ARE MACHINES BETTER AT SLOW THINKING? UNVEILING HUMAN-MACHINE INFERENCE GAPS IN ENTAILMENT VERIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Humans make numerous inferences in text comprehension to understand the meaning. This paper aims to understand the similarities and differences between humans and state-of-the-art Large Language Models (LLMs) in their ability to judge valid inferences. To this end, we leverage a comprehensively curated entailment verification benchmark that includes datasets from three NLP domains (NLI, contextual QA, and rationales) containing multi-sentence premises and requiring different types of knowledge. Our findings reveal LLMs' superiority in multi-hop reasoning across extended contexts requiring slow thinking, while humans excel in simple deductive reasoning tasks. Using these insights, we introduce a fine-tuned Flan-T5 model that outperforms GPT-3.5 and rivals GPT-4, offering a superior open-source LLM for entailment verification. As a practical application, we showcase the efficacy of our finetuned model in enhancing the self-consistency in model-generated CoT rationales, resulting in a 6% performance boost on average across three multiple-choice question-answering datasets.

1 INTRODUCTION

A prevailing notion in cognitive psychology is the recognition that humans make numerous inferences to understand discourse and text Garnham (1989). These inferences, serving diverse functions in text comprehension, extend beyond resolving lexical ambiguities or determining the referent of a pronoun; they play a crucial role in linking information from disparate sections of a text to establish its literal meaning. 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. Similarly, prior works Gururangan et al. (2018); McCoy et al. (2019); Geiger et al. (2020); Clark et al. (2020); Sanyal et al. (2022) have shown that large language models (LLMs) are still subpar at understanding negations in inference, use spurious correlations while reasoning, cannot reason through multi-step compositions robustly, etc. With the advent of large language models (LLMs) and their applications on natural language inference (NLI) Minaee et al. (2021); Nie et al. (2020), understanding the key difference in the inference abilities between humans and LLMs is crucial to making further progress, which is currently missing.

In this work, we aim to address this by first evaluating both humans and current LLMs Liu et al. (2019); Tafjord et al. (2022); Chung et al. (2022); Brown et al. (2020); OpenAI (2023) on a comprehensive entailment verification benchmark and then analyzing the performance across different reasoning categories based on the ease of inference deduction and the type of knowledge required. 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, complex reasoning. Predicting the entailment of such complex premise-hypothesis pairs often requires multi-hop reasoning, inferring missing information, robustness to spurious correlation, etc., which are largely missing from simpler datasets Gururangan et al. (2018); McCoy et al. (2019). Also, real-world scenarios that require inference skills, such as understanding stories, dialogues, etc., usually contain multi-sentence premises. Therefore, we select multiple datasets across three categories (NLI, contextual QA, and rationales) and convert them into NLI format, as required, to create a more suitable evaluation benchmark. As shown in Table 1, the datasets used in our study typically contain multi-sentence premises that require different types of knowledge to predict the

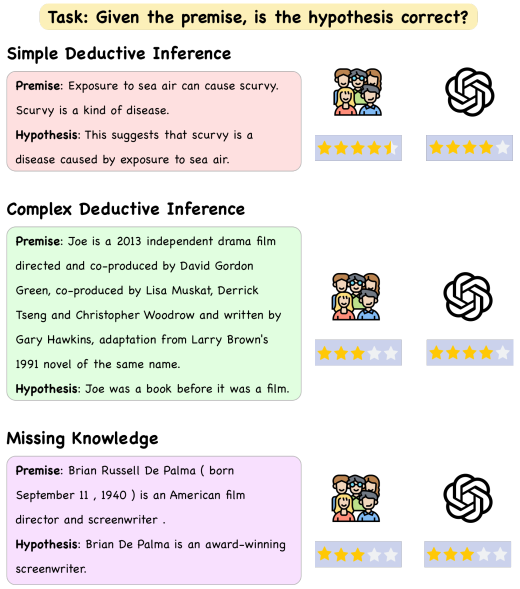
Desirable Properties	**NLI**			**Contextual QA**					**Rationale**	
	WaNLI	FEVER	ANLI	CosQA	SIQA	DREAM	BoolQ	RACE	Entailer	ECQA
Multi-sentence premise		✓	✓	✓		✓	✓	✓	✓	✓
Explanatory premise									✓	✓
Entity-grounded knowledge		✓	✓				✓		✓	
Commonsense knowledge				✓	✓					✓
Localized knowledge	✓			✓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

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.

entailment. Due to these challenges, we encourage researchers to use this benchmark for evaluating entailment verification.

Our analyses show that LLMs are stronger than humans at entailment tasks that involve multi-hop reasoning across long contexts that require slow thinking. This suggests that training LLMs to reason on diverse long context data is beneficial for entailment verification on such complex instances. In contrast, we find that humans are better at solving entailment tasks that require simple deductive reasoning using substitutions, negations, etc., indicating that current LLMs still lack consistency along these reasoning aspects. Further, we find that humans and LLMs perform comparably in entailment tasks requiring inferring some missing knowledge. These findings are further depicted in Figure 1 with motivating examples.

Among the different LLMs, we find that general instruction-finetuned models are better than task-finetuned models trained on a specific dataset category. We leverage this finding by finetuning a Flan-T5 Chung et al. (2022) model on a training subset containing datasets from each category. 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. We find that ranking-based finetuning is superior to classification as it can learn a softer decision boundary. Overall, our fine-tuned models outperform GPT-3.5 and perform comparably to GPT-4 on the benchmark, thus providing a strong open-sourced model for entailment verification.


