacciani, Giulia Rambelli, and Marianna Bolognesi. 2024. Quantifying generalizations: Exploring the divide between human and LLMs' sensitivity to quantification. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11811–11822, Bangkok, Thailand. Association for Computational Linguistics.

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

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

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

Annemarie Friedrich, 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.

Annemarie Friedrich, 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.

Annemarie Friedrich and Manfred Pinkal. 2015. Discourse-sensitive automatic identification of generic expressions. In *Annual Meeting of the Association for Computational Linguistics*, pages 1272–1281.

Venkata S Govindarajan, 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.

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Reto Gubelmann, Ioannis Katis, Christina Niklaus, and Siegfried Handschuh. 2024. Capturing the varieties of natural language inference: A systematic survey of existing datasets and two novel benchmarks. *Journal of Logic, Language and Information*, 33(1):21–48.

Evan Heit and Joshua Rubinstein. 1994. Similarity and property effects in inductive reasoning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20(2):411.

Aurelie Herbelot and Ann Copestake. 2011. Formalising and specifying underquantification. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Alina Leidinger, Robert van Rooij, and Ekaterina Shutova. 2023. The language of prompting: What linguistic properties make a prompt successful? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9210–9232, Singapore. Association for Computational Linguistics.

Alina Leidinger, Robert Van Rooij, and Ekaterina Shutova. 2024. Are LLMs classical or nonmonotonic reasoners? lessons from generics. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 558–573, Bangkok, Thailand. Association for Computational Linguistics.

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

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2758–2774, Singapore. Association for Computational Linguistics.

Meta. 2024a. Introducing llama 3.1: Our most capable models to date. Blog Post.

Meta. 2024b. Introducing meta llama 3: The most capable openly available llm to date. Blog Post.

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

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics.

Kanishka Misra, Julia Taylor Rayz, and Allyson Ettinger. 2022. A property induction framework for neural language models. In *Proceedings of the 44th Annual Conference of the Cognitive Science Society*.

NousResearch. 2023. Openhermes 2.5 - mistral 7b.

OpenAi. 2024. Hello gpt-4o. Blog Post.

Daniel N Osherson, Edward E Smith, Ormond Wilkie, Alejandro Lopez, and Eldar Shafir. 1990. Category-based induction. *Psychological review*, 97(2):185.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. LogicBench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13679–13707, Bangkok, Thailand. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

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

Valentina Pyatkin, Jena D. Hwang, Vivek Srikumar, Ximing Lu, Liwei Jiang, Yejin Choi, and Chandra Bhagavatula. 2023. ClarifyDelphi: Reinforced clarification questions with defeasibility rewards for social and moral situations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11253–11271, Toronto, Canada. Association for Computational Linguistics.

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

Kavel Rao, Liwei Jiang, Valentina Pyatkin, Yuling Gu, Niket Tandon, Nouha Dziri, Faeze Brahman, and Yejin Choi. 2023. What makes it ok to set a fire? iterative self-distillation of contexts and rationales for disambiguating defeasible social and moral situations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12140–12159, Singapore. Association for Computational Linguistics.

Raymond Reiter. 1978. On closed world data bases. In Hervé Gallaire and Jack Minker, editors, *Logic and Data Bases*, pages 55–76. Springer US, Boston, MA.

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

Rachel Rudinger, Vered Shwartz, Jena D Hwang, Chandra Bhagavatula, Maxwell Forbes, Ronan Le Bras, Noah A Smith, and Yejin Choi. 2020. Thinking like a skeptic: Defeasible inference in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4661–4675.

Abulhair Saparov and He He. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.

Alon Talmor, 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.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. 2021. Diagnosing the first-order logical reasoning ability through LogicNLI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3738–3747, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hugo Touvron, 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*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Jason Wei, 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.

Yeliang Xiu, Zhanhao Xiao, and Yongmei Liu. 2022. LogicNMR: Probing the non-monotonic reasoning ability of pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3616–3626, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

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

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

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, Karthik Ganesan, Wei-Lin Chiang, Jian Zhang, and Jiantao Jiao. Starling-7b: Improving helpfulness and harmlessness with rlaif. In *First Conference on Language Modeling*.

