Are Machines Better at Complex Reasoning? Unveiling Human-Machine Inference Gaps in Entailment Verification

Anonymous ACL submission

Abstract

Making inferences in text comprehension to understand the meaning is essential in language processing. This work studies the entailment verification (EV) problem of multi-sentence premises that requires a system to make multiple inferences implicitly. Studying EV for such complex premises is important because modern NLP problems, such as detecting *inconsistent* model-generated rationales, require complex multi-hop reasoning. However, current textual inference datasets mostly contain short premises that only partially focus on these challenges. To address this, we compile an EV benchmark that includes datasets from three NLP domains (NLI, contextual QA, and rationales) containing multi-sentence premises. On benchmarking humans and LLMs, we find that LLMs are better than humans in multi-hop reasoning across extended contexts, while humans perform better in simple deductive reasoning tasks. We also finetune a Flan-T5 model for EV using two training objectives to obtain a strong open-source model that outperforms GPT-3.5 and rivals GPT-4. Finally, we use this model to filter out inconsistent model-generated rationales in self-consistency decoding, resulting in a 6% accuracy improvement on average across three MCQ datasets.[1]

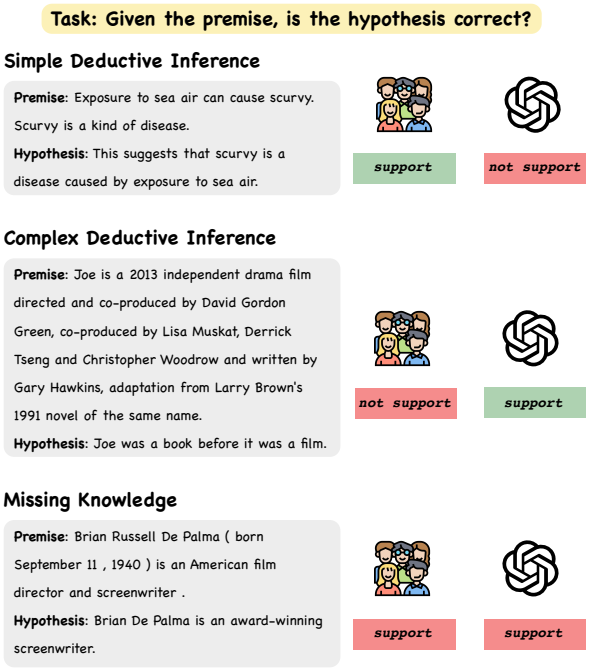

Figure 1: **Distinctions between human and LLM Inferences.** Examples of each reasoning type, along with the human and GPT-4 prediction, are shown. A green box means the prediction matches the true label, and red otherwise. Humans are more consistent in simple deductive reasoning, whereas LLMs excel in complex, multi-step inferences over long contexts. Both humans and LLMs are comparable in instances with missing knowledge. Please refer to Section 2 for more details.

1 Introduction

A prevailing notion in cognitive psychology exists that humans make numerous inferences to understand discourse and text (Garnham, 1989). These inferences play a crucial role in linking information from disparate sections of a text to establish its literal meaning and are closely associated with *reasoning*. With the recent applications of Large Language Models (LLMs) (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020; Chung et al., 2022) in NLP tasks that require inference skills (Minaee et al., 2021), it is thus essential to understand and improve upon the limitations of LLMs concerning different aspects of language inferences.

In this work, we focus on the task of *entailment verification* (EV) that classifies whether a given context supports a hypothesis. To assert the validity of a hypothesis, a system has to make multiple inferences from the given context and its internal knowledge, which requires complex multi-hop reasoning. Verifying the entailment of such complex premises has important applications in rationale-generating LLMs, such as chain-of-thoughts (CoT) (Wei et al., 2022), that typically generate multi-sentence ra-

[1]The code and data are uploaded here. We will publish the code, data, and finetuned checkpoints upon acceptance.

Desirable Properties	NLI			Contextual QA					Rationale	
	WaNLI	FEVER	ANLI	CosQA	SIQA	DREAM	BoolQ	RACE	Entailer	ECQA
Multi-sentence premise		✓	✓	✓		✓	✓	✓	✓	✓
Explanatory premise									✓	✓
Entity-grounded knowledge		✓	✓				✓		✓	
Commonsense knowledge				✓	✓					✓
Localized knowledge	✓			✓	✓	✓		✓		

Table 1: Comparisons between different datasets used for evaluation. We compare on two broad categories: type of premise (multi-sentence and explanatory) and type of knowledge tested (entity-grounded, commonsense, and localized). Please refer to Section 2.1 for more details.

tionales with incomplete information and suffer from inconsistent reasoning (Ye and Durrett, 2022). While EV is similar to Natural Language Inference (NLI) (Dagan et al., 2006; Manning and MacCartney, 2009), some interesting challenges distinguish it from NLI. Many existing textual inference datasets, such as SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), etc., mostly contain short sentence premises that only partially encapsulate the challenges of multi-sentence premises requiring complex reasoning. Predicting the entailment of such complex premise-hypothesis pairs often requires multi-hop reasoning, inferring missing information, etc., which is lacking from standard NLI datasets (Gururangan et al., 2018; McCoy et al., 2019). Thus, studying EV in the context of modern LLMs is essential and is currently missing.

To this end, we first compile an evaluation benchmark to study the EV problem by selecting multiple datasets across three categories: NLI, contextual QA, and rationales. As shown in Table 1, the datasets typically contain multi-sentence premises that require inferring different types of knowledge to predict the entailment. Thus, this benchmark is more complex than standard NLI datasets and can be used as a new evaluation benchmark.

Next, we evaluate LLMs and humans on this benchmark and make some interesting observations. Cognitive studies (Buschman et al., 2011; Cowan, 2001) have shown that an average human brain has a limited capacity to retain only four chunks in short-term memory, indicating a limitation of human inference abilities over long contexts. Our analysis shows that LLMs are indeed stronger than humans at tasks that involve multi-hop reasoning across long contexts, reinforcing this human limitation. In contrast, humans are better in cases that require simple deductive reasoning using substitutions, negations, etc., indicating that current LLMs lack consistency along these reasoning aspects. Further, humans and LLMs perform comparably in instances requiring inferring missing knowledge. These findings are depicted in Figure 1 with motivating examples.

Additionally, on comparison between LLMs, we find that models finetuned on a specific dataset category are usually strong within the category but don't generalize well on unseen categories. In contrast, instruction-finetuned models are better on average, with the best performing model, GPT-4, achieving 0.79 macro-F1 and outperforming the best-open-sourced model Flan-T5-xxl by 0.08 macro-F1 on average. In order to bridge this gap, we finetune a Flan-T5 (Chung et al., 2022) model on a training subset containing datasets from each category of the above data collection. To this end, we explore two different training approaches: a *classification-based* finetuning that learns to directly predict the label and a *ranking-based* finetuning that learns to rank the most supported hypothesis from a given pair of hypotheses for a given premise. Ranking-based finetuning is better than classification, specifically in contextual QA datasets, as it can learn a softer decision boundary. Overall, our fine-tuned Flan-T5 outperforms GPT-3.5, baseline Flan-T5, and performs comparably to GPT-4 on the benchmark, thus providing a strong open-sourced model for entailment verification.

Finally, we demonstrate the utility of our fine-tuned models on a downstream application of filtering unfaithful model-generated explanations. Self-consistency (SC) (Wang et al., 2023) decoding samples multiple model-generated reasoning paths from the LLM decoder and aggregates them to predict the most consistent answer. We use our finetuned Flan-T5 to filter out non-entailed reasoning chains before aggregating the final prediction, which leads to 6% performance improvement on average across three MCQ datasets.

2 Entailment Verification using LLMs

In this section, we formally define the task of entailment verification (EV), the datasets used to create the evaluation benchmark, and the evaluation procedure for evaluating different LLM baselines.

For a given premise p and a hypothesis (or claim) h, the task of entailment verification is to determine whether the context has information that directly confirms the hypothesis or not, i.e., whether the hypothesis follows from the information present in the context. This is a binary classification task defined as $f(p, h) = \{support, not\ support\}$, where f is a classifier (human/LLM).

2.1 Evaluation Benchmark

In this section, we list some desirable properties we want to include in the EV benchmark, followed by details of the dataset categories we select.

- **Type of Premise**: Typically, NLI datasets, such as SNLI, MNLI, etc., do not contain more than one sentence in the premise, potentially leading to shortcut learning. In contrast, we focus more on multi-sentence premises that require complex reasoning. We also consider datasets where the premise is a *rationale*, i.e., the premise is not just a logical precursor to the hypothesis but rather an explanation. This tests the ability to evaluate model-generated rationales (Wei et al., 2022).
- **Type of Knowledge**: Often, one or more information in the premise needs to be used to predict support. We categorize this information as entity-grounded, commonsense, or localized. Entity-grounded knowledge consists of information about entities and other general knowledge verifiable on the internet. These can be facts about general science, history, etc., or details of some known person, event, etc. It is possible to infer this information even if not mentioned in the premise. The commonsense knowledge is typically all information about everyday life that humans use implicitly but cannot always be verified online. This information is often missing from the premise and has to be inferred implicitly. Lastly, localized information is all other knowledge provided for understanding the events, people, or items mentioned in the premise that are not grounded to any known entity. This information depends on the premise's specific context and, thus, is impossible to infer unless stated explicitly. Please refer to Table 2 for examples of each knowledge type.

Knowledge	Examples
Entity-grounded	Barack Obama is born in USA.; Electrical energy is used by plants for making food.
Commonsense	If you are hurting, you might cry.; If you steal something, you can get in trouble.
Localized	The policeman helps her find her daughter.; Dan is 72 years old currently.

Table 2: Examples of different categories of knowledge. Please refer to Section 2.1 for more details.

We consider three data sources for creating the entailment verification benchmark, amounting to 10 datasets in total. In Table 1, we compare these datasets across the desirable characteristics mentioned earlier. Please refer to Appendix A for more details on the datasets used.

Natural Language Inference Given the close connection between NLI and EV, it is an obvious choice to consider appropriate NLI datasets for the benchmark. To convert an NLI dataset for EV, we merge the *neutral* and *contradict* labels to the *not support* label. We use the following NLI datasets in our benchmark: WaNLI (Liu et al., 2022), FEVER (Nie et al., 2019), and ANLI (Nie et al., 2020).