Task: Given the premise, is the hypothesis correct?

Simple Deductive Inference

Premise: Exposure to sea air can cause scurvy. Scurvy is a kind of disease.
Hypothesis: This suggests that scurvy is a disease caused by exposure to sea air.

Complex Deductive Inference

Premise: Joe is a 2013 independent drama film directed and co-produced by David Gordon Green, co-produced by Lisa Muskat, Derrick Tseng and Christopher Woodrow and written by Gary Hawkins, adaptation from Larry Brown's 1991 novel of the same name.
Hypothesis: Joe was a book before it was a film.

Missing Knowledge

Premise: Brian Russell De Palma (born September 11 , 1940) is an American film director and screenwriter .
Hypothesis: Brian De Palma is an award-winning screenwriter.

Figure 1: **Distinctions between human and LLM Inferences.** The entailment prediction performance of humans and LLMs are depicted by a 5-star rating scale. Humans are more consistent on simple deductive reasoning whereas LLMs excel at instances requiring complex, multi-step inferences over long contexts. Both humans and LLMs are comparable on instances with missing knowledge. Please refer to Section 2 for more details.

Finally, we demonstrate the utility of our finetuned models on a downstream application of filtering unfaithful model-generated explanations. Recent works Wei et al. (2022); Zhou et al. (2023); Wang et al. (2023) have prompted LLMs to generate natural language explanations for model predictions. However, such reasoning steps can sometimes be *inconsistent*, i.e., the explanation does not entail the model's final prediction Ye & Durrett (2022). In self-consistency (SC) Wang et al. (2023), multiple reasoning paths are first sampled from the LLM decoder for a given instance and then aggregated to predict the most consistent answer. We use our finetuned models to filter out non-entailed reasoning chains in the SC decoding strategy before aggregating the final prediction, which leads to 6% performance improvement on average across three MCQ datasets.

2 HUMANS VS. LLMS ON ENTAILMENT VERIFICATION

In this section, we define the datasets used to create the evaluation benchmark, the evaluation procedure for evaluating different LLM baselines and humans, and compare them across different reasoning types. The entailment verification task can be formally defined as follows: for a given premise (or context/explanation) p and a hypothesis (or claim) h, the task of entailment verification (EV) 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 $f(p, h) = \{\textit{support}, \textit{not support}\}$, where f is a classifier (human/LLM).

2.1 EVALUATION BENCHMARK

To benchmark humans and LLMs on the task of entailment verification, a basic criterion for dataset selection is that the dataset should contain a premise, a hypothesis, and a support/not support label associated with it. Additionally, we list some more desirable characteristics we want to cover in this study, as discussed below:

- **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 these information as entity-grounded, commonsense, or localized. Entity-grounded knowledge consists of information about entities and other general knowledge that are 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 these 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.

Next, we describe the three data sources we consider for creating the entailment verification benchmark. 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 NLI is an obvious choice of data source. 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), FEVER Nie et al. (2019), and ANLI Nie et al. (2020).

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.

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 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. Overall, 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).

Model	NLI			Contextual QA					Rationale		Avg
	WaNLI	FEVER	ANLI	CosQA	SIQA	DREAM	BoolQ	RACE	Entailer	ECQA	
RoBERTa	0.79	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	0.90	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	0.88	0.67	0.63	0.74	0.87	0.77	0.61	0.91	0.48	0.73
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 other LLMs on 100 sampled instances for each dataset. We report the macro-F1 score and highlight the differences between human and GPT-4 performance that are ≥ 0.10 in bold. **Takeaways**: GPT-4 is the best performing LLM across all the baselines considered. 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 2.5 for more analysis.

Rationale Lastly, we consider data sources where human-annotated explanations are 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), and ECQA Aggarwal et al. (2021).

2.2 LLM Evaluation Setup

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

Task-finetuned LLMs In this category, the models considered are already finetuned for either NLI or the exact entailment verification task itself. We evaluate RoBERTa Liu et al. (2019) (finetuned on NLI data) and Entailer-11B Tafjord et al. (2022) (finetuned on Entailer data) in this category. Please refer to Appendix B.1 for more details on the evaluation setup for these models.

Instruction-finetuned LLMs These are "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. 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:

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

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 few-shot prompts.

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.

2.3 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 imbalances 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.

2.4 Human vs. LLMs

First, we randomly sample 100 instances for 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. Among LLMs, we observe that instruction-finetuned LLMs are stronger than task-finetuned models that are trained on some selected datasets. This shows that using a specific dataset for fine-tuning does not necessarily improve the general inference abilities of LLMs. On comparisons between humans and LLMs, we find that humans beat all the baseline LLMs, except GPT-4. This shows that existing open-sourced LLMs are not at par with humans on this task. Additionally, although humans and GPT-4 perform comparably on average, there are large misalignments in different individual datasets. Specifically, we observe that ANLI, CosQA, SIQA, and Entailer are the four datasets with >0.1 absolute macro-F1 difference. Next, we perform more analysis on these four datasets to better understand these misalignments.