Caleb Ziems, Jane Dwivedi-Yu, Yi-Chia Wang, Alon Halevy, and Diyi Yang. 2023. NormBank: A knowledge bank of situational social norms. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7756–7776, Toronto, Canada. Association for Computational Linguistics.

A DefReasIng Details

All preprocessing code and relevant data files will be made publicly available under a CC-BY license.

A.1 Data Resources Used

***AnimalG-Ex*: Generics Exemplars Data** To construct DefReasIng, we use a subset of the dataset of generics and exemplars from Allaway et al. (2024). The license for this data was not specified by the authors but the paper is made available under a CC-BY license. Specifically, we use the *AnimalG* subset, which consists of generics about animals and the corresponding exemplars generated from their ExempliFI system. This subset contains 15028 generics and 325635 generated exemplars. We further filter this subset, keeping only generics that have valid exceptions. Note that Allaway et al. (2024) call these "default exceptions" (see §B for a discussion of this terminology). This filtering leaves a subset 8726 generics, along with their exemplars. We denote this subset *AnimalG-Ex*. Note that each generic (exemplar) consists of three components: concept, verb, and property. The datafiles include both parses (which indicate the text spans for the three components for item) and word forms (the plural and singular of the concept and verb, as well as the negation of the verb).

XCSLB Data We also use the XCSLB dataset (Devereux et al., 2014; Misra et al., 2022) in constructing DefReasIng. This data is licensed by Misra et al. (2023) under an Apache 2.0 License. XCSLB (Misra et al., 2022) is an expanded version of the CLSB (Centre for Speech, Language, and the Brain) dataset of concept-property norms (Devereux et al., 2014) used in Misra et al. (2023). The expanded dataset includes 521 concepts and 3927 properties. Note that these properties contain a verb, unlike in *AnimalG-Ex*.

The feature matrix (*FeatM*), indicating which concepts have which properties is very sparse. Therefore, we use the generics in *AnimalG-Ex* to somewhat decrease this sparsity. Specifically, we first match the concepts in *AnimalG-Ex* with the concepts in XCSLB. Then, for each generic we combine the verb and property to obtain a property span that can match with XCSLB. If the concept and property-span combination exists in XCSLB, we update *FeatM* accordingly. In total we make updates for 54 concepts with a median of 3 added feature values. We denote the updated *FeatM* as *FeatM+AG* and the portion of *AnimalG-Ex* that overlaps with XCSLB as *AnimalG-Ex-XCLSB*

A.2 Implementation Details

In order to extra $K^{\ominus}$ for the *N-alt* examples, we use XCSLB and the updated feature matrix *FeatM+AG*. The XCSLB data includes a categorization of concepts in 9 categories (bird, animal, sea creature, flower, invertebrate, vegetable, fish, food, fruit) which we use to categorize the concepts in *AnimalG-Ex-XCSLB*.

We now describe the process of obtaining $K^{\ominus}$ for a single generic with concept K. First, we randomly sample a category using a uniform distribution across categories, excluding the category that K belongs to. Next, having obtained this second category, we remove from its list of members any concept that *has* the property in the generic. Then we compute the similarity between K and each concept in the second category. To do this, we compute the Jaccard index between the feature vectors (obtained from *FeatM+AG*) for the two concepts. Finally, we select as candidate $K^{\ominus}$, the 5 concepts from the second category that are *most* similar to K. If there are fewer than 5 concepts, we take them all.

The *FeatM+AC* matrix is not complete in that the absence of a feature value (i.e., a concept is not annotated has having some property) does not guarantee that the concept does not have the property. Therefore, we construct a property statement for each candidate $K^{\ominus}$ of the form "Sometimes

			DefReasIng	DefReasIng∗*N-alt*
S-case	1step	R	8726	200
		X	8726	200
	2step	R	8726	200
		X	8726	200
S-alt		R	8217	194
		X	8217	194
W-case	1step	R	8726	200
		X	8726	200
	2step	R	8726	200
		X	8726	200
N-alt		R	200	200
		X	200	200
N-prop			8029	198

Table 5: Number of instances per question type DefReasIng and the DefReasIng∗*N-alt* subset. *1step* and *2step* indicate single and 2-step inheritance. *R* and *X* indicate real and nonsense types (see §3.4.2).