Contextual QA Next, we consider multiple-choice question-answering datasets where the task is to answer a question based on a given context and some options. We use a QA-to-statement converter model (Chen et al., 2021) to generate a hypothesis statement for each question option pair. Then, the hypothesis corresponding to the correct choice is marked as "*support*", while the rest are marked as "*not support*". This process is depicted in Figure 5 in Appendix A. We include the following datasets from this category: Cosmos QA (CosQA) (Huang et al., 2019), SocialIQA (SIQA) (Sap et al., 2019), DREAM (Sun et al., 2019), BoolQ (Clark et al., 2019), and RACE (Lai et al., 2017).

Rationale Lastly, we consider data sources where human-annotated explanations are available that justify the original hypothesis. In this case, we use the rationales as the premise. We use the following datasets: Entailer (Tafjord et al., 2022), and ECQA (Aggarwal et al., 2021).

2.2 Evaluation Metric

We use the macro-F1 score as the primary evaluation metric for comparing LLMs on the entailment verification task because there are label im-

Model	NLI			Contextual QA					Rationale		Avg
	WaNLI	FEVER	ANLI	CosQA	SIQA	DREAM	BoolQ	RACE	Entailer	ECQA	
RoBERTa	**0.81**	**0.92**	0.67	0.46	0.45	0.63	0.71	0.41	0.87	0.37	0.63
Entailer-11B	0.68	0.75	0.59	0.71	0.56	0.67	0.81	0.49	**0.91**	**0.49**	0.67
Flan-T5-xxl	0.71	0.79	0.68	0.66	0.55	0.79	**0.86**	0.60	0.88	**0.49**	0.70
GPT-3.5	0.76	0.81	0.62	0.67	0.59	0.79	0.76	0.52	0.76	0.48	0.68
GPT-4	0.79	0.86	**0.79**	**0.76**	0.61	**0.90**	0.84	**0.68**	0.80	0.48	**0.75**
Human	0.74 *(0.78)*	0.88 *(0.75)*	0.67 *(0.62)*	0.63 *(0.51)*	**0.74** *(0.54)*	0.87 *(0.68)*	0.77 *(0.67)*	0.61 *(0.57)*	**0.91** *(0.82)*	0.48 *(0.85)*	0.73 *(0.68)*
Human − GPT-4	-0.05	0.02	-0.12	-0.13	0.13	-0.03	-0.07	-0.07	0.11	0.00	-0.02

Table 3: Comparisons between human and baseline LLMs on 100 sampled instances from each dataset. We report the macro-F1 score and highlight the best results in bold. For the human baseline, we report the annotation agreements in parenthesis. **Takeaways**: Task-finetuned models perform well on specific dataset categories seen during finetuning, but instruction-finetuned models generalize better on average. GPT-4 is the best-performing LLM. It outperforms humans on ANLI and CosQA that require complex, multi-step reasoning. In contrast, humans are better on SIQA and Entailer that require simple deductive reasoning. Please refer to Section 3 for more analysis.

balances in our evaluation datasets. The macro-F1 score computes the unweighted mean of F1 scores for each class, ensuring equal importance for each class irrespective of the label statistics. Please refer to Appendix C for more discussions on the label imbalance of each dataset.

2.3 LLM Evaluation Setup

We evaluate two types of LLMs on the task of EV, as categorized below:

Task-finetuned LLMs In this, the models considered are already finetuned on some subset of the benchmark. We evaluate two models: RoBERTa (Liu et al., 2019) (finetuned on NLI datasets) and Entailer-11B (Tafjord et al., 2022) (finetuned on Entailer dataset). Refer to Appendix B.1 for more details on the evaluation setup for these models.

> **Premise**: {*premise*}
> **Hypothesis**: {*hypothesis*}
> **Question**: Given the premise, is the hypothesis correct?
> **Answer**:

Box 1: Prompt used to evaluate instruction-finetuned LLMs for entailment verification.

Instruction-finetuned LLMs These are language models trained on a collection of NLP tasks described using instructions, leading to generalization abilities to solve unseen tasks described using new instructions. Here, we evaluate Flan-T5-xxl (Chung et al., 2022), GPT-3.5 (Brown et al., 2020), and GPT-4 (OpenAI, 2023) models. To compute the label, we first modify a given premise-hypothesis pair (p, h) into a prompted input $\mathcal{P}$ using the prompt template as shown in Box 1. Next, we compute a score s as defined below:

$$s(p,h) = \frac{p_{LLM}(\text{“Yes”}|\mathcal{P})}{p_{LLM}(\text{“Yes”}|\mathcal{P}) + p_{LLM}(\text{“No”}|\mathcal{P})}, \quad (1)$$

where $p_{LLM}(\cdot|\mathcal{P})$ is the model's probability distribution over the vocabulary. If the score s is higher than a threshold (typically set to 0.5 in all our experiments), we assign the label *support*, else we assign the label *not support*. For GPT-4 evaluation, we directly check for the “Yes” / “No” label prediction as the token probabilities are not accessible via the API. Please refer to Appendix B.2 for more details about the models and ablations on prompts.

3 Evaluation of Humans and LLMs

First, we randomly sample 100 instances from each dataset (i.e., 1000 instances in total) and conduct a human evaluation on this subset to estimate average human performance. Please refer to Appendix E.1 for more details on the annotation procedure. Additionally, we evaluate the above LLMs on this sampled subset and report those numbers for fair comparisons with humans. Table 3 shows the overall evaluation results.

3.1 Comparison among LLMs

We observe that the task-finetuned models (rows 1-2 in Table 3) are weaker on average compared to the instruction-finetuned models (rows 3-5 in Table 3). However, there are some interesting exceptions. RoBERTa, which is finetuned on NLI

datasets, performs at par with GPT-4 on FEVER and outperforms it on WaNLI but falls behind on contextual QA and rationale datasets. On the other hand, Entailer-11B, which is finetuned on the Entailer dataset, outperforms GPT-4 on Entailer but lags on most of the other datasets. This demonstrates that finetuning an LLM using datasets from one of these categories is ineffective in outperforming models trained on more general data. Overall, we observe that GPT-4 is the best-performing entailment verification model on average and Flan-T5-xxl is the best open-source LLM for this task.

3.2 Comparison between Humans and LLMs

In Table 3, we report the human performance and the corresponding annotation agreement[2] (in parenthesis) for each dataset. We note that the agreement ratios are lowest for Contextual QA datasets, especially CosQA and SIQA. Questions in these datasets are often based on commonsense scenarios, and sometimes, the support for the wrong hypothesis can be debatable. Please refer to Appendix D for examples. Between humans and LLMs, we find that humans beat all the baseline LLMs, except GPT-4. This shows that existing open-sourced LLMs are subpar with humans on this task. Additionally, although humans and GPT-4 perform comparably on average, we note that large misalignments exist in different individual datasets. Specifically, we observe that ANLI, CosQA, SIQA, and Entailer are the four datasets with > 0.1 absolute macro-F1 difference. We analyze these four datasets in Section 3.3.

3.3 Effect of Reasoning Type

We design an analysis to understand further the misalignments between GPT-4 and humans. First, we categorize the type of reasoning required to predict an entailment into the following four categories:

- **Simple Deductive (R1)**: The premise contains sentences that can be minimally combined in one step to predict the support for the hypothesis. This typically tests skills such as substitution, understanding negations, word meanings, etc.
- **Complex Deductive (R2)**: More than one step of reasoning is required to solve the task. Typically, this tests skills like mathematical reasoning, combining multiple information in context, etc.

[2]We use a pairwise agreement ratio that computes the fraction of matched annotations over all pairs of annotations.

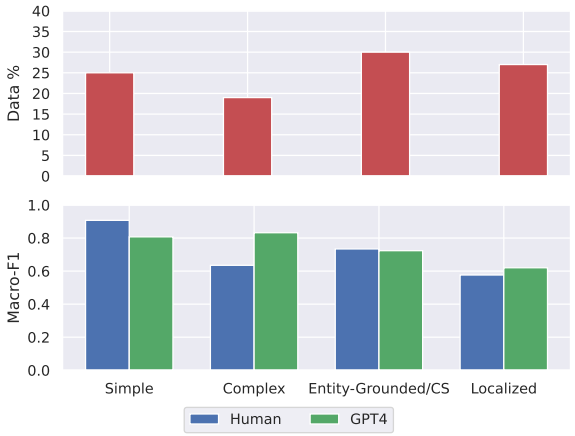

Figure 2: Analysis of the different reasoning types involved in entailment verification. **[Top]** Distribution of the reasoning types aggregated across four datasets: ANLI, CosQA, SIQA, and Entailer. **[Bottom]** Performance comparisons between humans and GPT-4. **Takeaway**: GPT-4 is better than humans at instances requiring complex reasoning, while humans are more consistent in simpler deductive reasoning tasks. Refer to Section 3.3 for more details.

- **Missing Entity-grounded/Commonsense Knowledge (R3)**: In this, some essential commonsense or entity-grounded knowledge is missing in the premise. Both humans and LLMs can implicitly invoke such information from the memory or the parametric knowledge obtained from pertaining, respectively.
- **Missing Localized Knowledge (R4)**: In this, information very specific to the premise is missing. Typically, this is information about the subjects in the context and is not grounded in any known entities. It is practically impossible for humans or LLMs to infer such information.

We note that these categories are mutually exclusive.[3] Figure 2 depicts the aggregated results. The top plot shows the percentage of each reasoning type among 400 samples from ANLI, CosQA, SIQA, and Entailer, and the bottom plot compares the human and GPT-4 macro-F1 scores. Please refer to Appendices E.2, E.3, and E.4 for details on the annotation setup, examples of each reasoning type, and detailed analysis, respectively.

The first type in Figure 2 is simple deductive reasoning ($\sim 25\%$ data). Here, humans perform better than GPT-4 by a small margin. Instances that require simple deductive reasoning usually use substitutions, negations, paraphrasing, etc., to prove entailment. We find that humans are more robust than GPT-4 in performing such simple deductive

[3]Deductive reasoning implies that the premise has all the necessary information. Thus, any missing knowledge instance falls under inductive reasoning.

