Say What You Mean! Large Language Models Speak Too Positively about Negative Commonsense Knowledge

Jiangjie Chen[♠], **Wei Shi**[♠], **Ziquan Fu**[♡]*, **Sijie Cheng**[♠], **Lei Li**[♣], **Yanghua Xiao**[♠◇†]
[♠]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[♡]System Inc. [♣]University of California, Santa Barbara
[◇]Fudan-Aishu Cognitive Intelligence Joint Research Center
{jjchen19, sjcheng20, shawyh}@fudan.edu.cn
wshi22@m.fudan.edu.cn, frank@system.com, leili@cs.ucsb.edu

Abstract

Large language models (LLMs) have been widely studied for their ability to store and utilize positive knowledge. However, negative knowledge, such as "*lions don't live in the ocean*", is also ubiquitous in the world but rarely mentioned explicitly in the text. *What do LLMs know about negative knowledge?* This work examines the ability of LLMs to negative commonsense knowledge. We design a constrained keywords-to-sentence generation task (CG) and a Boolean question-answering task (QA) to probe LLMs. Our experiments reveal that LLMs frequently fail to generate valid sentences grounded in negative commonsense knowledge, yet they can correctly answer polar yes-or-no questions. We term this phenomenon the *belief conflict* of LLMs. Our further analysis shows that statistical shortcuts and negation reporting bias from language modeling pre-training cause this conflict.[1]

1 Introduction

Most of the world knowledge exists in a positive and affirmative form (Molnar, 2000; Barker and Jago, 2012; Vrandečić and Krötzsch, 2014; Speer et al., 2017). As a result, large language models (LLMs) pre-trained on a colossal amount of texts, such as GPT-3 (Brown et al., 2020; Ouyang et al., 2022) and PaLM (Chowdhery et al., 2022), have demonstrated their remarkable abilities for storing and utilizing positive knowledge in downstream tasks. In contrast, negative knowledge, such as the commonsense statement that "*lions do not live in the ocean*", is rarely mentioned in the textual world (Hossain et al., 2022).[2] Such negative knowledge also exists in the real world, and is important for cognitive skills such as knowing *what is not true* or *what not to think* (MacDonald, 1965; Minsky, 1997; Barker and Jago, 2012). Therefore, we ask this question: *Do LLMs (such as GPT-3 models) acquire such implicit negative knowledge through extensive language modeling pre-training?*

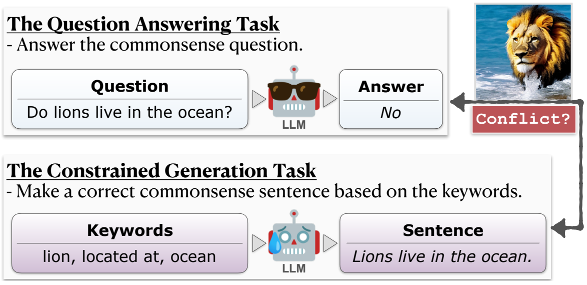

The Question Answering Task
- Answer the commonsense question.
Question
Do lions live in the ocean?
LLM
Answer
No
Conflict?
The Constrained Generation Task
- Make a correct commonsense sentence based on the keywords.
Keywords
lion, located at, ocean
LLM
Sentence
Lions live in the ocean.

Figure 1: An example of the probing tasks studied in this paper. For the same negative commonsense knowledge <*lion, located at, ocean*> which is false, we find LLMs often fail to generate texts grounded in such negative knowledge while knowing its validity according to question answering.

One important way of probing LLMs, which are mostly generative models, is checking whether the generated texts are knowledge-grounded. This is because the generation of texts is a direct manifestation of a model's internal beliefs towards world knowledge (Kassner et al., 2021; Sumers et al., 2021; Tafjord et al., 2022).[3] Knowledge-grounded text generation has been a focus of NLP research (Yu et al., 2022). For example, the CommonGen benchmark (Lin et al., 2020) evaluates generative commonsense reasoning that organizes concepts as keyword input and generates a sentence grounded in commonsense knowledge. However, previous work does not consider negative knowledge, nor do they probe the consistency between

*Work done while at Brain Technologies, Inc.

†Corresponding author.

[1] Resources of this paper are available at https://github.com/jiangjiechen/uncommongen.

[2] Hossain et al. (2022) report that sentences with negation hold up to 14.5% in the CommonsenseQA dataset (Talmor et al., 2019), 8.7% in QNLI (Rajpurkar et al., 2016), and 22.6-29.9% in general-purposed texts.

[3] Our definition of belief is derived from Kassner et al. (2021), which is the assignment of a truth value to a proposition. In our study, the context for the proposition is the world knowledge that models learned. Therefore, we define a model's belief about such knowledge as its prediction about the truth value of a certain piece of world knowledge.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 9890–9908
July 9-14, 2023 ©2023 Association for Computational Linguistics

what models know and what they generate. Another line of work on probing (Petroni et al., 2019; Ettinger, 2020; Kassner and Schütze, 2020; Cao et al., 2021) is conducted through the mask-infilling task. However, this task mainly evaluates bidirectional models (Devlin et al., 2019), and is not natural for unidirectional LLMs. Also, this task suffers from the *open-world problem* in evaluation, *i.e.*, there could be multiple valid answers to fill the mask. This is vital for evaluating negative knowledge, which has an infinite answer space, *e.g.*, lions don't live in the *sky, water, desk, car*, etc.