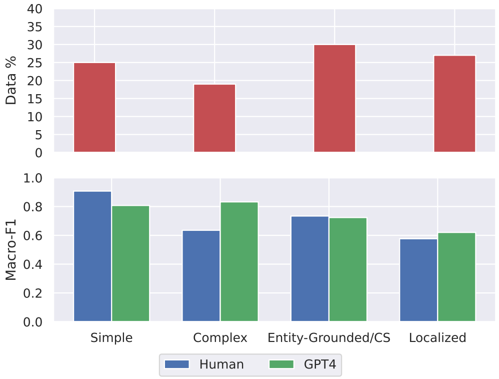


Figure 2: Analysis of the different reasoning types involved in entailment verification. **[Top]** Distribution of the reasoning types aggregated across four datasets: ANLI, CosQA, SIQA, and Entailer. **[Bottom]** Performance comparisons between humans and GPT-4. **Takeaway**: GPT-4 is better than humans at instances requiring complex reasoning, while humans are more consistent in simpler deductive reasoning tasks. Refer to Section 2.5 for more details.

2.5 Reasoning Type Analysis

We design a new analysis to further understand the reasons behind such misalignment between models and humans. First, we categorize the type of reasoning required to predict an entailment into the following four categories:

- **Simple Deductive (R1)**: In this type, the premise contains sentences that can be minimally combined in one step to predict the support for the hypothesis. This type typically tests skills such as substitution, negation, synonyms, etc.
- **Complex Deductive (R2)**: In this, more than one step of reasoning using information from the premise is required to solve the task. Typically, this type tests skills like mathematical reasoning, using multiple information in context to arrive at a conclusion, etc.
- **Missing Entity-grounded/Commonsense Knowledge (R3)**: This is a case where some essential commonsense or entity-grounded knowledge is missing in the premise. Such information can be implicitly invoked by humans and the model's parametric knowledge obtained from pretraining.
- **Missing Localized Knowledge (R4)**: In this, information very specific to the context of the premise is missing. Typically, this is information about the subjects in the context and is not grounded to any entities that can be known via the internet. It is practically impossible for humans or the model to infer such missing information.

Please refer to Appendix E.3 for examples of each reasoning type. We note that these categories are mutually exclusive[1]. Given this categorization, we first annotate the reasoning type of the 100 previously sampled instances for each dataset with absolute macro-F1 difference >0.1. We make

[1] 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 finetuning and the average is reported in "Seen Avg" column. Other datasets are zero-shot evaluated and average is reported in "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 3.2 for more details.

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. Please refer to Appendix E.2 for more details on the annotation setup. Figure 2 depicts the aggregated results for each reasoning type. The top plot shows the percentage of each reasoning type among 400 samples, and the bottom plot compares the human and GPT-4 macro-F1 scores. Please refer to Appendix E.4 for a detailed analysis of individual datasets.

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 (refer to Table 10 for examples). We find that humans are more robust than GPT-4 in performing such simple deductive 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 human baseline on complex reasoning that constitutes $\sim$ 20% data. This type of reasoning usually requires two skills: understanding multiple relevant information in the premise and combining them for reasoning. There are two potential reasons for this misalignment. First, compared to humans, GPT-4 is likely a stronger context processor, especially for long premises, since it has been trained on long-context data sources OpenAI (2023). Alternatively, another potential confounding factor is that AMT workers are typically incentivized to quickly finish annotations to maximize earnings. Thus, instances that require multi-step reasoning can have noisier annotations, potentially underestimating true human performance.

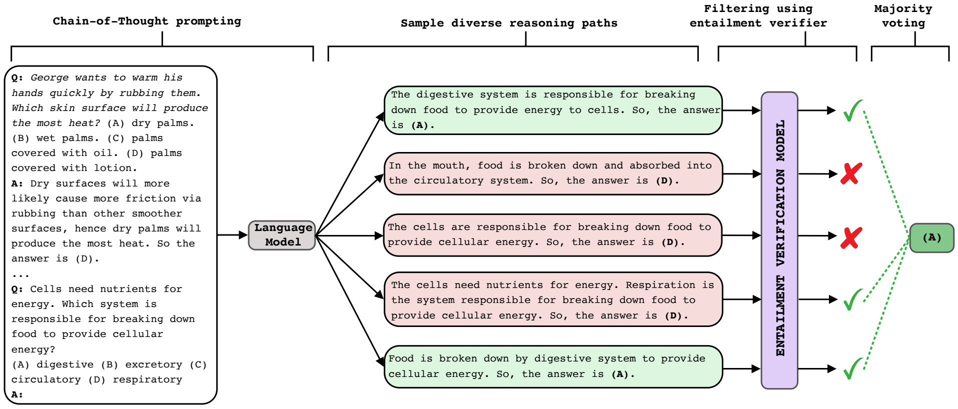
Lastly, we observe that approximately 30% of the data has some missing entity-grounded or commonsense information while $\sim$ 25% of the data has missing localized information. To correctly predict entailment in such instances, a system should be able to infer some of the missing grounded knowledge 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.

3 Training LLMs for Entailment Verification

In Section 2.4, we observed that open-sourced LLMs are lacking in performance compared to humans and close-sourced models such as GPT-4. Thus, in this section, we propose training strategies to finetune an open-sourced model that can perform competitively w.r.t. GPT-4. Among the different LLMs we compared in Table 3, we found that general instruction-finetuned models are better than task-finetuned models trained on a specific dataset category. Using this insight, we finetune a Flan-T5-xxl Chung et al. (2022) model using 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). Please refer to Appendix F.1 for more details on our training dataset selection criteria. We describe two approaches to finetuning and then discuss our findings.

3.1 Finetuning Formulations

This section describes the two fine-tuning formulations explored in this work.