Model	NLI			Contextual QA					Rationale		Avg	Seen Avg	Unseen Avg
	WaNLI	FEVER	ANLI†	CosQA	SIQA	DREAM	BoolQ	RACE†	Entailer	ECQA†			
GPT-4	**0.73**	**0.88**	**0.86**	0.79	0.69	**0.92**	**0.86**	**0.85**	**0.86**	0.48	0.79	-	-
GPT-3.5	0.70	0.83	0.69	0.70	0.67	0.81	0.78	0.69	0.82	0.48	0.72	-	-
Flan-T5-xxl	0.63	0.81	0.73	0.59	0.67	0.80	0.85	0.70	0.83	**0.50**	0.71	-	-
Flan-T5-xxl + *Class*	0.71	0.86	0.79	0.66	0.72	0.88	0.85	**0.85**	0.85	0.49	0.77	0.71	0.79
Flan-T5-xxl + *Rank*	0.69	0.85	0.77	**0.83**	**0.74**	0.89	0.85	**0.85**	**0.86**	0.48	0.78	0.70	0.82

Table 4: Comparison of classification and ranking-based Flan-T5-xxl finetuning with baseline LLMs on the complete evaluation benchmark. We report the macro-F1 for all datasets. †: Dataset is used in fine-tuning, and the average is reported in the "Seen Avg" column. Other datasets are zero-shot evaluated, and the average is reported in the "Unseen Avg" column. **Takeaways**: Ranking objective is better than classification on contextual QA datasets. Flan-T5-xxl + *Rank* outperforms Flan-T5-xxl and GPT-3.5, and performs comparably to GPT-4. Please refer to Section 4.2 for more details.

reasoning tasks, which is also observed in prior works (Sanyal et al., 2022; Nguyen et al., 2023).

Next, we find that GPT-4 significantly outperforms the humans on complex reasoning that constitutes $\sim$ 20% data. This usually requires two skills: understanding multiple relevant information in the premise and combining them for reasoning. GPT-4 is likely a stronger context processor, especially for long premises, since it has been trained on long-context data sources (OpenAI, 2023). In contrast, cognitive studies (Buschman et al., 2011; Cowan, 2001) have shown that an average human brain can retain only four chunks in short-term memory, thus limiting the long-context processing abilities of humans.

Lastly, we observe that approximately 30% of the data has some missing entity-grounded or commonsense information while $\sim$ 25% of the data has missing localized information. To correctly predict entailment in such cases, a system needs to infer some missing grounded knowledge while not hallucinating specific localized information not mentioned in the premise. We find that both humans and models are comparable across the reasoning types **R3** and **R4**, with **R4** being more challenging. This shows that both models and humans tend to hallucinate missing localized information.

4 Training LLMs for Entailment Verification

In Section 3.2, we observed that open-sourced LLMs lack performance compared to humans and close-sourced models such as GPT-4. From Section 3.1, we know that task-finetuned models perform well on a category when finetuned on data from the same category. Using this insight, we finetune the Flan-T5-xxl model on the train splits of datasets from each category, resulting in using ANLI (Nie et al., 2020), RACE (Lai et al., 2017), and ECQA (Aggarwal et al., 2021) for finetuning. Please refer to Appendix F.1 for more details on our training dataset selection criteria. Next, we describe the finetuning approaches and our key findings.

4.1 Finetuning Formulations

This section describes the two finetuning formulations explored in this work.

Classification This is the standard training paradigm where we finetune a Flan-T5-xxl model using the training data. We follow the same steps as the evaluation setup to create a prompted input using the prompt format in Box 1 and then define the cross-entropy loss over the "Yes" and "No" token logits. We refer to this finetuned model as "Flan-T5-xxl + *Class*".

Ranking In this approach, for a given premise and hypothesis pair (p, h), we define a *weaker hypothesis* h' as a statement such that the premise p supports h more strongly than h'. Then, for a given triplet (p, h, h'), we formulate the ranking task as predicting the hypothesis that is *more* supported by the premise. Given the triplet (p, h, h'), we define the margin ranking loss as follows:

$$\mathcal{L}_{ranking} = \max\{0, s(p, h) - s(p, h') + m\}, \quad (2)$$

where $s(p, h)$ is the entailment score as defined in Equation 1. The key advantage of this formulation over the classification is that ranking, by design, learns a softer decision boundary between the two labels. This can lead to better generalization, especially for contextual QA datasets. Sometimes, the wrong choice can be relatively less favorable compared to the best option in QA instead of being incorrect. As discussed in Section 3.2, this is indicated by the low agreement between human raters. Training to hard-classify the hypothesis for such options can be avoided by ranking them with the

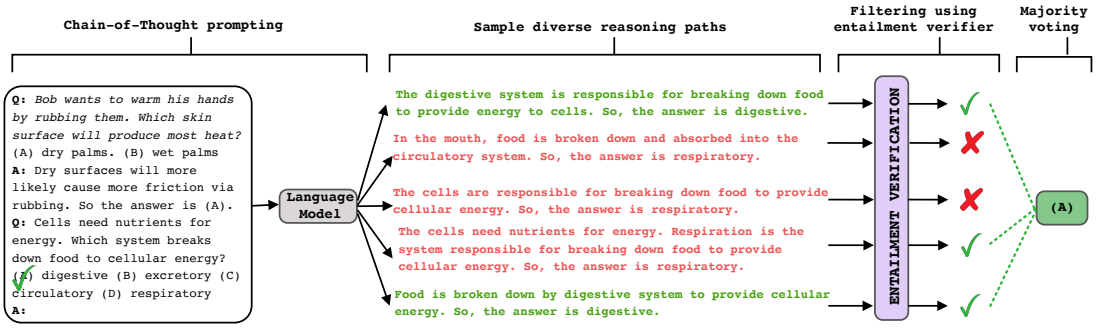

Figure 3: **Example of filtering CoT Rationales.** It consists of four steps: (1) CoT prompting, (2) Sampling multiple reasoning paths from the LLMs decoder, (3) Filtering out reasoning paths that don't support the model's prediction, (4) Aggregating the filtered reasoning paths to select the most consistent answer. The figure is inspired by self-consistency (Wang et al., 2023). Note that we only filter *inconsistent* rationales and not non-factual ones. Please refer to Section 5 for more details.

best hypothesis (corresponding to the right choice), thus learning a softer classification boundary. We refer to the finetuned model using the ranking objective as "Flan-T5-xxl + *Rank*". Please refer to Appendix F.2 for more details on the training data collection process for ranking.

4.2 Findings

Table 4 shows the evaluation results on the complete evaluation set (i.e., we use all the data points instead of 100 samples per dataset, which was used in Table 3). For our models, we separately average the results for the datasets already seen in training (namely, ANLI, RACE, and ECQA) and unseen during training into two columns, seen and unseen, respectively. First, we observe that all our finetuned models are consistently better across nine of ten datasets than the baseline Flan-T5-xxl. Finetuning improves 0.07 macro-F1 on average over Flan-T5-xxl. This shows that finetuning is overall beneficial in training the model on the task of entailment verification.

Next, we observe that compared to classification, the ranking formulation is beneficial for the contextual QA datasets CosQA, SIQA, and DREAM. This demonstrates that the ranking objective improves contextual QA datasets' generalization, which is expected. Additionally, our ranking model outperforms GPT-3.5 and performs comparably to GPT-4, with stronger performance on contextual QA datasets and weaker performance on NLI datasets. Thus, Flan-T5-xxl + *Rank* is a strong *open-sourced* model for entailment verification and can be used as an alternative to GPT-4.

5 Application: Filtering CoT Rationales

Recently, Wang et al. (2023) proposed self-consistency (SC), a decoding technique to improve

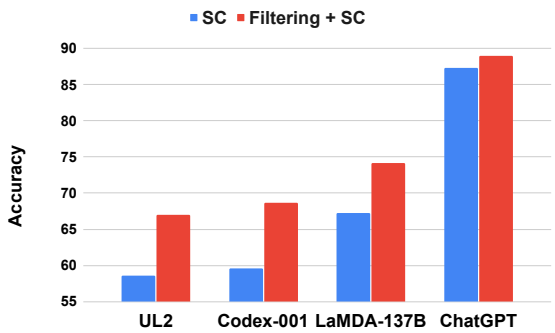

Figure 4: Comparisons between self-consistency (SC) and Filtering + SC. We report the accuracy metric averaged across three MCQ datasets for four different LLMs. **Takeaway**: Filtering consistently improves performance over SC baseline, with more gains for weaker base models such as UL2. Please refer to Section 5 for more details.

over chain-of-though (CoT) reasoning (Wei et al., 2022) in LLMs, whereby multiple CoT rationales are sampled for a given input instance and a majority voting overall predicted labels is considered as the final prediction. However, generative LLMs can potentially output rationales that are inconsistent (Ye and Durrett, 2022), i.e., the rationale does not support the corresponding model prediction. Such inconsistency can, in turn, degrade the overall self-consistency results. Evaluating the consistency between an explanation and the corresponding prediction can be framed as an *entailment verification* (EV) task, as described below. Please refer to Appendix G.2 for examples of consistent and inconsistent CoTs.

Approach As shown in Figure 3, we can use a verifier as an intermediate filtering step to filter out the inconsistent rationales before computing the majority vote. For this, we define the generated CoT rationale as the premise and use the QA-to-statement model (Chen et al., 2021) as defined in Section 2.1 to convert the question and model's

prediction into a hypothesis. Next, we calculate the entailment score of all the premise-hypothesis pairs using a verifier (Equation 1). Finally, we select the top-k rationales for majority voting, discarding the rest. We set $k = 5$ for all our experiments.

Findings In figure 4, we compare the vanilla SC with the filtering+SC approach described above. Following (Wang et al., 2023), we compute the average performance of these methods across three MCQ datasets for four different base CoT models (UL2 (Tay et al., 2023), Codex-001 (Brown et al., 2020), LaMDA-137B (Thoppilan et al., 2022), and ChatGPT[4] (OpenAI, 2022)). Please refer to Appendix G for details on the datasets and more comparisons with Flan-T5-xxl. We observe that filtering leads to a consistent performance gain over SC across all CoT base models. This demonstrates the advantage of the filtering approach. Next, we find that the improvements are more prominent for weaker base models such as UL2 than the stronger ones (ChatGPT). For instance, filtering UL2 generated-rationales can even achieve comparable performance with vanilla SC over LaMDA-137B. In comparison, the gains for filtering ChatGPT CoTs are ~ 1.7 %. This shows that weaker models are prone to generating inconsistent CoTs and thus benefit more from this approach. But at the same time, even stronger models such as ChatGPT can still benefit from consistency checks. Please refer to Appendix G.2 for examples of filtered CoT rationales.

6 Related Works

Natural Language Inference NLI (Dagan et al., 2006; Manning and MacCartney, 2009) is one of the core NLP problems in which the relationship between a premise and hypothesis is classified as either entailment, contradiction, or neutral. Prior works have mainly trained LLMs and evaluated them on standard NLI datasets (Bowman et al., 2015; Williams et al., 2018; Wang et al., 2019; Nie et al., 2020, 2019; Liu et al., 2022). Another line of work (Mishra et al., 2020; Chen et al., 2021) has used question-answer-to-NLI conversion (Demszky et al., 2018) to transform QA datasets into NLI format and solve them. In fact verification literature (Bekoulis et al., 2021; Thorne et al., 2018), retrieved pieces of evidence have been used to verify the claim using an entailment verifier (Nie et al., 2019; Guan et al., 2023). Recently, NLI models have been used to verify the entailment of model-generated explanations (Tafjord et al., 2022; Jung et al., 2022; Mitchell et al., 2022). In this work, we curate a diverse NLI benchmark for evaluating LLMs and humans by using datasets from all the above NLI applications.