In this study, we investigate the belief of LLMs about negative commonsense knowledge through the lens of *text generation*. Since LLMs have become a foundational service (Bommasani et al., 2021) and cannot be easily trained, we apply in-context learning (Brown et al., 2020) for the probing tasks, which is tuning-free. We design a Constrained Sentence Generation (CG) probing task, following Lin et al. (2020), where the model must generate a knowledge-grounded sentence based on a given triple <s, r, o>. For example, given a triple "<*lion*, *located at*, *ocean*>", a model should generate "*lions do not live in the ocean*". This task is rather simple and clear. The output sentence basically contains the same information as the input keywords. Thus, the generated texts are easy to evaluate according to the appearance of negation. We also add a Boolean Question Answering (QA) task that asks LLMs whether a knowledge triple is valid, which shows their beliefs about this piece of knowledge. An example is given in Figure 1.

In our experiments, we find that LLMs of different sizes and shapes often produce hallucinated claims of negative knowledge, even if they answer yes-or-no questions about it correctly. We term this phenomenon the *belief conflict*, *i.e.*, actions (generating texts with it) conflict with its belief (answering question about it). Hallucinated generation of negative knowledge is seen in both our probing tasks and downstream tasks, such as explanation generation (Chen et al., 2022; Jung et al., 2022), where negative knowledge plays an important role in the argumentation of refutation. Further analysis shows that this problem stems from the statistical shortcuts and reporting bias of negation during pre-training. Moreover, such implicit biases can be alleviated through explicit reasoning with Chain-of-Thought prompting (Wei et al., 2022b), such as syllogistic deduction and related fact comparison.

The main contributions of this paper are summarized as follows: *1)* We are the first to investigate LLMs' belief about negative knowledge in the commonsense domain, which may shed light on a previously unstudied aspect of LLMs' abilities. *2)* We propose to probe generative LLMs through constrained sentence generation, which is effective for evaluating generated texts grounded in positive and negative knowledge. *3)* Through extensive experiments, we identify and analyze LLMs' *belief conflict* phenomenon on negative commonsense knowledge, and provide insights on the causes and solutions of such problems.

2 Related Work

Negative Knowledge Negative knowledge refers to information that describes what is not true, what cannot be done, or what does not exist, while everything that exists is positive (Molnar, 2000; Barker and Jago, 2012). It plays an important role in the human reasoning process, because to think effectively, we need to know what "not to think" (Minsky, 1997). Current research of negative knowledge in NLP mainly focuses on developing negative knowledge bases that store relational negative commonsense knowledge (Arnaout et al., 2021; Safavi et al., 2021; Arnaout et al., 2022) and utilizing negative knowledge within arguments or explanations to refute a candidate (Camburu et al., 2018; Aggarwal et al., 2021; Chen et al., 2022). This paper is based on these resources to probe the belief of LLMs about the relations of everyday concepts that are not true.

Understanding Negation in Texts The manifestation of negative knowledge in texts is the phenomenon of negation (Horn and Wansing, 2022), which is difficult for pre-trained LMs to understand, *e.g.*, filling "*birds cannot* [MASK]" with "*fly*" (Kassner and Schütze, 2020). Negation has been shown to be spuriously correlated with negative or contradictory labels due to the data distribution (Gururangan et al., 2018; Ettinger, 2020; Lai et al., 2021; Branco et al., 2021; Tian et al., 2022), raising doubts about the performance of previous models. Furthermore, LMs may ignore the existence of negative words when understanding texts (Kassner and Schütze, 2020) or processing prompts (Jang et al., 2022), which can be alleviated with unlikelihood training objective (Welleck et al., 2020) during training (Hosseini et al., 2021) or specifying pragmatic contexts (Gubelmann and

Handschuh, 2022). While most current research focuses on NLU, this work fills in a gap in the investigation of the negation phenomenon in the context of text generation.

Knowledge-Grounded Language Models A major goal of NLP has been to ground LMs in world knowledge, such as factual knowledge (Vrandečić and Krötzsch, 2014) and commonsense knowledge (Speer et al., 2017). A line of work (Petroni et al., 2019; Kassner and Schütze, 2020; Cao et al., 2021) directly probes the knowledge implicitly learned by LMs through mask-infilling. However, such a probing paradigm only works for contextual LMs such as BERT (Devlin et al., 2019), leaving generative ones, especially modern LLMs, understudied. Another line of work focuses on making LM-generated sentences grounded in knowledge (Petroni et al., 2020; Liu et al., 2021). Lin et al. (2020) designed a constrained text generation task, CommonGen, which asks a model to generate a sentence given a set of concepts, testing the generative commonsense reasoning of LMs. However, these studies do not investigate text generation grounded in negative knowledge, which is the focus of this work.

In-Context Learning In-context learning (ICL; Brown et al., 2020) has become a prevailing paradigm for deploying LLMs (*e.g.*, the GPT-3 family Brown et al., 2020; Chen et al., 2021; Ouyang et al., 2022) for downstream tasks. Through ICL, LLMs can solve tasks directly based on input-output examples without parameter updates (Min et al., 2022a; Rubin et al., 2022). Furthermore, recent work (Wei et al., 2022b; Wang et al., 2022) reveals that the ceiling performance determined by the scaling law can be beaten with ICL by generating immediate rationales, *i.e.*, the Chain of Thought (CoT) prompting. Since LLMs are becoming a foundational service that do not need fine-tuning, our probing on LLMs are based on ICL.

3 Probing Protocol

In this section, we set up an evaluation protocol to understand what LLMs know about (negative) commonsense knowledge of everyday concepts.

3.1 The `CSK-PN` Dataset

We limit the scope of the knowledge probed to relational knowledge between commonsense concepts, *i.e.*, *relational knowledge triples*, which exist widely in knowledge graphs and are commonly studied by the community (Auer et al., 2007; Vrandečić and Krötzsch, 2014; Speer et al., 2017). Given a triplet in the form of <s, r, o> with a subject concept s, a relation r and an object concept o, we define a negative fact as $\neg r(s, o)$ if the truth value of $r(s, o)$ is `False` according to commonsense knowledge, and a (positive) fact if otherwise.