Chain-of-Thought prompting
Sample diverse reasoning paths
Filtering using entailment verifier
Majority voting
Language Model
ENTAILMENT VERIFICATION MODEL
✓
✗
✗
✓
✓
(A)

Q: *George wants to warm his hands quickly by rubbing them. Which skin surface will produce the most heat?* (A) dry palms. (B) wet palms. (C) palms covered with oil. (D) palms covered with lotion.
A: Dry surfaces will more likely cause more friction via rubbing than other smoother surfaces, hence dry palms will produce the most heat. So the answer is (D).
...
Q: 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
A:

The digestive system is responsible for breaking down food to provide energy to cells. So, the answer is **(A)**.

In the mouth, food is broken down and absorbed into the circulatory system. So, the answer is **(D)**.

The cells are responsible for breaking down food to provide cellular energy. So, the answer is **(D)**.

The cells need nutrients for energy. Respiration is the system responsible for breaking down food to provide cellular energy. So, the answer is **(D)**.

Food is broken down by digestive system to provide cellular energy. So, the answer is **(A)**.

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). Please refer to Section 4 for more details.

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 triple (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:

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

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 w.r.t. to the best option in QA instead of being absolutely incorrect. Training to hard-classify the hypothesis for such options can be avoided by ranking them with the 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.

3.2 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 out 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.

4 APPLICATION: FILTERING COT RATIONALES

Recently, Wang et al. (2023) proposed self-consistency (SC), a decoding technique to improve 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. But generative LLMs can potentially output rationales that are inconsistent Ye & Durrett (2022), i.e., the rationale does not support the corresponding model prediction. Such inconsistency can, in turn, degrade the overall self-consistency results. This motivates an important application of entailment verification systems in this paradigm. Please refer to Appendix G.1 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 $k = 10$ 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 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 $\sim$ 1.5 %. 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.1 for examples of filtered CoT rationales.

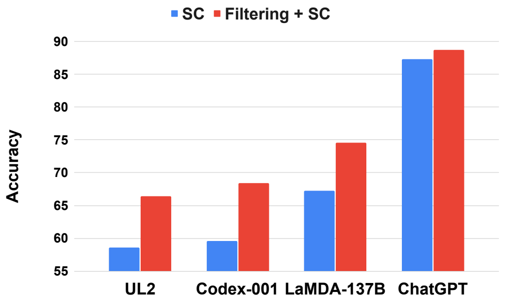


Figure 4: Comparisons between self-consistency (SC) and Filtering + SC. We report the accuracy metric averaged across three MCQ datasets for four different LLMs. **Takeaway**: Filtering consistently improves performance over SC baseline, with more gains for weaker base models such as UL2. Please refer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

5 RELATED WORKS

Natural Language Inference NLI Dagan et al. (2006); Manning & 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 strong, 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 & Chang (2023). Recently, Ye & 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 a strong open-sourced LLM that can be easily deployed to check entailment errors in LLM reasoning chains.

6 Conclusion

We studied the entailment verification problem in the context of LLMs. Specifically, we sourced datasets across three different categories (NLI, contextual QA, and rationales) and analyzed both human and model performance across these datasets. We found some misalignments between the two, whereby models are better than humans in complex reasoning and humans are relatively more consistent on simpler reasoning tasks. We also explored different finetuning objectives to train LLMs for verification. Our fine-tuned models outperform GPT-3.5 and are at par with GPT-4. Finally, we demonstrated a practical application of entailment verification in filtering out inconsistent model-generated rationales while using self-consistency decoding. Overall, we believe our work provides a meaningful comparison of inference capabilities between human and LLMs, a comprehensive evaluation benchmark, an open-sourced entailment verification model, and interesting applications of the task.

Limitations

Even though our work demonstrates exciting results on entailment verification tasks by finetuning LLMs, several limitations can be potentially improved. We only tried encoder-decoder-based models for finetuning. However, other models with different architectures (like decoder-only) can also be considered. 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. 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.

Acknowledgments

This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract #2022-22072200006, the DARPA MCS program under Contract No. N660011924033, the Defense Advanced Research Projects Agency with award W911NF-19-20271, NSF IIS 2048211, and gift awards from Google and Amazon. The views and conclusions contained herein are those of the authors. They should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. We would like to thank all the USC INK research lab collaborators for their constructive feedback on the work.

REFERENCES

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. Explanations for CommonsenseQA: New Dataset and Models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *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)*, pp. 3050–3065, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.238. URL https://aclanthology.org/2021.acl-long.238.

Giannis Bekoulis, Christina Papagiannopoulou, and Nikos Deligiannis. A review on fact extraction and verification. *ACM Comput. Surv.*, 55(1), nov 2021. ISSN 0360-0300. doi: 10.1145/3485127. URL https://doi.org/10.1145/3485127.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, and Jian Su (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075.

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. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Timothy J. Buschman, Markus Siegel, Jefferson E. Roy, and Earl K. Miller. Neural substrates of cognitive capacity limitations. *Proceedings of the National Academy of Sciences*, 108(27): 11252–11255, 2011. doi: 10.1073/pnas.1104666108. URL https://www.pnas.org/doi/abs/10.1073/pnas.1104666108.

Jifan Chen, Eunsol Choi, and Greg Durrett. Can NLI models verify QA systems' predictions? In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3841–3854, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.324. URL https://aclanthology.org/2021.findings-emnlp.324.

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. Scaling instruction-finetuned language models, 2022.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300.

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

Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. In Christian Bessiere (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 3882–3890. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

Nelson Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24(1):87–114, 2001. doi: 10.1017/S0140525X01003922.