Reasoning in LLMs With the advent of general-purpose LLMs (Brown et al., 2020; Chung et al., 2022; OpenAI, 2023), many prompting strategies have been proposed to generate a natural language reasoning along with the model's prediction (Wei et al., 2022; Zhou et al., 2023; Yao et al., 2023; Huang and Chang, 2023). Recently, Ye and Durrett (2022) have found that such generations can sometimes be unreliable due to non-factual and inconsistent reasoning, while Huang et al. (2023) have argued that LLMs struggle to self-correct such issues without external feedback.

Prior works have addressed this limitation by oversampling reasoning chains and marginalizing (Wang et al., 2023), using the LLMs itself to recheck their reasoning (Madaan et al., 2023; Miao et al., 2023), leveraging external knowledge source to verify factuality (Zhao et al., 2023), using deterministic solvers to improve faithfulness (Lyu et al., 2023), decomposing the reasoning steps into smaller steps (Ling et al., 2023), etc. While the progress is impressive, some of these are either specialized approaches for math-specific datasets or heavily rely on close-sourced LLMs (GPT-3.5, GPT-4, etc.) for verification. In contrast, here we focus on natural language datasets and develop an open-sourced model for the verification task.

7 Conclusion

We studied the EV problem in the context of LLMs. Specifically, we sourced datasets across three different categories (NLI, contextual QA, and rationales) and analyzed human and LLM performance on the benchmark. We found that LLMs are better than humans in complex reasoning, while humans are more consistent on simpler reasoning tasks. We also explored different training objectives to finetune open-sourced LLMs for EV. Our finetuned model outperforms GPT-3.5, baseline Flan-T5-xxl, and is comparable to GPT-4. Finally, we applied the EV model to filter out inconsistent model-generated CoTs in self-consistency decoding, achieving improvements over the baseline self-consistency approach.

[4]ChatGPT refers to the `gpt-3.5-turbo-0613` model.

Limitations

Even though our work demonstrates exciting results on entailment verification tasks by finetuning LLMs, several limitations can be potentially improved.

Data Processing Our strategy to convert a QA pair into a statement using the QA-to-statement converter model can have errors that can cascade both in the evaluation dataset and our fine-tuned models. Better strategies for this pipeline would help with data quality.

Finetuning We only tried encoder-decoder-based models for finetuning. However, other models with different architectures (like decoder-only) can also be considered in the future. For computing the entailment score in Equation 1, we only considered the probability of "Yes" and "No" tokens in the entire vocabulary. Other alternative expressions like "YES"/"NO", "True"/"False", etc., can also be considered to make the score more robust. Finally, our training objective outputs entailment scores instead of directly generating answers. Answer generation as a training objective can be more robust since it is a stricter objective than our scoring technique.

Potential Risks We using existing datasets, with some post-processing, to train our models. Thus, any issues in the existing dataset in terms of bias, toxicity, etc., can potentially affect our model. Also, since we use a pretrained checkpoint, we also inherit any existing biases in the baseline model. The model is trained to always output scores, even if the data is well outside the training distribution. This is an existing issue with most NLP models and can be mitigated by additional checks for domain shift.

References

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for CommonsenseQA: New Dataset and Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065, Online. Association for Computational Linguistics.

Giannis Bekoulis, Christina Papagiannopoulou, and Nikos Deligiannis. 2021. A review on fact extraction and verification. *ACM Comput. Surv.*, 55(1).

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33.

Timothy J. Buschman, Markus Siegel, Jefferson E. Roy, and Earl K. Miller. 2011. Neural substrates of cognitive capacity limitations. *Proceedings of the National Academy of Sciences*, 108(27):11252–11255.

Jifan Chen, Eunsol Choi, and Greg Durrett. 2021. Can NLI models verify QA systems' predictions? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3841–3854, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge.

Nelson Cowan. 2001. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24(1):87–114.

Kevin Crowston. 2012. Amazon mechanical turk: A research tool for organizations and information systems scholars. In *Shaping the Future of ICT Research. Methods and Approaches*, pages 210–221, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7358–7370, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Alan Garnham. 1989. Inference in language understanding: What, when, why and how. In RAINER DIETRICH and CARL F. GRAUMANN, editors, *Language Processing in Social Context*, volume 54 of *North-Holland Linguistic Series: Linguistic Variations*, pages 153–172. Elsevier.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics (TACL)*.

Jian Guan, Jesse Dodge, David Wadden, Minlie Huang, and Hao Peng. 2023. Language models hallucinate, but may excel at fact verification.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1266–1279, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter

Clark. 2023. Self-refine: Iterative refinement with self-feedback.

Christopher D. Manning and Bill MacCartney. 2009. *Natural language inference*. Stanford University.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Ning Miao, Yee Whye Teh, and Tom Rainforth. 2023. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: A comprehensive review. *ACM Comput. Surv.*, 54(3).

Anshuman Mishra, Dhruvesh Patel, Aparna Vijayakumar, Xiang Li, Pavan Kapanipathi, and Kartik Talamadupula. 2020. Reading comprehension as natural language inference:a semantic analysis. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 12–19, Barcelona, Spain (Online). Association for Computational Linguistics.

Eric Mitchell, Joseph Noh, Siyan Li, Will Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher Manning. 2022. Enhancing self-consistency and performance of pre-trained language models through natural language inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1768, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ha Thanh Nguyen, Randy Goebel, Francesca Toni, Kostas Stathis, and Ken Satoh. 2023. A negation detection assessment of gpts: analysis with the xnot360 dataset.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

OpenAI. 2022. Introducing chatgpt. `https://openai.com/blog/chatgpt`.

OpenAI. 2023. Gpt-4 technical report.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Soumya Sanyal, Zeyi Liao, and Xiang Ren. 2022. RobustLR: A diagnostic benchmark for evaluating logical robustness of deductive reasoners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9614–9631, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. SocialIQA: Commonsense reasoning about social interactions. In *EMNLP*.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2022. Entailer: Answering questions with faithful and truthful chains of reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2078–2093, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. Ul2: Unifying language learning paradigms.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew

Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020a. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020b. Huggingface's transformers: State-of-the-art natural language processing.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models.

Xi Ye and Greg Durrett. 2022. The unreliability of explanations in few-shot prompting for textual reasoning. In *Advances in Neural Information Processing Systems*.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5823–5840, Toronto, Canada. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

A Evaluation Datasets

In this section, we describe the datasets used in our evaluation. We mention some important challenges that make these datasets useful for benchmarking the entailment verification task. Please refer to Table 5 for these datasets' train/dev/test statistics.

Natural Language Inference datasets NLI is an obvious choice of data source as it is a more general case of the entailment verification problem. While converting an NLI dataset for our task, we merge the *neutral* and *contradict* labels to the *not support* label. We use the following NLI datasets in our benchmark:

- **WaNLI** (Liu et al., 2022): This is a new NLI dataset built using worker and AI collaboration. This challenging dataset improves over the existing NLI dataset MultiNLI (Williams et al., 2018). The test split is used when doing the evaluation.
- **FEVER** (Nie et al., 2019): This is a modification of the original FEVER dataset (Thorne et al., 2018) in which the claim is paired with textual evidence from Wikipedia to convert it into an NLI format dataset. This pairing uses existing state-of-the-art evidence extraction systems to find relevant evidence for each claim. Premises in this dataset typically contain multiple sentences, which is one of our focus areas. As the test split is unavailable, we report results on the dev split for evaluation.
- **ANLI** (Nie et al., 2020): This is a large-scale NLI dataset that was collected using an adversarial human-and-model-in-the-loop procedure. Like FEVER, this dataset tests factual knowledge, and the premises typically contain multiple sentences. During evaluation, the test split is considered.

Contextual QA datasets Next, we consider QA datasets where the task is to answer a question based on a given context and some options. We use an off-the-shelf QA-to-statement converter model (Chen et al., 2021) to generate a hypothesis statement for each question option pair. Then, the hypothesis corresponding to the correct choice is marked as "*support*", while the rest are marked as "*not support*" to create the entailment verification dataset. We depict this process in Figure 5. The green box is the valid hypothesis and the red ones are the invalid hypothesis corresponding to the given context. Overall, we include the following datasets from this category:

- **Cosmos QA** (CosQA) (Huang et al., 2019): This dataset contains multiple-choice questions (MCQs) that require an understanding of commonsense-based reading comprehension to answer a question. The key challenge in this dataset is understanding people's everyday narratives described in the context that can have some missing commonsense knowledge that needs to be inferred implicitly. Since the test split is missing, we evaluate models on the dev split instead.
- **SocialIQA** (SIQA) (Sap et al., 2019): Similar to CosQA, this is another MCQ benchmark for commonsense reasoning about social situations that probes emotional and social intelligence in a variety of everyday situations. This dataset has more nuanced commonsense knowledge requirements, which makes it a challenging dataset for our task. Similarly, results on the dev split are reported, given that the test is missing.
- **DREAM** (Sun et al., 2019): This is a dialogue-based reading comprehension MCQ dataset that focuses on multi-turn dialogue understanding. Here, the unique challenge is inferring the events discussed across long, multi-turn dialogues. During evaluation, we use the test split as it is available.
- **BoolQ** (Clark et al., 2019): This is a True/False QA dataset consisting of aggregated queries to the Google search engine. Questions in this dataset require complex and difficult entailment-like inference to solve, making it a good set for evaluation. The test split is lacking for this dataset, and we can only report results on the dev split.
- **RACE** (Lai et al., 2017): The reading comprehension dataset from examinations (RACE) is one of the most popular machine reading comprehension datasets containing questions from English exams for middle and high school students. These questions are designed by domain experts for testing specific human reading skills, thus making it a good evaluation set for our task. We report evaluating results on test split for this dataset.

Rationale datasets Lastly, we consider data sources where human-annotated explanations are

Dataset Statistics	NLI			Contextual QA					Rationale	
	WaNLI	FEVER	ANLI	CosQA	SIQA	DREAM	BoolQ	RACE	Entailer	ECQA
Train	102,885	208,346	162,865	77,468	100,212	18,348	18,854	341,412	-	7,598
Dev	-	19,998	3,200	8,970	5,859	6,120	6,540	18,944	7,849	1,090
Test	5,000	-	3,200	-	-	6,123	-	19,172	-	2,194

Table 5: The number of examples in train/dev/test splits for different datasets. Some datasets do not have certain splits, and those statistics are left blank. For each dataset in our benchmark, we use the test split if available; otherwise, we use the dev split. Please refer to Appendix A for more details about each dataset.