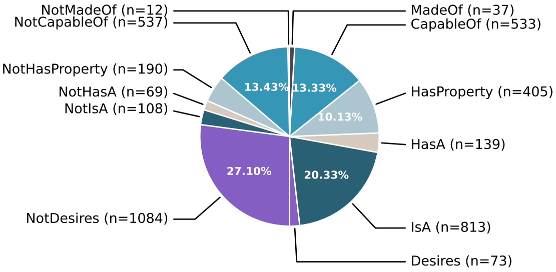


Figure 2: The configuration of the `CSK-PN` dataset.

Dataset Statistics We build the probing dataset (denoted as `CSK-PN`) based on the knowledge triples filtered by Safavi et al. (2021), which are the challenging ones sourced from ConceptNet (Speer et al., 2017). We also remove invalid triples with pronouns, negation, and adjectives as subjects or objects. The final dataset contains a total of 4,000 triples with six pairs of positive or negative relations (*e.g.*, IsA and NotIsA), and the positive and negative splits have the same size (1:1). Detailed information of `CSK-PN` is shown in Figure 2.

3.2 Probing Task Formulation

The most commonly used probing task for understanding whether LMs have certain types of knowledge is mask-infilling (Devlin et al., 2019; Petroni et al., 2020; Kassner and Schütze, 2020). However, this task is not suitable for generative LMs, as the mask must exist at the end of a sentence.

We argue that LLMs, which are mainly autoregressive text generation models (Radford et al., 2019; Brown et al., 2020; Ouyang et al., 2022; Scao et al., 2022), should be investigated by *text generation* with text decoding from a large sentence space. Therefore, we propose to use *Constrained Sentence Generation* (CG) as the primary task to investigate LLMs, coupled with *Boolean Question Answering* (QA) for comparison, which is a common approach to probing the belief of models (Tafjord et al., 2022; Richardson et al., 2022).

Task 1: Boolean Question Answering (QA) The Boolean QA task requires LLMs to express its belief about a fact by answering a yes-or-no question. We first transform every triplet <s, r, o> into a yes or no question q, where we remove the negation in r for negative facts. For example, a prompt goes like this:

> *Answer commonsense questions with yes or no:*
> (*Examples for in-context learning*)
> **Question**: do lions live in the ocean?
> **Answer**: no

where underlined texts are completed by LLMs. To generate the questions, we adopt InstructGPT using in-context learning (§4.1). The questions are 94% valid according to a manual inspection of 50 random cases.[4]

Task 2: Constrained Sentence Generation (CG) Generating texts is a direct manifestation of a model's belief. However, evaluating generated texts is notoriously difficult in NLP, especially without references. Therefore, we design a *keyword-to-sentence* task to make the probing more controllable, which is similar to COMMONGEN (Lin et al., 2020). Given a triple <s, r, o>, models need to generate sentences grounded in (negative) knowledge, *i.e.*, add negation cues (*e.g.*, *not*, *unable*) in the sentence if necessary, *e.g.*,

> *Write a short and factual sentence according to commonsense based on the keywords:*
> (*Examples for in-context learning*)
> **Keywords**: lion, located at, ocean
> **Sentence**: lions don't live in the ocean.

We remove the NOT prefix from the negated relations. Note that we allow the paraphrasing of the input keywords, making it a *soft*-constrained sentence generation task.

3.3 Evaluation Metrics

Metric for QA The QA task can be easily evaluated by checking the generated token *yes* and *no* (cased and uncased). We define `TP` and `TN` as the accuracy on the positive and negative splits in `CSK-PN`, and `Acc` as the accuracy on the whole dataset (*i.e.*, $\texttt{Acc} = (\texttt{TP} + \texttt{TN})/2$, since the positive and negative splits have equal size). For rare scenarios ($< 1\%$) that LLMs do not generate a `yes` or `no` token, we compare the conditional probability of these two tokens.

Metric for CG Due to the controlled task setting, which essentially forces LLMs to decide whether and how to add a negation cue during decoding, the CG task can be efficiently evaluated by detecting the existence of *negation cues* (*e.g.*, not, unable, etc.) in the generations. Following the QA task, we also use `TP` and `TN` as accuracy metrics. To implement this metric, we first use keywords-based matching for negation cues, followed by a RoBERTa model (Liu et al., 2019) as a *token classifier* looking for unmatched negation cues.[5] This metric produces 1 or 0 based on the finding of negation cues in a sentence. After manual inspection of 200 cases, we find that this metric is correct 97% of the time, which is reliable for evaluating such a constrained probing task. Errors are mostly due to double negations and ambiguous negative cues (*e.g.*, *less*, *opposite*, etc.), which are quite rare.

Can we trust negation detection as the metric to evaluate CG? We manually evaluate the factuality of generated texts based on commonsense knowledge and see whether the CG metric (detection of negation) correlates well with humans in this task. Note that only the sentences that make common sense and adhere to the keywords constraints are accepted as true during manual annotation. After examining 100 cases, we find that the agreement between human judgment and this metric achieves 95%. This is predictable, since this task is rather easy and constrained, yet LLMs do not solve it well, especially not very consistent with the QA task. Errors made by the metric are mostly because *1)* generated sentences use uncertain adverbs to modify the sentences, *e.g.*, *may*, *some*, etc.; *2)* noisy triples in the dataset. Overall, we think this metric is trustworthy and evaluates this task far better than most popular text generation metrics.