$K^{\ominus}$ [property-span]" (e.g., "Sometimes cats have wings") where [property-span] is the property as it appears in *FeatM+AG*. We then pass these statements to a truth filter. In particular, we give GPT-3.5-Turbo the following instruction

> Is the following statement true? Please answer only with "yes" or "no".

We keep candidates where the responses is "no". If multiple candidates pass the filter, we take the one that is most similar to the base concept K. We use the same hyperparameters for this filter and for the experiments with GPT-3.5 (see Appendix C.1).

For the examples in our dataset that use nonsense types, we randomly choose one of 6 possible nonsense types for each generic and use the same type for all questions based on that generic. These types are taken from Allaway et al. (2024) and are: Dofik, Yeb, Wumox, Bafu, and Goq.

A.3 Complete Statistics

We show complete dataset statistics for DefReasIng and DefReasIng∗*N-alt* in Table 5. Recall that DefReasIng small is a subset of DefReasIng.

A.4 Risks and Intended Use

The intended use of DefReasIng is to evaluate LM and facilitate improvements in LM reasoning. The data is not intended for fine-tuning models in its current state. This is because it is synthetically created and therefore likely contains spurious artifacts that a LM could learn during fine-tuning. Although fine-tuning is not the intended use, it is possible that developers may still use it for that purpose, thereby achieving high performance on the task. Additionally, once the data is made publicly available, it may be included at some point in the training data of future LMs. Therefore, in future caution should be used when evaluating LMs for which the training data includes data from 2024 onwards. Finally, we note that our dataset does not contain any information related to people, including personally identifying information.

B Semantics of Generics

We use semantics of generics and exemplars from Allaway et al. (2024) to extract the types used in the examples in DefReasIng. As noted in §3.4, we refer to exemplars using our own names for them for clarity. In the following, we briefly summarize the main idea of Allaway et al. (2024) and then describe how the names we use for exemplars line up with those in their original paper.

The semantic framework from Allaway et al. (2024) uses ideas about information structure to relate generics (e.g., "cats are cute") and exemplars to an implicit discourse question (e.g., "what is cute?"). Depending on the focus of that question, different types of exemplars are valid. In particular, they define five types of exemplars, four of which we use in constructing our dataset. We detail our name and the original term, along with a brief definition, below

- **Instantiations** (originally: *concept-focused instantiations*). These are subtypes of the concept that have the property from the generic. For example, for the generic "birds have wings", these are examples of birds that do have wings (e.g., seagulls, owls).

- **Exceptions** (originally: *default exceptions*). These are examples where the generic does not hold. For example, for the generic "birds can fly" these would be examples of birds that cannot fly (e.g., ostriches, emus).

- **Concept-alternate examples** (originally: *concept-focused exceptions*). These are examples of alternate relevant concepts that also have the property specified in the generic. For example, for "birds can fly", these might be concepts like "bats" or "flying squirrels".

- **Property-alternate examples** (originally: *property-focused exceptions*). These are examples of other properties that the concept from the generic possesses. For example, for

Mistral	`mistralai/Mistral-7B-Instruct-v0.2`
Mixtral	`mistralai/Mixtral-8x7B-Instruct-v0.1`
Hermes	`teknium/OpenHermes-2.5-Mistral-7B`
Starling	`berkeley-nest/Starling-LM-7B-alpha`
Zephyr	`HuggingfaceH4/zephyr-7b-beta`
Llama2	`meta-llama/Llama-2-13b-chat-hf`
Llama3	`meta-llama/Meta-Llama-3-8B-Instruct`
Llama3.1	`meta-llama/Meta-Llama-3.1-8B-Instruct`
Wizard	`WizardLMTeam/WizardLM-13B-V1.2`
GPT-3.5	`gpt-3.5-turbo-0613`
GPT-4	`gpt-4-turbo-2024-04-09`
GPT-4o	`gpt-4o-2024-05-13`

Table 6: Specific checkpoints of models used in our experiments.