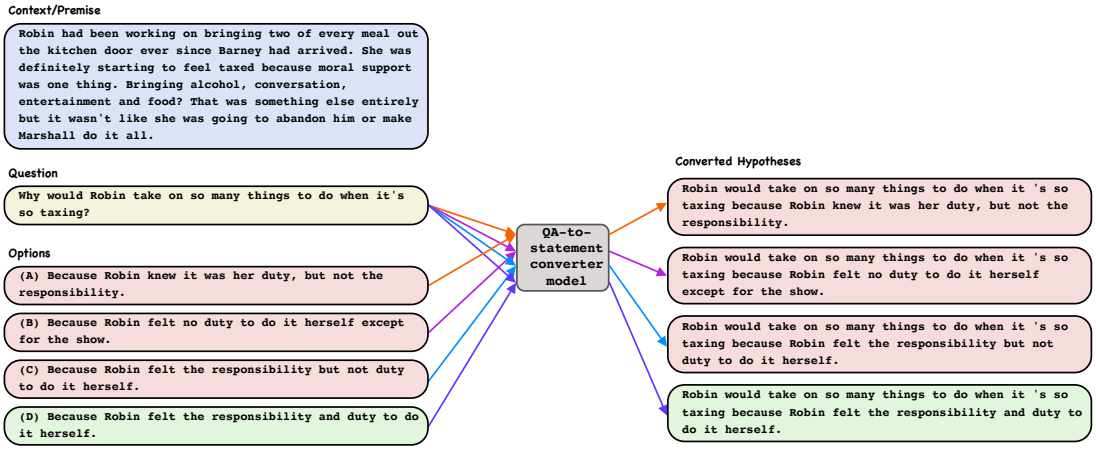

Figure 5: **Example of Contextual QA data conversion.** For a given question and corresponding option, the QA-to-statement converter model generates a sentence combining the two. We use this as the hypothesis corresponding to that option. Please refer to Appendix A for more details.

available that justify the original hypothesis (or the correct option, in the case of QA datasets). In this case, we use the rationales as the premise. We use the following datasets:

- **Entailer** (Tafjord et al., 2022): This dataset contains entailment-style statements and corresponding rationales obtained from EntailmentBank dataset (Dalvi et al., 2021) and crowdsourcing. The dataset mainly contains science domain statements and tests simple deductive reasoning skills whereby sentences from the premise must be combined to either support or refute the hypothesis. The test split is also missing for this dataset, and we can only evaluate it on the dev split.

- **ECQA** (Aggarwal et al., 2021): This is a human-annotated explanation dataset for CommonsenseQA (Talmor et al., 2019). We only use the explanations for the correct choice as the explanations for the incorrect choices are often trivial. It is a complete dataset and by convention, we use the test split for evaluation.

B Details on LLM Evaluation

We evaluate two types of LLMs on the task of entailment verification, as categorized below:

B.1 Task-Finetuned models

In this category, the models considered are already finetuned for either NLI or the exact entailment verification task itself. We evaluate two models in this category.

RoBERTa (Nie et al., 2020; Liu et al., 2019) This is a strong pre-trained RoBERTa-Large model with the corresponding model card on HuggingFace[5] (Wolf et al., 2020b) called "ynie/roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli". It is a specifically pre-trained RoBERTa-Large for NLI task and includes the combination of SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), FEVER (Nie et al., 2019) and ANLI (Nie et al., 2020) datasets as the training data. Hence, this incurs a potential data leakage problem, as we also test it on FEVER and ANLI. To some extent, it explains the strong performance of RoBERTa on NLI datasets in Table 3.

[5] https://huggingface.co/models

The model is used as a classifier in evaluation, and three classes are available. Class 0 corresponds to "*entail*", class 1 corresponds to "*neutral*" and class 2 means "*not entail*". In our experiment setting, we only regard class 0 as the "*Yes*" label and combine the remaining two classes to be the "*No*" label.

Entailer (Tafjord et al., 2022) Entailer is a T5-based model (Raffel et al., 2020) trained to answer hypotheses by building proof trees containing chains of reasoning. It can either generate valid premises for a given hypothesis or predict a score for a given premise and hypothesis. We evaluate Entailer-11B (model name "allenai/entailer-11b" on HuggingFace) in our experiments. Similarly, Entailer dataset (Dalvi et al., 2021) is in the training set, making this model very competitive when evaluating the same dataset.

We strictly follow the official implementation of the model to acquire class labels.[6] The "entailment_verifier" is called to decide if the hypothesis can be implied from the premise. If the answer is "*True*", then the class label will be "*Yes*" and vice versa.

B.2 Instruction-Finetuned models

These are the more recent "general-purpose" language models trained on a collection of NLP tasks described using instructions, leading to generalization abilities to solve unseen tasks described using new instructions. The models included in our evaluation from this category are described below.

Flan-T5-xxl (Chung et al., 2022) It is instruction-tuned from T5 (Raffel et al., 2020) on 1.8K+ tasks. We adopt a publicly available version on HuggingFace (Wolf et al., 2020a) with model card name "google/flan-t5-xxl". Flan-T5-xxl is also exposed to data leakage issues. BoolQ (Clark et al., 2019), ECQA (Aggarwal et al., 2021), and ANLI (Nie et al., 2020) have appeared in its training data. However, this is not a serious problem in the fine-tuning stage because we transform original datasets into entailment verification format before using them for model finetuning.

We extract labels from the model by focusing on the output probabilities of two words, "*Yes*" and "*No*". After applying the softmax function to those two probabilities, we finalize the label as the word with a probability larger than a given threshold, as described in Equation 1.

GPT-3.5 (Brown et al., 2020) It is a general-purpose autoregressive decoder-only LLM accessible via the OpenAI Completions API.[7] We utilize "text-davinci-003" in OpenAI's API for evaluation, and the label determination procedure is similar to the one in Flan-T5-xxl.

GPT-4 (OpenAI, 2023) It is the latest generative model published by OpenAI, which is optimized for creativity and long context inputs. It is accessible via the OpenAI Chat API.[8] We adopt plain "gpt-4" in OpenAI's API for our experiments. Unlike other models, the output probabilities are not accessible. Thus, we constrain the model to predict a single token as its generated prediction. In the ideal case, we can directly use the output text as the label if "*Yes*" or "*No*" is produced. If some other token is generated, we choose one of the labels at random. We note that this occurrence is rare in our GPT-4 evaluation runs.

B.2.1 Prompt Robustness Evaluation

To assess the robustness of the model in section 2.3, we design different prompt formats but hold the order of premise and hypothesis in the prompt unchanged. Table 6 presents all prompts we tested with Flan-T5-xxl and their corresponding averaged results across datasets. The results suggest that the Flan-T5-xxl model is robust to the variation in the prompt and yields relatively consistent results. This characteristic is maintained in Flan-T5-xxl + *Class* and Flan-T5-xxl + *Rank* as well since they are developed over based on Flan-T5-xxl.

B.2.2 Few-Shot Evaluation

Few-shot is an effective and promising strategy when testing the performance of a model (Brown et al., 2020). We also include this analysis by randomly picking two examples from Entailer as demonstrations and incorporating them into the prompt. We test Flan-T5-xxl, Flan-T5-xxl + *Rank*, and GPT-4 in this setting and represent results in Table 7. The few-shot setting yields promising improvement for Flan-T5-xxl, substantiating that the few-shot is a beneficial approach to teaching the prompt to the model. However, it is not as helpful

[6] https://github.com/allenai/entailment_bank/blob/main/entailer.md

[7] https://platform.openai.com/docs/api-reference/completions

[8] https://platform.openai.com/docs/api-reference/chat

Prompt	Avg
Premise: {premise}\n Hypothesis: {hypothesis}\n Given the premise, is the hypothesis correct?\n Answer:	0.71
Premise: {premise}\n Hypothesis: {hypothesis}\n Given the premise, is the hypothesis supported?\n Answer:	0.70
Premise: {premise}\n Hypothesis: {hypothesis}\n Based on the premise, is the hypothesis correct?\n Answer:	0.71
Premise: {premise}\n Hypothesis: {hypothesis}\n Does the premise support the hypothesis?\n Answer:	0.69
Given the premise {premise}, is the hypothesis {hypothesis} supported?\n Answer:	0.70
We are given the premise: {premise}. Can we conclude the hypothesis: {hypothesis}?\n Answer:	0.72

Table 6: Comparison of averaged results between different prompt formats used for Flan-T5-xxl evaluation. **Takeaway**: The model is robust to variations in the prompt and generates consistent results. Please refer to Appendix B.2 for more details.

Model	NLI			Contextual QA					Rationale		Avg
	WaNLI	FEVER	ANLI	CosQA	SIQA	DREAM	BoolQ	RACE	Entailer	ECQA	
GPT-4	0.73	0.88	0.86	0.79	0.69	0.92	0.86	0.85	0.86	0.48	0.79
GPT-4 + *few-shot*	0.75	0.87	0.84	0.75	0.62	0.91	0.86	0.81	0.83	0.48	0.77
Flan-T5-xxl	0.63	0.81	0.73	0.59	0.67	0.80	0.85	0.70	0.83	0.50	0.71
Flan-T5-xxl + *few-shot*	0.67	0.84	0.77	0.66	0.71	0.84	0.84	0.74	0.85	0.49	0.74
Flan-T5-xxl + *Rank*	0.69	0.85	0.77	0.83	0.74	0.89	0.85	0.85	0.86	0.48	0.78
Flan-T5-xxl + *Rank* + *few-shot*	0.69	0.86	0.79	0.77	0.74	0.89	0.84	0.85	0.86	0.48	0.78

Table 7: Comparison of performance between base models and models with few-shot setting. **Takeaway**: For Flan-T5-xxl, few-shot prompting boosts the performance, while it is not beneficial for the other two models. Please refer to Appendix B.2 for more analysis.