4 *Do LLMs have negative commonsense knowledge?*

In this section, we use `CSK-PN` to investigate LLMs' belief about negative commonsense knowledge. More importantly, *can LLMs generate texts grounded in negative commonsense knowledge?*

4.1 Probing LLMs with In-Context Learning

To execute the probing tasks without fine-tuning, we exploit the few-shot in-context learning (Brown

[4] Bad cases are mostly due to the quality of the triples, *e.g.*, <*swim, has property, full of water*>: *is swimming full of water?*

[5] The model is trained on the CONDAQA dataset (Ravichander et al., 2022), which has 14,182 QA pairs with more than 200 unique negation cues.

Model	k	Perf. on QA			Perf. on CG			Cns.
		TP	TN	Acc	TP	TN	Acc	
Flan-T5 (3B)	2	79.1	84.0	81.5	96.5	19.4	57.9	56.2
	10	82.7	80.2	81.4	96.9	19.8	58.4	59.7
Flan-T5 (11B)	2	84.1	81.0	82.6	97.5	15.9	56.7	57.7
	10	85.4	80.8	83.1	**97.6**	28.2	62.9	65.9
GPT-3	2	76.0	58.9	67.5	83.9	28.4	56.1	54.4
	10	74.7	66.9	70.8	30.9	**79.8**	55.3	53.7
Codex$_{002}$	2	**89.2**	81.7	**85.4**	96.6	38.0	67.3	70.1
	10	88.1	81.8	84.9	93.2	68.8	81.0	84.5
Instruct-GPT$_{001}^{\text{curie}}$	2	85.2	51.1	68.2	90.1	21.9	56.0	67.3
	10	70.0	65.8	67.9	71.5	40.8	56.1	58.2
Instruct-GPT$_{001}$	2	78.1	83.6	80.9	94.9	25.0	60.0	57.7
	10	79.5	81.6	80.6	79.2	55.4	67.3	68.2
Instruct-GPT$_{002}$	2	81.7	**86.1**	83.9	92.9	48.7	72.1	71.2
	10	84.1	84.7	84.4	88.9	61.4	75.1	77.5
Instruct-GPT$_{003}$	2	87.9	81.3	84.6	95.1	58.1	76.6	80.5
	10	89.0	79.5	84.2	91.1	73.6	82.3	**87.9**
ChatGPT	2	82.9	82.0	82.4	89.8	69.8	79.8	79.2
	10	81.5	85.7	83.6	90.4	78.4	**84.4**	84.1

Table 1: Main results of different LLMs, which are obtained with k examples ($|E^+| = |E^-|$). **Cns.** denotes the consistency between QA and CG. The best results are **bolded** and the second best are underlined.

et al., 2020) ability of LLMs. We manually write 32 examples, with 16 examples for positive knowledge (denoted as E^+) and 16 for negative knowledge (E^-).[6] In the experiments, we randomly sample a total number of k examples from E^+ and E^-, where $|E^+| = |E^-|$ if not specified.[7]

Choices of LLMs We use LLMs that can do in-context learning, so that models stay fixed during probing. We choose Flan-T5 (Chung et al., 2022), GPT-3 (175B, `davinci`; Brown et al., 2020) and GPT-3.5 series, *e.g.* Codex ($\geq$175B, `code-davinci-002`; Chen et al., 2021) and InstructGPT (Ouyang et al., 2022): all are capable of in-context learning. Flan-T5 is an encoder-decoder LLM with instruction tuning based on T5 (Raffel et al., 2020). Codex extends GPT-3 through code training and instruction fine-tuning, and InstructGPT extends Codex through further tuning of the instructions. In our experiments, we mainly explore GPT-3.5 models. We use the 6.7B variant of InstructGPT (`text-curie-001`) and the $\geq$175B variants, *i.e.*, `text-davinci-001` (tuned on instructions), `text-davinci-002` (tuned on code and instructions), and `text-davinci-003` (further tuned with reinforcement learning with human feedback, RLHF).[8] For deterministic predictions, all models use greedy decoding (temperature as 0.0)[9]. We use InstructGPT$_{002}$ as the default LLM for experiments due to its powerful capability and the fact that it has been extensively researched and applied as of the time of writing this paper. We also include the recent ChatGPT (OpenAI, 2022), which is built upon InstructGPT and trained with dialogue data and RLHF.

4.2 The Belief Conflict

We report the results of the probing tasks in Table 1 for LLMs with 2- and 10-shot in-context learning. Based on the results, we discover a clear conflict of LLMs, that LLMs behave inconsistently in QA and CG tasks on negative commonsense knowledge, which we term *belief conflict*. Such conflict manifests itself in two ways: the gap between TP and TN on the CG task, and the gap of TN between the QA and CG tasks. In general, belief conflicts exist across LLMs of various sizes and structures. Ablated results per relation is presented in Appendix B.3.

When specifically asked, LLMs can distinguish between positive and negative commonsense knowledge, as evidenced by stable and balanced scores for positive and negative splits in the QA task. For CG, LLMs seem to accurately generate sentences grounded in positive knowledge according to TP. However, they perform poorly in negative knowledge, even for the best-performing LLMs, *i.e.*, Codex$_{002}$, InstructGPT$_{002,003}$, as shown by the lower bars of the CG on the negative split.[10] Also, the inconsistency between QA and CG reflects this conflict, as the content generated by a trustworthy AI system should consistent and faithful to what it believes. We present a case study and error analysis in Appendix B.5.

Among these LLMs, InstructGPT$_{003}$ and ChatGPT achieve much better results than others. We assume that such improvements are probably a result of training LLMs with human feedback (*e.g.*,

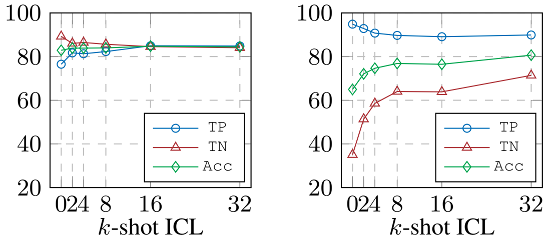
[6] Examples can be found in Appendix A.1

[7] Example prompts for two tasks are in Appendix A.2.