Kevin Crowston. Amazon mechanical turk: A research tool for organizations and information systems scholars. In Anol Bhattacherjee and Brian Fitzgerald (eds.), *Shaping the Future of ICT Research. Methods and Approaches*, pp. 210–221, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-35142-6.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In Joaquin Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc (eds.), *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pp. 177–190, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. Explaining answers with entailment trees. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7358–7370, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.585. URL https://aclanthology.org/2021.emnlp-main.585.

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

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

Alan Garnham. Inference in language understanding: What, when, why and how. In RAINER DIETRICH and CARL F. GRAUMANN (eds.), *Language Processing in Social Context*, volume 54 of *North-Holland Linguistic Series: Linguistic Variations*, pp. 153–172. Elsevier, 1989. doi: https://doi.org/10.1016/B978-0-444-87144-2.50009-4. URL https://www.sciencedirect.com/science/article/pii/B9780444871442500094.

Atticus Geiger, Kyle Richardson, and Christopher Potts. Neural natural language inference models partially embed theories of lexical entailment and negation. In Afra Alishahi, Yonatan Belinkov, Grzegorz Chrupała, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad (eds.), *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 163–173, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.16. URL https://aclanthology.org/2020.blackboxnlp-1.16.

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

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

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 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)*, pp. 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017. URL https://aclanthology.org/N18-2017.

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1049–1065, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.67. URL https://aclanthology.org/2023.findings-acl.67.

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

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 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)*, pp. 2391–2401, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1243. URL https://www.aclweb.org/anthology/D19-1243.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. Maieutic prompting: Logically consistent reasoning with recursive explanations. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 1266–1279, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.82. URL https://aclanthology.org/2022.emnlp-main.82.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082.

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

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. WANLI: Worker and AI collaboration for natural language inference dataset creation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 6826–6847, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.508. URL https://aclanthology.org/2022.findings-emnlp.508.

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

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

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

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

Tom McCoy, Ellie Pavlick, and Tal Linzen. 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*, pp. 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL https://aclanthology.org/P19-1334.

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

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: A comprehensive review. *ACM Comput. Surv.*, 54(3), apr 2021. ISSN 0360-0300. doi: 10.1145/3439726. URL https://doi.org/10.1145/3439726.

Anshuman Mishra, Dhruvesh Patel, Aparna Vijayakumar, Xiang Li, Pavan Kapanipathi, and Kartik Talamadupula. Reading comprehension as natural language inference:a semantic analysis. In Iryna Gurevych, Marianna Apidianaki, and Manaal Faruqui (eds.), *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pp. 12–19, Barcelona, Spain (Online), December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.starsem-1.2.

Eric Mitchell, Joseph Noh, Siyan Li, Will Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher Manning. Enhancing self-consistency and performance of pre-trained language models through natural language inference. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 1754–1768, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.115. URL https://aclanthology.org/2022.emnlp-main.115.

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

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

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 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, 2020.

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

OpenAI. Gpt-4 technical report, 2023.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Soumya Sanyal, Zeyi Liao, and Xiang Ren. RobustLR: A diagnostic benchmark for evaluating logical robustness of deductive reasoners. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9614–9631, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.653. URL https://aclanthology.org/2022.emnlp-main.653.

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

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231, 2019. doi: 10.1162/tacl_a_00264. URL https://aclanthology.org/Q19-1014.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. Entailer: Answering questions with faithful and truthful chains of reasoning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2078–2093, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.134. URL https://aclanthology.org/2022.emnlp-main.134.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 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)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421.

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. Ul2: Unifying language learning paradigms, 2023.

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. Lamda: Language models for dialog applications, 2022.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 809–819, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1074. URL https://aclanthology.org/N18-1074.

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

Adina Williams, Nikita Nangia, and Samuel Bowman. 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)*, pp. 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.

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

Xi Ye and Greg Durrett. The unreliability of explanations in few-shot prompting for textual reasoning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=Bct2f8fRd8S.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5823–5840, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.320. URL https://aclanthology.org/2023.acl-long.320.

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. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WZH7099tgfM.

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, else we use the dev split. Please refer to Appendix A for more details about each dataset.

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). And the test split is used when doing 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 not available, we report results on 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. 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 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 dev split are reported given 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 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 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 have to be combined to either support or refute the hypothesis. The test split is also missing for this dataset and we can only evaluate 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 corresponding model card on HuggingFace[2] 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 FEVER and ANLI in Table 3.

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 "*Yes*" label and combine the remaining two classes to be "*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 on the same dataset.

We strictly follow the official implementation of the model to acquire class labels.[3] 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.

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

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

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**: Model is robust to variations in the prompt and generating consistent results. Please refer to Appendix B.2 for more details.

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. Following are models included in our evaluation.

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 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 finetuning 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.

GPT-3.5 Brown et al. (2020) It is a general-purpose autoregressive decoder-only LMs accessible via the OpenAI Completions API.[4] We utilize "text-davinci-003" in OpenAI's API for evaluation, and the label determination procedure is quite similar to the one in Flan-T5-xxl. We monitor the probability of "*Yes*" and "*No*" tokens. After applying the softmax function to them, we select the token that attains a probability larger than a threshold to be our label.

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.[5] We adopt plain "gpt-4" in OpenAI's API for our experiments. Unlike other models, the output probabilities are not accessible. Alternatively, we firstly constrain the model to predict "*Yes*" or "*No*" only by putting some instructions into the prompt. Then, we check the output text. In the ideal case, we can directly use the output text as the label since only "*Yes*" or "*No*" is produced. Otherwise, we need to randomly choose one as the label.