"birds can fly" properties might include "sing" and "build nests".

We note that the full semantics as defined by Allaway et al. (2024) is quite complicated, drawing on multiple linguistic concepts, and we therefore refer the interested reader to their original paper for full details.

C Experiment Details

C.1 Hyperparamters

We use Huggingface[8] to run models for our experiments. The specific checkpoints used are shown in Table 6. For *Mistral*, we use the `mistral-inference` and `mistral-common` packages to run inference. All experiments are run on an NVIDIA RTX A6000 GPU. For generation, we set the maximum generation length to 10 new tokens, and we use temperature 0.0. For OpenAI models, we set both the frequency penalty and presence penalty to 0.0.

As noted in the limitations, we run *Mixtral* quantized due to computational constraints. For this, we use the BitsAndBytes package[9]. Additionally, the following models are loaded in 16-bit floats since they were trained that way: *Llama2*, *Llama3*, *Llama3.1*, and *Wizard*.

C.2 Prompts

For all models except *Wizard* we use Huggingface to automatically construct model input using the model-specific chat template format. Since this format is not implemented for *Wizard*, we automatically construct the input with the format shown in Table 7.

[8] `huggingface.co`

[9] `https://huggingface.co/docs/bitsandbytes/main/en/index`

A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions. [Instructions] USER: [Prompt] ASSISTANT:

Table 7: Chat template for generating model output from *Wizard*. [Square brackets] indicate placeholders for the system instructions and prompt.

As discussed in §4.2, we average our results over three prompts (see Table 8). The first prompt (see (a)) uses natural and colloquial language while the second (see (b)) uses logic terms. The third prompt uses chain-of-thought prompting with the more natural language from the first prompt (see (c)).

(a)	*Please answer with only "more likely", "less likely", or "it has no impact".* Consider the following information: [$\mathcal{P}^i$] From this information we can draw a conclusion about [$K^{\mathcal{H}}$]. Conclusion: [$\mathcal{H}$]. Now suppose we are given additional information. Additional information: [$\mathcal{P}^x$] Given the additional information, how likely are you to believe the conclusion?
(b)	*Please answer with only "strengthens", "weakens" or "it has no impact".* Consider the following premises: [$\mathcal{P}^i$] This entails the conclusion that [$\mathcal{H}$]. Additional information: [$\mathcal{P}^x$] How does the given additional information impact the conclusion??
(c)	*Please answer with only "more likely", "less likely", or "it has no impact".* Let's think step by step. First, consider the following information: [$\mathcal{P}^i$] From this information we can draw a conclusion about [$K^{\mathcal{H}}$]. Conclusion: [$\mathcal{H}$]. Now suppose we are given additional information. Additional information: [$\mathcal{P}^x$] Given the additional information, how likely are you to believe the conclusion?

Table 8: The three prompts used in our experiments. The system instruction is in *italics* about the dashed line.

D Supplementary Results

We show in Table 9 the full results across the 13 types of instances for each model and prompt.

In Table 10 we show the Pearson's r correlation between the presence of "not" in the additional premises and the prediction of "weakens" by a