[8] `https://beta.openai.com/docs/model-index-for-researchers`

[9] We find our findings in the experiments are consistent for different temperatures, according to Appendix B.1.

[10] The only exception is GPT-3 (`davinci`). It scores poorly on the positive split with 10-shot learning, with TN exceeding TP. This happens when $k \geq 4$, while its 6.7B variant (`curie`) behaves consistently with others. Detailed results for GPT-3 are in Appendix B.2.



(a) Results (%) on QA. (b) Results (%) on CG.

Figure 3: Performance change for InstructGPT$_{002}$ on both tasks as the number of example (k) increases.

RLHF) based on the disclosed differences between them by OpenAI. Another evidence is that the recent ChatGPT also expresses great capabilities of generating negative knowledge, even better than InstructGPT$_{003}$ in this regard. We hypothesize that this is because negative knowledge and rebuttal statements are frequently used in human feedback to steer the model, *e.g.*, admitting errors or instructing the model not to do something. To validate this claim, future work could conduct more rigorous comparisons on public available LLMs, which would be an interesting research problem to trace certain abilities of LLMs to a specific period of training.

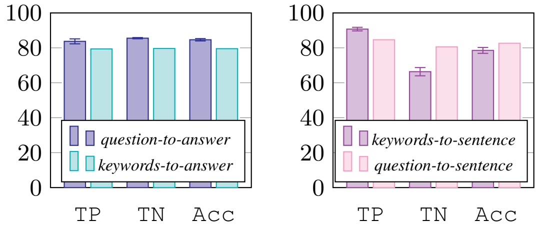
Sensitivity to the Number of In-Context Examples To find whether adding more examples helps solve the probing tasks, we increase the in-context examples from 0 to 32. Figure 3(a) shows a consistent finding with previous results, that LLMs are so good at answering yes or no questions that the number of examples does not affect much of the QA performance. Figure 3(b) shows that, adding more examples helps generate both positive and negative commonsense knowledge. However, the gap between `TP` and `TN` in the CG task still exists.

5 Analysis on the Belief Conflict

5.1 *Could keywords as task input hinder the manifestation of LLMs' belief?*

The task input difference for CG and QA leads to a concern that LMs may find it easier to understand natural questions (QA) than keywords (CG); hence, the belief conflict. In response to this concern, we change the input of the two tasks. For example, the keywords-to-answer task takes the form as:

> *Can these keywords form a truthful common sense fact? Answer with yes or no.*
> **Keywords**: lion, located at, ocean
> **Answer**: no



(a) Results (%) on QA. (b) Results (%) on CG.

Figure 4: Results of InstructGPT$_{002}$ when switching the task inputs between *question* and *keywords*, where $k = 10$. Columns with error bars show the ranges of the influence brought by different instruction wordings.

As for the question-to-sentence task:

> *Answer the question by writing a short sentence that contains correct common sense knowledge.*
> **Question**: do lions live in the ocean?
> **Sentence**: lions don't live in the ocean.

Results In Figure 4(a), we see a 4-point performance decrease given *keywords* as input for QA, which is not significant in comparison, and the results on the positive and negative splits are as balanced as before. This implies that LLMs' imbalanced performance in CG is not due to the use of keywords as input. In Figure 4(b), CG performance is greatly improved given *question* as input, approximating QA results. Our assumption is that CG is basically transformed into QA, because the textual corpus has seen too many negated texts following a Boolean question and rephrasing it, *e.g.*, "*...? No, lions do not live in the ocean.*" To validate this, we provide LLMs with zero-shot question-to-sentence instructions, and check if the output sentences start with *yes* or *no* given an input question. If our assumption is correct, models without examples will be biased toward QA even with a question-to-sentence instruction. The results of models optimized for instructions show that: 84.58% of sentences generated by InstructGPT$_{002}$ begin with yes or no, and 80.28% for InstructGPT$_{003}$. With 10 examples, this number drops to less than 4%. Thus, these results confirms that question-to-sentence generation degenerates to the QA task.

As a result, we conclude that the keyword-to-sentence (CG) is an appropriate and challenging task to probe generative LLMs. Employing keywords as input does not impact LLMs' grasp of the task (Figure 4(a)), while using questions as input may produce shortcuts that obscure whether LLMs can generate texts of negative commonsense knowledge (Figure 4(b)). Even if we use different instruction wordings (instructions are at Appendix A.2), none escapes the belief conflict, as shown by the error bars in Figure 4. Additionally, this experiment brings up the problem of how LLMs encode commonsense knowledge. According to this experiment, commonsense knowledge seems to be stored in LLMs in the same manner as it is in the corpus. LLMs struggle to generalize them, as evidenced by the keyword inputs for negative knowledge that do not have a statistical shortcut from pre-training.

5.2 *Will the keyword co-occurrence within corpus affect LLMs' generation?*

LLMs are essentially statistical models. In this experiment, we investigate the influence of *word co-occurrence in the corpus* on the CG task, which is one of the most common statistical factors. We categorize the dataset into buckets based on keywords co-occurrence on naturally existing corpora such as OMCS (706K sentences, Singh et al., 2002) and Wikipedia (1M, a subset built by Gao et al. (2021)). The co-occurrence for each triple is calculated by $\frac{\sum_{i,j} \texttt{cooccur}(w_i, w_j)}{l_s l_o}$, where $w_i \in s, w_j \in o$, and l_s, l_o denote the word count of subject s and object o, discarding stopwords.