B.2.1 Prompts for Robustness Assessment

To assess the robustness of the model in section 2.2, 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 model is insensitive 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 generally constructed 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*,

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

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

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 boosts the performance while it is not beneficial for other two models. Please refer to Appendix B.2 for more analysis.

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 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.

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 datasets we evaluate. More details are presented in Appendix C.

D Results on Hypothesis Groundedness

We define the *groundedness* of a hypothesis based on the type of knowledge required to establish its validity. Using this categorization, we classify the datasets into two categories as follows:

- **Knowledge-based**: A knowledge-based hypothesis typically contains entity-grounded or commonsense knowledge. Such a hypothesis is usually well-defined (i.e., the premise is not required to ground it), and models can potentially leverage their internal knowledge to verify the entailment without using the premise. The datasets that contain such knowledge-based hypotheses are FEVER, ANLI, BoolQ, Entailer, and ECQA.

- **Contextual**: A contextual hypothesis requires localized knowledge, i.e., the premise/context grounds the hypothesis. These hypotheses cannot be verified on their own without using the premise. Datasets with contextual hypotheses are WaNLI, CosQA, SIQA, DREAM, and RACE.

Performance w.r.t. groundedness Figure 5 shows the aggregate performance across the two hypothesis categories described above. First, we observe that performance on the knowledge-based hypotheses is higher on average than contextual hypotheses for all models (except GPT-3.5 and GPT-4, for which they are comparable). This is likely because models can potentially leverage their internal knowledge stored in the parameters to check the validity of the knowledge-based hypotheses. Such a scenario is harder for contextual cases since the hypothesis solely depends on the context and would require a strong context understanding to predict the hypothesis entailment.

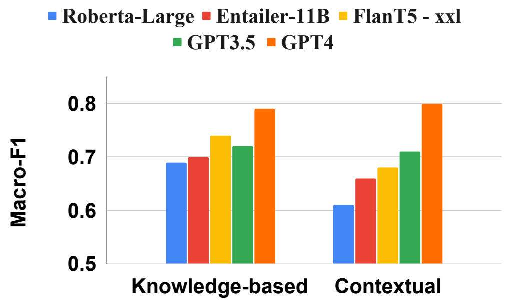


Figure 5: Performance comparisons of LLMs across two hypothesis categories. We report the average macro-F1 score for each model. **Takeaways**: Contextual hypotheses are harder than knowledge-based hypotheses. Scaling model/training benefits the performance of contextual cases more than knowledge-based ones. Please refer to Appendix D for details.

The models in the plot are arranged with increasing complexity, i.e., along the x-axis, either the model size or the amount/variety of pretraining data increases. We observe that the performance on contextual hypotheses almost consistently improves with scaling, whereas the trend is not so strong for the knowledge-based case. This shows that scaling is useful for training LLMs in understanding contexts. However, this does not necessarily improve LLMs' knowledge retention or retrieval skills.

E Human Alignment Experiments

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

In every HIT, we used Figure 8 as the reasoning type annotation format. It was an intense job with two tasks involved. The instructions and label explanations were explicitly stated at the beginning of the format. For task 1, after reading the premise and the claim, annotators should 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

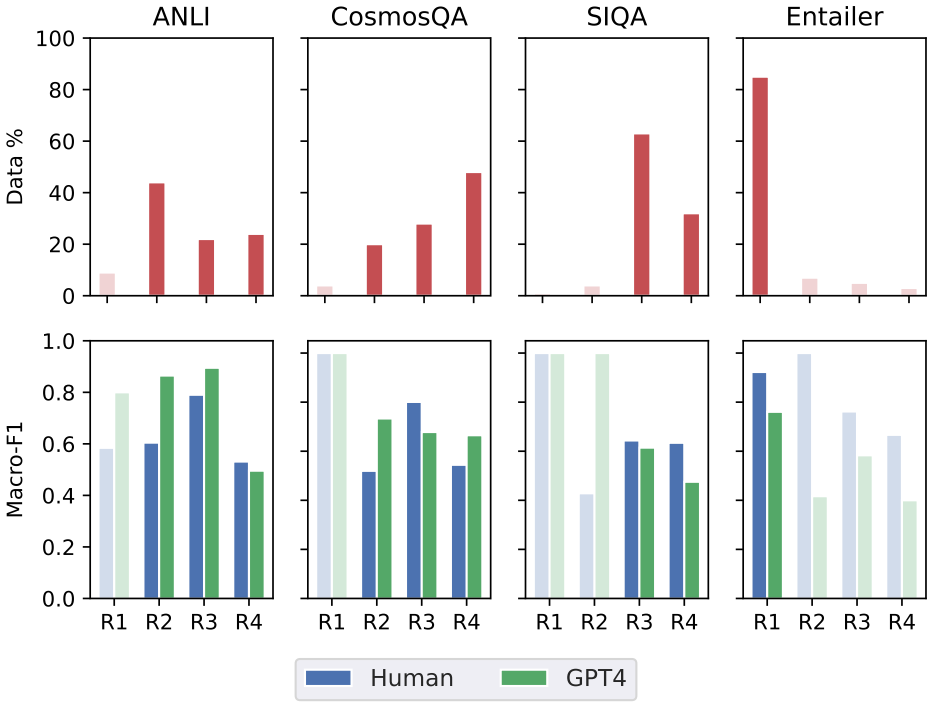


Figure 6: Analysis of different reasoning types involved in entailment verification for each dataset. **[Top]** Distribution of the reasoning types for each dataset studied. **[Bottom]** Macro-F1 performance comparison between humans and GPT-4. We fade out all bars with distribution percentage $\leq 10\%$ 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.

of the claim in task 2. Otherwise, the type of missing information should be decided in task 1, and task 2 did not apply to these cases. As stated in 2.5, we aggregated answers from 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 NLP background knowledge was required to understand task descriptions and label explanations, we recruited Computer Science graduate students instead of the general public to finish this MTurk annotation. Similarly, 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 10 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** 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 prominent 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 note that the amount of training data is quite low for the Rationale category. 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.