	S-case				*S-alt*		*W-case*				*N-alt*		*N-prop*	Overall		
	1step		2step				1step		2step							
	R	X	R	X	R	X	R	X	R	X	R	X		Acc.	$F1$	
Mistral	0.957	0.985	0.986	0.989	0.127	0.027	0.052	0.085	0.234	0.495	0.975	0.955	0.749	0.522	0.458	**(a)**
	0.794	0.568	0.844	0.827	0.012	0.001	0.562	0.798	0.984	0.899	1.0	1.0	0.777	0.650	0.613	**(b)**
	0.922	0.973	0.952	0.992	0.016	0.001	0.047	0.111	0.135	0.211	1.0	1.0	0.857	0.479	0.411	**(c)**
Mixtral	0.949	0.990	0.956	0.988	0.398	0.545	0.582	0.730	0.961	0.893	0.535	0.335	0.376	0.766	0.647	**(a)**
	0.905	0.970	0.977	0.981	0.046	0.154	0.446	0.276	0.616	0.423	0.990	0.695	0.734	0.599	0.547	**(b)**
	0.925	0.975	0.889	0.937	0.329	0.398	0.472	0.549	0.901	0.340	0.770	0.670	0.390	0.672	0.580	**(c)**
Starling	0.996	1.0	0.999	0.998	0.231	0.233	0.985	0.998	0.999	0.999	0.0	0.0	0.001	0.775	0.542	**(a)**
	1.0	1.0	1.0	1.0	0.967	0.997	0.029	0.003	0.041	0.009	0.005	0.0	0.125	0.557	0.319	**(b)**
	0.996	1.0	0.999	0.998	0.273	0.254	0.984	0.999	0.999	0.999	0.0	0.0	0.0	0.781	0.546	**(c)**
Zephyr	0.903	0.897	0.862	0.973	0.216	0.179	0.860	0.985	0.990	0.998	0.160	0.230	0.311	0.750	0.627	**(a)**
	0.573	0.070	0.445	0.316	0.009	0.0	0.726	0.829	0.974	0.990	0.975	0.760	0.849	0.529	0.514	**(b)**
	0.917	0.941	0.905	0.980	0.114	0.085	0.806	0.989	0.981	0.999	0.225	0.575	0.338	0.742	0.628	**(c)**
Hermes	0.981	0.987	0.989	0.989	0.541	0.630	0.985	0.998	0.998	0.999	0.0	0.0	0.0	0.832	0.581	**(a)**
	0.999	0.999	0.998	0.997	0.879	0.974	0.900	0.948	0.924	0.976	0.0	0.0	0.033	0.877	0.634	**(b)**
	0.978	0.980	0.988	0.987	0.324	0.315	0.997	0.999	0.999	1.0	0.0	0.0	0.0	0.789	0.550	**(c)**
Llama2	0.30	0.967	0.947	0.971	0.845	0.874	0.193	0.138	0.369	0.399	0.015	0.015	0.021	0.604	0.398	**(a)**
	0.002	0.0	0.005	0.004	0.001	0.0	0.011	0.0	0.141	0.015	1.0	1.0	1.0	0.105	0.083	**(b)**
	0.920	0.956	0.877	0.795	0.547	0.545	0.486	0.563	0.707	0.830	0.020	0.035	0.032	0.663	0.484	**(c)**
Llama3	0.997	1.0	0.999	1.0	0.938	0.986	0.886	0.988	0.999	1.0	0.0	0.0	0.0	0.892	0.625	**(a)**
	0.998	1.0	0.998	0.996	0.886	0.886	0.931	0.914	1.0	1.0	0.005	0.0	0.063	0.882	0.653	**(b)**
	0.997	1.0	0.999	1.0	0.936	0.982	0.942	0.993	1.0	1.0	0.0	0.0	0.0	0.897	0.629	**(c)**
Llama3.1	0.946	0.910	0.973	0.980	0.652	0.427	0.981	0.993	0.999	1.0	0.0	0.0	0.005	0.811	0.569	**(a)**
	0.001	0.0	0.030	0.036	0.0	0.0	1.0	1.0	1.0	1.0	0.220	0.145	0.115	0.385	0.253	**(b)**
	0.954	0.957	0.953	0.972	0.692	0.559	0.968	0.990	1.0	1.0	0.005	0.0	0.006	0.828	0.581	**(c)**
Wizard	0.834	0.866	0.733	0.472	0.220	0.241	0.709	0.956	0.925	0.983	0.005	0.005	0.108	0.646	0.500	**(a)**
	0.861	0.733	0.959	0.978	0.751	0.472	0.774	0.634	0.905	0.769	0.240	0.210	0.193	0.733	0.592	**(b)**
	0.750	0.794	0.603	0.411	0.184	0.126	0.770	0.927	0.845	0.908	0.020	0.005	0.199	0.598	0.485	**(c)**
*GPT-3.5**	0.995	1.0	0.990	1.0	0.892	0.849	0.650	0.860	0.895	0.900	0.010	0.025	0.030	0.699	0.546	**(a)**
	0.985	0.965	0.985	0.985	0.114	0.083	0.915	0.905	0.990	0.985	0.650	0.490	0.354	0.729	0.699	**(b)**
	1.0	1.0	0.980	0.980	0.990	0.959	0.690	0.985	0.940	1.0	0.015	0.005	0.005	0.734	0.562	**(c)**
*GPT-4**	0.020	0.0	0.430	0.145	0.0	0.0	0.760	0.960	0.970	0.980	1.0	1.0	0.995	0.561	0.550	**(a)**
	0.010	0.0	0.145	0.0	0.0	0.0	0.52	0.96	1.0	0.99	1.0	1.0.	1.0	0.512	0.489	**(b)**
	0.030	0.0	0.060	0.005	0.0	0.0	0.53	0.92	0.940	0.955.	1.0	1.0	1.0	0.497	0.474	**(c)**
*GPT-4o**	0.005	0.0	0.185	0.165	0.0	0.0	0.995	1.0.	1.0	1.0	1.0	1.0	0.944	0.563	0.540	**(a)**
	0.025	0.005	0.185	0.045	0.0	0.0	0.960	1.0	0.990	0.975	1.0	1.0	0.995	0.555	0.527	**(b)**
	0.005	0.005	0.055	0.015	0.0	0.0	0.960	1.0	1.0	1.0	1.0	1.0	0.889	0.535	0.503	**(c)**