From Figure 5, we have an interesting finding that three of the best-performing LLMs from Table 1 suffer from a performance drop at the > 1000 bucket of the negative split (TN), the most frequent data bucket. In contrast, LLMs achieve the best performance this bucket on the positive split (TP). We conclude that the hard-to-generate negative knowledge for LLMs tend to be those in which they have seen many subjects and objects appear together. For example, *worm* and *bird* usually co-occur in sentences, but models tend to generate "*worms can eat birds.*" Such statistical shortcuts hinder the generation of negative knowledge. This is also validated by TP results, where LLMs find it easy to generate sentences with frequently co-occurring entities in a positive fact.

5.3 *How does the balance of positive and negative examples affect negation bias?*

A possible answer for the difference between CG and QA is that: LMs suffer from reporting bias of negation during pre-training, while answering questions with yes or no is quite balanced in the corpora. We validate this problem by mitigating the negation bias through adjusting the examples of positive and negative cases. With more E^-s, LLMs are encouraged to generate more negations.

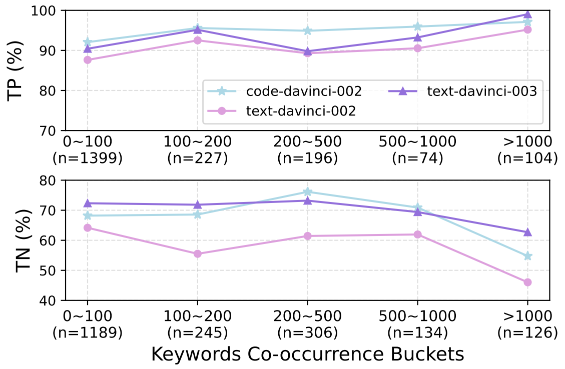


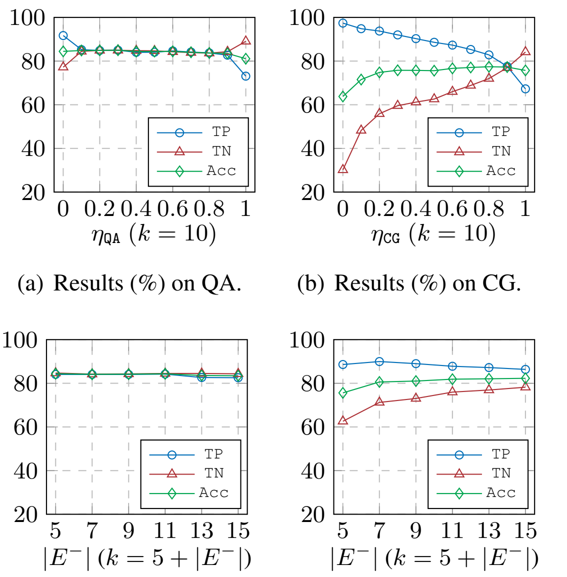
Figure 5: 10-shot CG results of three best-performing LLMs on different co-occurrence buckets. $a \sim b$ denotes that keywords co-occurrence in a bucket ranges from a to b. n is the number of triples in a bucket.

Results Figure 6(a), 6(b) adjust the ratio $\eta = \frac{|E^-|}{k}$ while fixing k. Figure 6(a) shows that InstructGPT$_{002}$ is very resilient against the example ratio in the QA task, except for extreme cases where only E^+s or E^-s are presented (*i.e.*, $\eta \in \{0, 1\}$). This also demonstrates the robustness of adopting QA results as LLMs' belief. In Figure 6(b), the CG performance on the negative split is improving as η grows. The turning point appears somewhere near $\eta \in (0.9, 1)$ when E^- takes over all the examples. Also, TP drops as E^+ becomes less. What if we add E^- without dropping E^+? In Figure 6(c), 6(d), we keep E^+ as constant ($|E^+| = 5$) and increase $|E^-|$ from 5 to 15. With enough amount of E^+, TN to CG continues to increase without sacrificing TP.

Overall, Figure 6 presents the possibility that we can overcome the belief conflict brought about by reporting bias by increasing negated texts in the training data or in-context examples. However, this is not always feasible in practice.

5.4 *Do Chain-of-Thought help generate texts with negative commonsense knowledge?*

Can the implicit reporting bias be overcome by explicit reasoning? Recent studies (Wei et al., 2022b,a) discover that the Chain-of-Thought (CoT) prompting technique shows the emergent reasoning abilities of LLMs. CoT generates intermediate steps in natural language, extending <input, output> to <input, *chain-of-thought*, output>. We adopt two instances of CoT: deductive reasoning and fact comparison, whose examples are manually written,



(a) Results (%) on QA.

(b) Results (%) on CG.

(c) Results (%) on QA.

(d) Results (%) on CG.

Figure 6: Results of InstructGPT$_{\texttt{002}}$ as the numbers of E^+ and E^- change. Figure (a) and (b) increase $\eta = |E^-|/k$ while fixing $k = 10$. Figure (c) and (d) add more E^- while fixing $|E^+| = 5$.

which are in Appendix A.1.

Deductive Reasoning Prompting We instantiate CoT with deductive argumentation in the form of *syllogism* (two premises and one conclusion). The prompt is extended into <input, "*Let's think step by step*: ...", output> with intermediate steps. A natural way to identify a negative proposition is deductive reasoning with *modus tollens*, *i.e.*, denying the consequent (Speranza and Horn, 2010; Bobzien, 2020): "If P then Q. Not Q. Therefore, Not P." For example, "*If something is a intelligent being (P), then it must have the ability to think (Q). Computers cannot think (Not Q). Therefore, computers are not intelligent beings (Not P).*"

To reason about positive propositions, we use *modus ponens* logic, *i.e.*, affirming the antecedent (Bobzien, 2020): "If P then Q. P. Therefore, Q." For example, "*Things with lightweight bodies and strong wing muscles (P) can usually fly (Q). Birds have these physical characteristics (P). Therefore, birds can fly. (Q)*" Notice that the deduction is not strictly logical but is enough to arrive at commonsense knowledge.