> 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.

Method	CSQA	ARC-e	ARC-c	Average
UL2 + SC	55.77	70.33	49.57	58.56
UL2 + Filter (Flan-T5-xxl) + SC	61.75	75.15	54.94	63.95
UL2 + Filter (Flan-T5-xxl + *Rank*) + SC	63.06	77.39	58.81	66.42
Codex-001 + SC	54.80	71.70	52.20	59.57
Codex-001 + Filter (Flan-T5-xxl) + SC	62.90	74.20	56.91	64.67
Codex-001 + Filter (Flan-T5-xxl + *Rank*) + SC	67.16	77.86	60.20	68.41
LaMDA-137B + SC	62.90	78.90	59.90	67.23
LaMDA-137B + Filter (Flan-T5-xxl) + SC	71.33	82.20	64.16	72.56
LaMDA-137B + Filter (Flan-T5-xxl + *Rank*) + SC	71.74	83.84	68.09	74.56
ChatGPT + SC	78.40	96.30	87.20	87.30
ChatGPT + Filter (Flan-T5-xxl) + SC	81.20	96.40	87.90	88.50
ChatGPT + Filter (Flan-T5-xxl + *Rank*) + SC	81.00	96.50	88.70	88.73

Table 9: 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.

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 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.

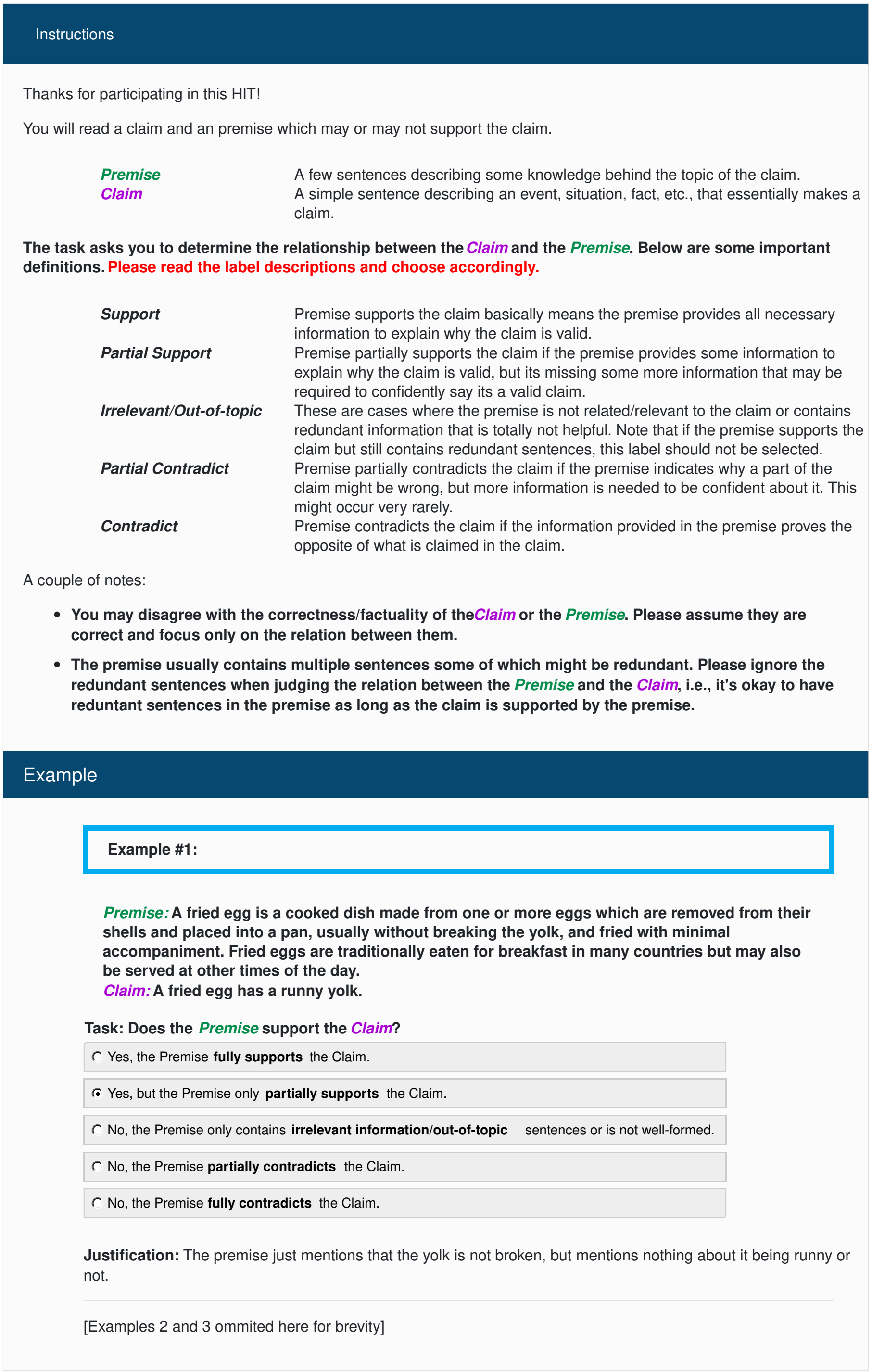
- $\mathcal{B}+$ Flan-T5-xxl + SC: In this, we use a pre-trained Flan-T5-xxl for filtering out the inconsistent rationales before the majority voting. We keep the top-10 rationales after scoring them using Flan-T5-xxl.
- $\mathcal{B}+$ Flan-T5-xxl + *Rank* + SC: This is the same as above, but instead, 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[6] for details on the CoT prompt formats. The results are shown in Table 9. 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 finetuning approach. Please refer to Section 4 for more findings.