Dataset	Majority Prediction
WaNLI	0.39
FEVER	0.40
ANLI	0.40
CosQA	0.41
SIQA	0.40
DREAM	0.40
BoolQ	0.35
RACE	0.43
Entailer	0.43
ECQA	1.00

Table 8: The macro-F1 score of majority label (most frequent label) prediction for different datasets. For reference, the score of a well-balanced dataset is 0.67. Those figures indicate that the label imbalance issue exists in the datasets we evaluate. More details are presented in Appendix C.

as our finetuning strategies, which give even better performance. On the other hand, few-shot does not bring significant gains for Flan-T5-xxl + *Rank*, suggesting that finetuning has already helped the model have a comprehensive understanding of the prompt, and extra demonstrations are unnecessary. As for the GPT-4, simply using examples from Entailer and applying the same prompt for all datasets seem detrimental.

C Majority Prediction and Label Imbalance

In Table 8, we show the performance of an oracle model that predicts the most frequent label in a dataset. For a label-balanced dataset, the macro-F1 score of such a majority prediction model would be 0.67 (precision 0.5 and recall 1.0). The datasets in the evaluation set have some label imbalance, as evidenced by the lower majority label prediction scores. Since we convert existing 3-class NLI and multi-choice QA datasets into our binary classification task format, it inherently has more *not support* labels. We have more *support* instances for the rationale datasets since the dataset creators usually only annotate the rationale for the right choice. Specifically, the ECQA dataset only has positive instances, leading to a 1.0 macro-F1 score for majority prediction (*support* label). Since it has all *support* labels, any model predicting even a single *non support* label gets penalized severely, as is seen in ECQA results. Because of this label imbalance in the datasets, we report the macro-F1 scores instead of accuracy or micro-F1 metrics.

D Debatable Cases in Contextual QA

We include two examples of SIQA dataset in Table 9 to illustrate the debatable hypotheses in Contex-

tual QA datasets. These examples demonstrate that some less probable hypotheses can also be considered plausible in certain situations. In the first example, option "run screaming" best describes the audience's reactions after listening to a scary ghost story. However, there are also situations where people are so terrified and start to cry, making the option "cry" somehow debatable. Similarly, in the second example, it is common sense that people will express gratefulness after someone else builds a house for them. But it is reasonable to say that Quinn is intrigued by the house after seeing the amazing architecture. In both cases, apart from the most adequate one, there are still options leading to some partially supported hypotheses, which explains the low human annotation agreement in these cases.

E Human Evaluation and Analysis

We adopted Amazon Mechanical Turk (MTurk) (Crowston, 2012) for data collection. Two annotation formats (Figure 7 and Figure 8) were devised for the human evaluation task and reasoning type analysis task, respectively. During the annotations, each annotator was compensated according to $15/hour per the U.S. minimum wage.

E.1 Human Evaluation Details

In each HIT, the annotator was presented with a format exactly like Figure 7, including detailed task descriptions and label explanations. Annotators were expected to read the premise and claim first, then determine the supportiveness of the claim based on the premise and choose the corresponding label. Initially, we only provided three labels — "*support*", "*irrelevant*" and "*contradict*". But later, we realized that annotators could not explicitly identify labels for some ambiguous instances where the premise only partially supported or contradicted the claim. Hence, we introduced two weak labels ("*partially support*" and "*partially contradict*") to remedy this issue. When collating results for analysis, we internally combined "*support*" and "*partially support*" to be "*support*", and the rest to be "*not support*", aligning to the standard entailment verification (EV) setup. Each instance was annotated by 3 MTurk annotators, and a majority verdict determined the label. The Fleiss's kappa score (Fleiss, 1971) we got was 0.6, indicating a moderate level of agreement among annotators.

E.2 Reasoning Type Annotation Details

We annotate the reasoning type of the 100 sampled instances for each dataset with absolute macro-F1 difference > 0.1 in Table 3. We make this choice intending to attribute the largely misaligned datasets (namely, ANLI, CosQA, SIQA, and Entailer) since some random noise in the annotation and sampling process can potentially also cause some misalignment.

In every HIT, we used Figure 8 as the reasoning type annotation format. The instructions and label explanations were explicitly stated at the beginning of the HIT. For task 1, after reading the premise and the claim, annotators had to decide whether the supportiveness of the claim could be decided by only referring to the information in the premise. If yes, annotators needed to choose the corresponding difficulty level of reasoning about the supportiveness of the claim in task 2. Otherwise, the type of missing information had to be decided in task 1, and task 2 did not apply to these cases. We aggregated answers from the two tasks and categorized them into four types. Both **R1** and **R2** were types where the premise contained all necessary information. The difference was that the difficulty level of reasoning for **R1** was *Easy* while for **R2** was *Moderate*. We combined *Missing Entity-grounded Information* and *Missing Commonsense Information* to be **R3**. Finally, the *Missing Localized Information* label corresponded to **R4**.

Given that some strong NLP background knowledge was required to understand task descriptions and label explanations, we recruited Computer Science graduate students instead of the general mTurk workers to finish this annotation. Every instance was assigned to 2 students, and the majority vote determined the label. The Fleiss's kappa score for this job was 0.62, showing a substantial inter-annotator agreement.

E.3 Reasoning Type Examples

Table 11 incorporates three examples from each reasoning type, providing more insight into those types.

In the third example of **R1**, the first sentence in the premise states that "*iron oxide*" comes from "*oxygen*" and "*rust*". The second sentence shows those two substances are "*gases*" at room temperature. Therefore, combining them will be sufficient to entail the hypothesis.

However, unlike **R1**, the third example of **R2**

Question & Options	Corresponding Hypothesis of Option
Alex told a very scary ghost store to the campers around the campfire. What will Skylar want to do next?	
(A) run screaming	**Skylar will want to run screaming next.**
(B) laugh it all	Skylar will want to laugh it all next.
(C) cry	Skylar will want to cry next.
After finalizing the plans with the architect and the contractors, Austin built Quinn's house last week. How would Quinn feel as a result?	
(A) intrigued by the building of the house	Quinn would feel intrigued by the building of the house as a result.
(B) grateful for the hard work that was done	**Quinn would feel grateful for the hard work that was done as a result.**
(C) curious about why the house was built	Quinn would feel curious about why the house was built as a result.

Table 9: Examples from Contextual QA datasets with a debatable hypothesis. The correct answers are marked. More analysis can be found in Appendix D.

requires three steps of reasoning that "De Baandert was a multi-use stadium.", "It was mostly used for football matches.", and "The stadium was able to hold 22,000 people.". The hypothesis can be disproved with those steps because "22,000 people" is just the maximum capacity.

As for the first example of **R3**, some missing commonsense information like "*it is not wise to give more money to a person who keeps playing in a detrimental situation.*" should be combined with the premise to disprove the hypothesis.

In the first example of **R4**, the next movement of "*Addison*" is the missing information specific to the context depicted by the premise. The hypothesis can not be directly disproved without that piece of information.

E.4 Reasoning Type Analysis

We depict the aggregated results of the reasoning type annotation for each dataset in Figure 6. Here, the first row shows the frequency of each reasoning type in that dataset, and the corresponding plot in the second row compares the human and GPT-4 macro-F1 scores. We fade out the columns with data percentages less than 5% because such low frequency might not lead to conclusive observations.

We note that type **R1** is most prominent in Entailer dataset. Here, we observe that humans are significantly better than models. This shows that humans are usually more consistent with simple deductive reasoning. Similar findings about consistency in human deductive reasoning skills have been reported in prior works (Sanyal et al., 2022; Nguyen et al., 2023).

Reasoning type **R2**, which requires more complex reasoning, is dominant in ANLI and CosQA. For this type, we find that models are superior to humans. Complex reasoning requires two skills: understanding multiple relevant information in the premise and then using them for reasoning. We hypothesize that models are stronger context processors than humans because they have been trained on long-context data (OpenAI, 2023).

The reasoning type **R3** is present in ANLI, CosQA, and SIQA. From Table 1, we know that ANLI mostly require entity-grounded knowledge, whereas CosQA and SIQA specifically test commonsense knowledge. Here, we find that humans are stronger than models in commonsense knowledge (CosQA and SIQA), whereas models are better in ANLI that requires entity-grounded knowledge. This shows that humans can infer missing social/commonsense knowledge more easily since these are inherently known to humans. In contrast, models can retrieve the entity-grounded knowledge stored in their parameters more efficiently.

Lastly, we find that the reasoning type **R4**, indicating missing localized knowledge, is also promi-

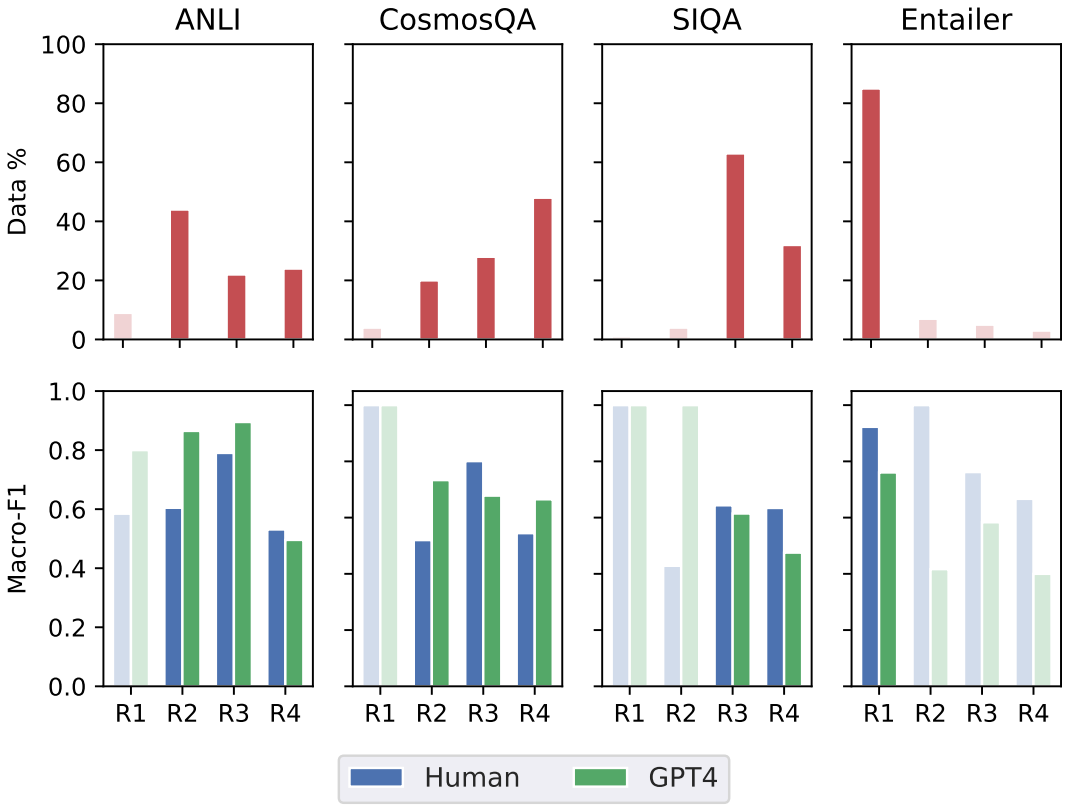

Figure 6: Analysis of different reasoning types involved in entailment verification for each dataset. **[Top]** Distribution of the reasoning types for each dataset studied. **[Bottom]** Macro-F1 performance comparison between humans and GPT-4. We fade out all bars with distribution percentage $\leq 10\%$ since they are insignificant to draw meaningful conclusions. **Takeaways**: Humans are better at simple reasoning (**R1**) and commonsense reasoning (**R3**). GPT-4 is superior at complex reasoning (**R2**) and entity-grounded reasoning (**R3** ANLI). Trends for **R4** are mixed. Please refer to Appendix E.4 for details.

nent in ANLI, CosQA, and SIQA. Here, we find that the trends are a bit mixed. We find that for SIQA, humans are better at recognizing missing localized knowledge, but in CosQA, models outperform humans. This is likely because SIQA typically contains short contexts based on everyday social situations that are easier for humans to understand. In contrast, CosQA has longer contexts with rarer situations requiring more complex understanding.