Fact Comparison Prompting Deduction emphasizes the intensional aspects of the fact, whereas fact comparison highlights the extensional comparison between counterpart facts (Fitting, 2006). For example, the related fact for "*lions do not live in the ocean*" is "*lions live in the land*". A negative fact often comes with a core fact that is true, which has been shown to be useful in explaining why a claim is wrong (Cheng et al., 2022). Therefore, we extend the <input, output> in each example by <input, "*Related fact*: ...", output>. For positive cases, we write a related fact for consistent examples.

Model	CoT	$k = 2$ (1:1)			$k = 10$ (1:1)		
		TP	TN	Acc	TP	TN	Acc
Codex$_{\texttt{002}}$	None	**96.6**	38.0	67.3	**93.2**	68.8	81.0
	Deduction	86.9	**56.6**	71.7	83.5	73.0	78.3
	Fact	92.9	53.7	**73.3**	86.8	**76.6**	**81.7**
Instruct-GPT$_{\texttt{002}}$	None	**92.9**	51.4	72.1	**88.9**	61.4	75.1
	Deduction	87.0	**57.3**	72.1	84.3	**70.7**	**77.5**
	Fact	89.1	55.5	**72.2**	85.5	69.2	77.4

Table 2: Performance on the CG task when enhanced with different types of CoT prompting, *i.e.*, deductive argumentation (*Deduction*) and fact comparison (*Fact*).

Results Table 2 displays the results of Codex$_{\texttt{002}}$ and InstructGPT$_{\texttt{002}}$. Both CoT instances improve LLMs' performance on TN, showing the benefit of explicit reasoning for deriving negative knowledge, where different models prefer different rationales. However, the increase in TN comes at the expense of a performance drop in TP. This is mostly because models previously predicted most of the cases to be positive, making TP irrationally high. Overall, these results suggest that, even though LLMs picked up implicit bias during pre-training, it can be overcome by making the reasoning chain explicit.

Nevertheless, deductive reasoning seems to be more rigid about confirming commonsense knowledge with a lower TP. This can be attributed to the fact that commonsense knowledge contains exceptions (Allaway et al., 2022), *e.g.*, *birds can fly but penguins can't*. Thus, LLMs with deductive reasoning may hold concerns about exceptions for confirming a commonsense fact, leading to a significant lower TP than fact comparison. We conduct a simple experiment of exceptions in Appendix B.4, which shows that adding adverbs of degree (*e.g.*, *usually*, *generally*) in the texts alleviates the belief conflict, but the problem still exists.

6 Closing Remarks

In this study, we explored and quantified the limitations of LLMs in generating texts grounded in negative commonsense knowledge that they seem to know, a phenomenon we term as "belief conflict". To investigate this, we probe LLMs with a constrained sentence generation (CG) task, coupled with a QA task. Our experiments demonstrated the existence of the belief conflict in all LLMs when it comes to negative knowledge, which is mostly brought by quantifiable statistical shortcuts such as keywords co-occurrence. We also see that this can be lessened by giving more in-context examples of negative knowledge or by using a chain-of-thought (CoT) prompting method to explain the explicit reasoning process for deriving negative knowledge.

With the rapid increase of the study on language-based reasoning (Clark et al., 2020; Tafjord et al., 2021; Wei et al., 2022b), there would be cause for concern if LLMs have trouble generating proofs or reasoning steps with negative knowledge. With all the good scores they achieve at QA tasks, whether they can be trusted with their knowledge expressed during generation, which is one of the most prominent way of human-AI interaction, is still questionable. In this sense, the study of negative knowledge creates a good testbed for assessing real language-based reasoning skills for LLMs without the statistical heuristics they memorized. We hope that the findings in this work could raise the awareness of the community on negative knowledge for LLMs in downstream text generation tasks.

Limitations

In this work, we highlight that the probing tasks are placed in the commonsense domain that are generally acknowledged by people in most situations. We do not consider the exceptions of commonsense knowledge, which has gradually drawn some research attentions (Do and Pavlick, 2021; Allaway et al., 2022). Exceptions are important for negative knowledge and are widely used in tasks such as argumentation or deductive reasoning. However, in the experiments, we find that such exceptions might make models generate commonsense statements with uncertain adverbs (*e.g.*, *may*, *some*, etc.) on rare cases.

Another limitation of this work is that the probing task is based only on relational commonsense knowledge from commonsense knowledge bases such as ConceptNet. We design the keyword-to-sentence task mostly for the purpose of convenient evaluation for text generation, which is notoriously known as difficult. The probing and evaluation of LLMs' belief about negative knowledge in more complex tasks are beyond the scope of this work, but really interesting and challenging. Also, other types of knowledge could be studied in a similar way, such as negative social, temporal and spatial knowledge, to name but a few.

In this paper, we identify the belief conflict problem in LLMs through extensive experiments. Future work could explore more advanced training or prompting-based methods to improve the consistency between a model's belief and its actions (text generation for various tasks), especially for negative knowledge.

Ethical Statement

The commonsense knowledge triples from ConceptNet may include offensive and biased sentences, which may also exist in the dataset that we use in this work. As stated before, the identification of commonsense negative knowledge may slightly vary from people from different cultural and social background when considering exceptions.

Acknowledgement

We thank the anonymous reviewers for their valuable comments. We also thank Siyu Yuan and Jian Xie from Fudan University, and Kexun Zhang, Yujian Liu, Qingxiu Dong and Xuandong Zhao from UC Santa Barbara for their useful suggestions and discussions for the manuscript. This research is funded by the Science and Technology Commission of Shanghai Municipality Grant (No. 22511105902).