G.1 Filtering Examples

Table 11 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 will have a low entailment score leading to a lower ranking and getting filtered out.

[6] `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.

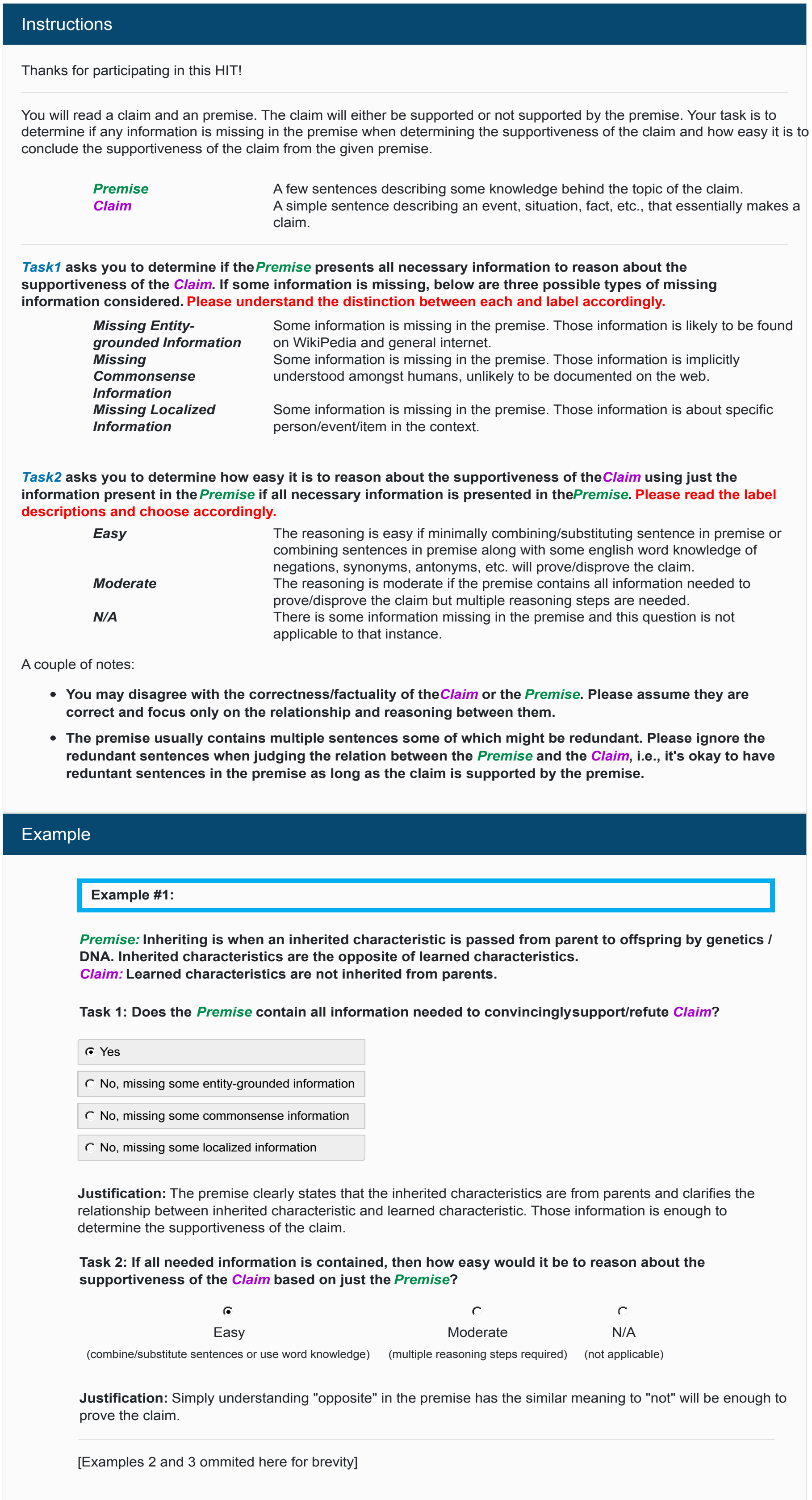
○ No, the Premise **partially contradicts** the Claim.

○ 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 convincinglysupport/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 10: Comparison between the performance of human and GPT-4 on examples from four reasoning types. Each example includes the golden label from the dataset and shows if human/GPT-4 make the correct decision. The definition of those reasoning types and detailed analysis can be found in Section 2.5.

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 11: Filtering examples from three distinct datasets. Each example contains two outputs that are kept and three outputs that are filtered out in ranking procedure.