Table 9: Accuracy on DEFREASING across reasoning types (§3.3) and macro-$F1$. * indicates the evaluation is done only on DEFREASING∗*N-alt*. *1step* and *2step* indicate single and 2-step inheritance respectively. *R* and *X* indicate real and nonsense types respectively, as used in the additional premises. (a), (b), and (c) indicate the three different prompts used (see Table 8 for examples).

model. Note that correlation is computed across examples where "weakens" is *not* the correct label. The p-values for significance are computed using a two-sided t-test. The correlation and p-values computed using Scipy[10] and therefore p-values of 0.0 are indicative of numerical underflow. Additionally, *GPT-4o* does not predict any "weakens" label for the subset considered for prompts (b) and (c). Therefore, the correlation (and p-value) are undefined (NaN) in these cases.

[10] https://docs.scipy.org/doc/scipy/reference/stats.html

Model	Prompt	*r*	*p-value*
Mistral	(a)	-0.0203	6.55×10^{-7}
	(b)	0.00446	0.2758
	(c)	-0.01383	0.0007
Mixtral	(a)	0.20015	0.0
	(b)	0.00493	0.2277
	(c)	0.10161	6.66×10^{-137}
Starling	(a)	0.76711	0.0
	(b)	-0.00363	0.3755
	(c)	0.76647	0.0
Zephyr	(a)	0.17775	0.0
	(b)	0.03781	2.30^{-20}
	(c)	0.19970	0.0
Hermes	(a)	0.41424	0.0
	(b)	0.11259	7.79×10^{-168}
	(c)	0.62790	0.0
Llama2	(a)	0.16859	0.0
	(b)	0.01053	0.0100
	(c)	0.34940	0.0
Llama3	(a)	-0.00822	0.0444
	(b)	0.16597	0.0
	(c)	0.01328	0.0012
Llama3.1	(a)	0.38990	0.0
	(b)	0.01775	1.42×10^{-5}
	(c)	0.33312	0.0
Wizard	(a)	0.41731	0.0
	(b)	0.31695	0.0
	(c)	0.44461	0.0
GPT-3.5*	(a)	0.24811	1.84×10^{-26}
	(b)	0.36812	1.98×10^{-58}
	(c)	0.24427	1.12×10^{-25}
GPT-4*	(a)	-0.02103	0.3747
	(b)	NaN	NaN
	(c)	NaN	NaN
GPT-4o*	(a)	-0.02574	0.2769
	(b)	0.00396	0.8670
	(b)	-0.04210	0.0753

Table 10: Pearson's *r* correlation and corresponding p-values for each model and prompt combination. Correlation is computed between the presence of "not" in the additional premises $\mathcal{P}^x$ and the prediction of "weakens" as the label. * indicates the computation is done on DEFREASINGS$*$*N-alt*.