Overall, we conclude that GPT-4 outperforms humans in complex deductive reasoning and situations involving entity-grounded knowledge, whereas humans are more consistent at simple reasoning and situations requiring commonsense knowledge.

F Finetuning LLMs

In this section, we describe more details about the training dataset used for finetuning, our negative data collection strategy that was used in the ranking formulation, and other finetuning details.

F.1 Training Dataset Selection

To train the Flan-T5-xxl model, we create a training dataset using representative datasets from each category. We pick the ANLI, RACE, and ECQA datasets to represent NLI, contextual QA, and rationale categories, respectively. We select the datasets with diverse entailment challenges and aim to maximize the total training data. We note that the amount of training data is quite low for the rationale category. Thus, we also include the StrategyQA (Geva et al., 2021) dataset in the training set to alleviate this. Similar to BoolQ, StrategyQA has a Yes/No type of questions and their corresponding explanations. We convert the question-answer pair into a hypothesis using the QA-to-statement converter (Chen et al., 2021) as described in Section 2.1.

F.2 Negative Data Collection for Ranking

In the ranking formulation, for a given premise and hypothesis pair (p, h), we need to find some weaker hypothesis h' to use the ranking loss defined by Equation 2. We collect such weaker hypotheses

Method	CSQA	ARC-e	ARC-c	Average
UL2 + SC	55.77	70.33	49.57	58.56
UL2 + Filter (Flan-T5-xxl) + SC	62.49	74.50	55.72	64.24
UL2 + Filter (Flan-T5-xxl + *Rank*) + SC	63.80	77.24	60.08	67.04
Codex-001 + SC	54.80	71.70	52.20	59.57
Codex-001 + Filter (Flan-T5-xxl) + SC	65.44	73.57	54.27	64.43
Codex-001 + Filter (Flan-T5-xxl + *Rank*) + SC	68.39	78.16	59.51	68.69
LaMDA-137B + SC	62.90	78.90	59.90	67.23
LaMDA-137B + Filter (Flan-T5-xxl) + SC	71.17	80.89	63.65	71.90
LaMDA-137B + Filter (Flan-T5-xxl + *Rank*) + SC	72.15	82.95	67.32	74.14
ChatGPT + SC	78.40	96.30	87.20	87.30
ChatGPT + Filter (Flan-T5-xxl) + SC	81.86	96.52	88.32	88.90
ChatGPT + Filter (Flan-T5-xxl + *Rank*) + SC	81.88	96.57	88.63	89.03

Table 10: Comparison of Chain-of-Thought filtering performance. We consider four self-consistency baselines. For each baseline, we experiment with both Flan-T5-xxl and Flan-T5-xxl + *Rank* to filter out inconsistent rationals. **Takeaway**: Our fintuning strategy brings notable improvements when compared to both baseline and Flan-T5-xxl filtering model. For more details and analysis, please check Appendix G.

For a given premise and a valid hypothesis, generate five alternate hypotheses contradicted by the premise. Try to avoid using the negation words such as "not", "never", etc. The output should be numbered from 1 to 5.
Premise: {*premise*}
Hypothesis: {*hypothesis}*

Box 2: Prompt format for generating alternate negative hypothesis for a given premise-hypothesis pair. Please refer to Section F.2 for details.

in two ways and then combine them to form the training data. The two techniques are described below:

- **Using incorrect options**: The contextual QA category has naturally occurring negative data. For a given question and choices, we pair the hypothesis corresponding to the correct option with all other hypotheses corresponding to the wrong options to create the ranked data.

- **GPT-3.5 prompting**: The other way we generate negative data is by prompting GPT-3.5. Specifically, we use the prompt format shown in Box 2 to generate alternate hypotheses contradicted by the original premise. We only select premise and hypothesis pairs that originally have *support* label. GPT-3.5 generated hypotheses are then considered negative samples and paired with the original hypothesis. We repeat this for all the training datasets (ANLI, RACE, ECQA, and StrategyQA).

F.3 Hyperparameters and other details

During training, we select the learning rate from the set $\{7e^{-5}, 1e^{-4}, 2e^{-4}\}$, per GPU batch size from the set $\{6, 8\}$, margin m in Equation 2 from the set $\{0.2, 0.3, 0.5\}$, and warmup ratio 0.1. The model is trained for 1400 steps on a cluster of 8 A6000 GPUs. We evaluate the model every 200 steps and save the checkpoint if the model shows improvements on a held-out development set.

G Chain-of-Thought Filtering

We study three variants of CoT Filtering as mentioned below:

- $\mathcal{B}$ + SC: This is the self-consistency baseline. Here, $\mathcal{B}$ is the base model used to sample CoTs. We sample 40 CoTs for each instance before computing the majority predicted label.

- $\mathcal{B}$ + Flan-T5-xxl + SC: In this, we use a pretrained Flan-T5-xxl for filtering out the inconsistent rationales before the majority voting. We keep the top-5 rationales after scoring them using Flan-T5-xxl.

- $\mathcal{B}$ + Flan-T5-xxl + *Rank* + SC: This is the same as above, but instead, we use our ranking-finetuned Flan-T5-xxl model for filtering.

Following (Wang et al., 2023), we use four different base CoT model: UL2 (Tay et al., 2023), Codex-001 (Brown et al., 2020), LaMDA-137B (Thoppilan et al., 2022), and ChatGPT (OpenAI, 2022). Further, we compute the CoTs and analyze the performance of the above methods for

three multi-choice QA datasets, namely, CommonsenseQA (Talmor et al., 2019) and AI2 Reasoning Challenge (Clark et al., 2018) (easy (ARC-e) and challenge (ARC-c) variants). Please refer to (Wang et al., 2023; Wei et al., 2022) and the associated code[9] for details on the CoT prompt formats. The results are shown in Table 10. We note a consistent improvement between the three variants, with Flan-T5-xxl + *Rank* model performing the best. This demonstrates the advantage of our entailment fine-tuning approach. Please refer to Section 5 for more findings.

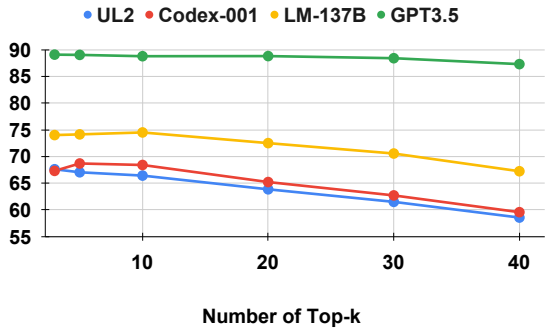

Figure 9: Results comparison among different k values in CoT filtering application. $k = 5$ is the best choice on average. For detailed analysis, please refer to Appendix G.1.

G.1 Ablation of Top-k Filtering

We examine our top-k selecting strategy by choosing different values of k and compare their results. We pick up k from $\{3, 5, 10, 20, 30\}$ and present the corresponding aggregated results in Figure 9. The general trend is that, as k increases, the model performance increases first and then starts to decrease, reaching the peak at $k = 5$. When k is small, there are only a limited number of CoT rationals, and the majority voting process is vulnerable to potential outliers. On the other hand, if k is too large, then many noisy results are included, leading to poorer performance.

G.2 Examples of Filtered CoTs

Table 12 presents three CoT reasoning examples, each including two outputs that are kept and three that are filtered out by ranking. According to the table, outputs supported by strong rationales are ranked highly and kept. On the other hand, if the rationale is irrelevant to the prediction (like rationale 3 in example 1), the rationale itself is incomplete (like rationale 5 in example 1), or the rationale supports another option rather than the prediction (like rationale 4 in the example 1), then such output has a low entailment score leading to a lower ranking and getting filtered out.

[9] https://openreview.net/attachment?id=1PL1NIMMrw&name=supplementary_material

Instructions

Thanks for participating in this HIT!

You will read a claim and an premise which may or may not support the claim.

Premise	A few sentences describing some knowledge behind the topic of the claim.
Claim	A simple sentence describing an event, situation, fact, etc., that essentially makes a claim.

The task asks you to determine the relationship between the *Claim* and the *Premise*. Below are some important definitions. Please read the label descriptions and choose accordingly.

Support	Premise supports the claim basically means the premise provides all necessary information to explain why the claim is valid.
Partial Support	Premise partially supports the claim if the premise provides some information to explain why the claim is valid, but its missing some more information that may be required to confidently say its a valid claim.
Irrelevant/Out-of-topic	These are cases where the premise is not related/relevant to the claim or contains redundant information that is totally not helpful. Note that if the premise supports the claim but still contains redundant sentences, this label should not be selected.
Partial Contradict	Premise partially contradicts the claim if the premise indicates why a part of the claim might be wrong, but more information is needed to be confident about it. This might occur very rarely.
Contradict	Premise contradicts the claim if the information provided in the premise proves the opposite of what is claimed in the claim.

A couple of notes:

- **You may disagree with the correctness/factuality of the *Claim* or the *Premise*. Please assume they are correct and focus only on the relation between them.**
- **The premise usually contains multiple sentences some of which might be redundant. Please ignore the redundant sentences when judging the relation between the *Premise* and the *Claim*, i.e., it's okay to have reduntant sentences in the premise as long as the claim is supported by the premise.**

Example

Example #1:

Premise: **A fried egg is a cooked dish made from one or more eggs which are removed from their shells and placed into a pan, usually without breaking the yolk, and fried with minimal accompaniment. Fried eggs are traditionally eaten for breakfast in many countries but may also be served at other times of the day.**
Claim: **A fried egg has a runny yolk.**

Task: Does the *Premise* support the *Claim*?

- ○ Yes, the Premise **fully supports** the Claim.
- ◉ Yes, but the Premise only **partially supports** the Claim.
- ○ No, the Premise only contains **irrelevant information/out-of-topic** sentences or is not well-formed.
- ○ No, the Premise **partially contradicts** the Claim.
- ○ No, the Premise **fully contradicts** the Claim.

Justification: The premise just mentions that the yolk is not broken, but mentions nothing about it being runny or not.

[Examples 2 and 3 ommited here for brevity]

Figure 7: **Human Evaluation Format.** We use this format to evaluate human performance on the Entailment Verification (EV) task. Please refer to Appendix E.1 for more details about the annotation procedure.

Instructions

Thanks for participating in this HIT!

You will read a claim and an premise. The claim will either be supported or not supported by the premise. Your task is to determine if any information is missing in the premise when determining the supportiveness of the claim and how easy it is to conclude the supportiveness of the claim from the given premise.

Premise	A few sentences describing some knowledge behind the topic of the claim.
Claim	A simple sentence describing an event, situation, fact, etc., that essentially makes a claim.

Task1 **asks you to determine if the** ***Premise*** **presents all necessary information to reason about the supportiveness of the** ***Claim*****. If some information is missing, below are three possible types of missing information considered. Please understand the distinction between each and label accordingly.**

Missing Entity-grounded Information	Some information is missing in the premise. Those information is likely to be found on WikiPedia and general internet.
Missing Commonsense Information	Some information is missing in the premise. Those information is implicitly understood amongst humans, unlikely to be documented on the web.
Missing Localized Information	Some information is missing in the premise. Those information is about specific person/event/item in the context.

Task2 **asks you to determine how easy it is to reason about the supportiveness of the** ***Claim*** **using just the information present in the** ***Premise*** **if all necessary information is presented in the** ***Premise*****. Please read the label descriptions and choose accordingly.**

Easy	The reasoning is easy if minimally combining/substituting sentence in premise or combining sentences in premise along with some english word knowledge of negations, synonyms, antonyms, etc. will prove/disprove the claim.
Moderate	The reasoning is moderate if the premise contains all information needed to prove/disprove the claim but multiple reasoning steps are needed.
N/A	There is some information missing in the premise and this question is not applicable to that instance.

A couple of notes:

- **You may disagree with the correctness/factuality of the** ***Claim*** **or the** ***Premise*****. Please assume they are correct and focus only on the relationship and reasoning between them.**
- **The premise usually contains multiple sentences some of which might be redundant. Please ignore the redundant sentences when judging the relation between the** ***Premise*** **and the** ***Claim*****, i.e., it's okay to have reduntant sentences in the premise as long as the claim is supported by the premise.**

Example

Example #1:

Premise: **Inheriting is when an inherited characteristic is passed from parent to offspring by genetics / DNA. Inherited characteristics are the opposite of learned characteristics.**
Claim: **Learned characteristics are not inherited from parents.**

Task 1: Does the ***Premise*** **contain all information needed to convincingly support/refute** ***Claim*****?**

- (●) Yes
- ( ) No, missing some entity-grounded information
- ( ) No, missing some commonsense information
- ( ) No, missing some localized information

Justification: The premise clearly states that the inherited characteristics are from parents and clarifies the relationship between inherited characteristic and learned characteristic. Those information is enough to determine the supportiveness of the claim.

Task 2: If all needed information is contained, then how easy would it be to reason about the supportiveness of the ***Claim*** **based on just the** ***Premise*****?**

(●)	( )	( )
Easy	Moderate	N/A
(combine/substitute sentences or use word knowledge)	(multiple reasoning steps required)	(not applicable)

Justification: Simply understanding "opposite" in the premise has the similar meaning to "not" will be enough to prove the claim.

[Examples 2 and 3 ommited here for brevity]

Figure 8: **Reasoning Type Annotation Format.** This format collects the reasoning type of sampled instances from each dataset. The detailed annotation procedure can be found in Append E.2.

Type	Example	Entails	Human	GPT4
[R1]	**Premise**: Seoul Train is a 2004 documentary film that deals with the dangerous journeys of North Korean defectors fleeing through or to China. These journeys are both dangerous and daring, since if caught, they face forced repatriation, torture, and possible execution. **Hypothesis**: Seoul Train was filmed in 2002 to depict the dangerous journey of North Korea.	No	✗	✓
	Premise: My family history goes back a long way. My ancestors on my mothers side were a mix of English and Scandinavian Mormon converts that came to Utah in the 19th century. My father side is an unknown. **Hypothesis**: It might be true that your family history has a short history.	No	✓	✓
	Premise: An iron oxide can be made from oxygen and rust. Oxygen and rust are gases at room temperature. **Hypothesis**: An iron oxide can be made from two elements that are gases at room temperature.	Yes	✓	✗
[R2]	**Premise**: How to make deep fried watermelon. Cut the watermelon in half, down its length. Then cut each half in half, again cutting down the length. Place the four wedges on a board for cutting. **Hypothesis**: To deep fry a watermelon, it should be cut into 6 pieces.	No	✗	✓
	Premise: The freshwater mussels used to live in the place where the mountain range is located. A freshwater mussel is a kind of water animal that lives in freshwater. **Hypothesis**: The mountain range used to be covered by freshwater.	Yes	✓	✗
	Premise: De Baandert was a multi-use stadium in Sittard-Geleen, Netherlands. It was used mostly for football matches and hosted the home matches of Fortuna Sittard. The stadium was able to hold 22,000 people. It was closed in 1999 when Fortuna Sittard Stadion opened. **Hypothesis**: 22,000 people go to football matches at De Baandert.	No	✗	✓
[R3]	**Premise**: Sasha spent Austin's money trying to win a prize even when the odds were stacked against her. **Hypothesis**: Austin will want to pull out more money next.	No	✗	✓
	Premise: George Dayton (born 1827, died 1938) lived in Union Township in what is now Rutherford, New Jersey, and represented Bergen County in the New Jersey Senate from 1875 to 1877. Dayton moved to Closter, New Jersey, in 1890 and became the clerk of Harrington Township, New Jersey. **Hypothesis**: George Dayton lived in New Jersey his entire life.	No	✗	✗
	Premise: We fought aliens on Mars for Mars Bars, which gave us superspeed. We didn't stop at Jupiter, coz that would have made us "stupider". Pluto was very chilly, so we ate it. Ahem, chille that is. **Hypothesis**: This person will attempt to make a fun rhyme with the names of the other six planets.	Yes	✓	✗
[R4]	**Premise**: Robin put their hand on Addison's wrist to keep Addison from shaking uncontrollably. **Hypothesis**: Addison will want to do good next.	No	✗	✓
	Premise: Terry Pratchett's Going Postal is a two-part television film adaptation of "Going Postal" by Terry Pratchett, adapted by Richard Kurti and Bev Doyle and produced by The Mob, which was first broadcast on Sky1, and in high definition on Sky1 HD, at the end of May 2010. **Hypothesis**: Terry Pratchett's Going Postal adaptation for television is faithful to the original novel.	No	✗	✓
	Premise: Co-producing the film is Happy Madison, the company owned by Sandler, who has been a compatriot of Schneider's since they met early in their careers, as mutually struggling stand-up comedians in Los Angeles. **Hypothesis**: Comedians in Los Angeles struggle.	No	✓	✗

Table 11: Comparison between the performance of humans and GPT-4 on examples from four reasoning types. Each example includes the golden label from the dataset and shows if human/GPT-4 makes the correct decision. The definition of those reasoning types and detailed analysis can be found in Section 3.3 and Appendix E.3.

Question & Options	(Rationale, Prediction)	Kept
[ARC_easy] Cells need nutrients for energy. Which system is responsible for breaking down food to provide cellular energy? **(A)** digestive **(B)** excretory **(C)** circulatory **(D)** respiratory	(Digestive system takes in food, breaks down food, and takes away the unusable parts of the food using the excretory system., **B**)	✓
	(Circulatory system is responsible for eliminating waste from the body. Circulatory system is also responsible for breaking down food into nutrients that the cells can use., **C**)	✓
	(Cellular energy is provided by the chemical breakdown of food., **C**)	✗
	(In the mouth, food is broken down and absorbed into the circulatory system., **D**)	✗
	(Since the digestive system breaks down food to provide cellular energy., **B**)	✗
[CSQA] A human wants to submerge himself in water, what should he use? **(A)** whirlpool bath **(B)** coffee cup **(C)** cup **(D)** soft drink **(E)** puddle	(The answer should be the thing that will help the human submerge himself in water. Of the above choices, only whirlpool baths can help the human submerge himself in water., **A**)	✓
	(The answer should be an item that allows the human to fully submerge himself. Of the above choices, only a whirlpool bath allows the human to fully submerge himself., **A**)	✓
	(The answer should be something that can perform the submerging function. Of the above choices, only the soft drink can float., **D**)	✗
	(The answer must be something that a human would use. Of the above choices, the only thing a human would use is a soft drink., **D**)	✗
	(The answer must be something that is liquid. Of the above choices, only a cup of water is liquid., **C**)	✗
[ARC_challenge] In pea plants, the trait for round seeds is dominant over the trait for wrinkled seeds. If a pure dominant plant with round seeds is crossed with a plant with wrinkled seeds, what can be predicted about the offspring plants? **(A)** They will always produce round seeds. **(B)** They will only produce wrinkled seeds. **(C)** Each offspring plant will have some round and some wrinkled seeds. **(D)** Unless the plant with wrinkled seeds was pure recessive, no prediction can be made.	(The offspring plants will only produce round seeds as the round seed trait is dominant over the wrinkled seed trait., **A**)	✓
	(If plant with round seeds is crossed with plant with wrinkle seeds, the dominant trait will be passed down to the offspring; so each offspring plant will have round seeds., **A**)	✓
	(If a pure dominant plant with round seeds is crossed with a plant with wrinkled seeds, the offspring plant will produce round seeds., **C**)	✗
	(Just like the traits in humans, there is no dominant trait for a particular characteristic., **C**)	✗
	(The recessive trait is the one that causes the plant to produce round seeds. So option Unless the plant with wrinkled seeds was pure recessive, no prediction can be made., **D**)	✗

Table 12: CoT filtering examples from three MCQ datasets. Each example contains two outputs that are kept and three outputs that are filtered out in the entailment verification pipeline. Please refer to Appendix G.2 for more details.