References

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

Emily Allaway, Jena D Hwang, Chandra Bhagavatula, Kathleen McKeown, Doug Downey, and Yejin Choi. 2022. Penguins don't fly: Reasoning about generics through instantiations and exceptions. *arXiv preprint arXiv:2205.11658*.

Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. 2021. Negative knowledge for open-world wikidata. In *Companion Proceedings of the*

Web Conference 2021, WWW '21, page 544–551, New York, NY, USA. Association for Computing Machinery.

Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. 2022. Uncommonsense: Informative negative knowledge about everyday concepts. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 37–46, New York, NY, USA. Association for Computing Machinery.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Stephen Barker and Mark Jago. 2012. Being positive about negative facts. *Philosophy and Phenomenological research*, pages 117–138.

Susanne Bobzien. 2020. Ancient Logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Summer 2020 edition. Metaphysics Research Lab, Stanford University.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Ruben Branco, António Branco, João António Rodrigues, and João Ricardo Silva. 2021. Shortcutted commonsense: Data spuriousness in deep learning of commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1521, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. Knowledgeable or educated guess? revisiting language models as knowledge bases. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874, Online. Association for Computational Linguistics.

Jiangjie Chen, Rui Xu, Ziquan Fu, Wei Shi, Zhongqiao Li, Xinbo Zhang, Changzhi Sun, Lei Li, Yanghua Xiao, and Hao Zhou. 2022. E-KAR: A benchmark for rationalizing natural language analogical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3941–3955, Dublin, Ireland. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Sijie Cheng, Zhiyong Wu, Jiangjie Chen, Zhixing Li, Yang Liu, and Lingpeng Kong. 2022. Unsupervised explanation generation via correct instantiations. *arXiv preprint arXiv:2211.11160*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

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

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

Nam Do and Ellie Pavlick. 2021. Are rotten apples edible? challenging commonsense inference ability with exceptions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2061–2073, Online. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Melvin Fitting. 2006. Intensional logic.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Reto Gubelmann and Siegfried Handschuh. 2022. Context matters: A pragmatic study of PLMs' negation understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4602–4621, Dublin, Ireland. Association for Computational Linguistics.

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

Laurence R. Horn and Heinrich Wansing. 2022. Negation. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Winter 2022 edition. Metaphysics Research Lab, Stanford University.

Md Mosharaf Hossain, Dhivya Chinnappa, and Eduardo Blanco. 2022. An analysis of negation in natural language understanding corpora. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 716–723, Dublin, Ireland. Association for Computational Linguistics.

Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. Understanding by understanding not: Modeling negation in language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1301–1312, Online. Association for Computational Linguistics.

Joel Jang, Seonghyeon Ye, and Minjoon Seo. 2022. Can large language models truly understand prompts? a case study with negated prompts. *arXiv preprint arXiv:2209.12711*.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations. *arXiv preprint arXiv:2205.11822*.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. 2021. BeliefBank: Adding memory to a pre-trained language model for a systematic notion of belief. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8849–8861, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuxuan Lai, Chen Zhang, Yansong Feng, Quzhe Huang, and Dongyan Zhao. 2021. Why machine reading comprehension models learn shortcuts? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 989–1002, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. 2021. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):6418–6425.

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

Charles MacDonald. 1965. The role of negation in human knowledge. *Laval théologique et philosophique*, 21(1):80–114.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022a. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Marvin Minsky. 1997. Negative expertise.

George Molnar. 2000. Truthmakers for negative truths. *Australasian Journal of philosophy*, 78(1):72–86.

OpenAI. 2022. Chatgpt.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

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

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Abhilasha Ravichander, Matt Gardner, and Ana Marasović. 2022. Condaqa: A contrastive reading comprehension dataset for reasoning about negation. *arXiv preprint arXiv:2211.00295*.

Ehud Reiter. 2019. Natural language generation challenges for explainable ai. *arXiv preprint arXiv:1911.08794*.

Kyle Richardson, Ronen Tamari, Oren Sultan, Reut Tsarfaty, Dafna Shahaf, and Ashish Sabharwal. 2022. Breakpoint transformers for modeling and tracking intermediate beliefs. *arXiv preprint arXiv:2211.07950*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Tara Safavi, Jing Zhu, and Danai Koutra. 2021. NegatER: Unsupervised Discovery of Negatives in Commonsense Knowledge Bases. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5633–5646, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, pages 1223–1237, Berlin, Heidelberg. Springer Berlin Heidelberg.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

J.L. Speranza and Laurence R. Horn. 2010. A brief history of negation. *Journal of Applied Logic*, 8(3):277–301.

Theodore R Sumers, Robert D Hawkins, Mark K Ho, and Thomas L Griffiths. 2021. Extending rational models of communication from beliefs to actions. *arXiv preprint arXiv:2105.11950*.

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

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2022. Entailer: Answering questions with faithful and truthful chains of reasoning. *arXiv preprint arXiv:2210.12217*.

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

Bing Tian, Yixin Cao, Yong Zhang, and Chunxiao Xing. 2022. Debiasing nlu models via causal intervention and counterfactual reasoning.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

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

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2022. A survey of knowledge-enhanced text generation. *ACM Computing Surveys (CSUR)*.

Task 1: Boolean Question Answering (QA)

Answer the commonsense questions with yes or no:
/* Examples */
Question: can birds fly?
Answer: yes
###
Question: is water spicy?
Answer: no
/* Test data */
Question: are needles used for writing?
Answer: no

Task 2: Constrained Sentence Generation (CG)

Write a short and factual sentence according to commonsense based on the keywords:
/* Examples */
Keywords: birds, capable of, fly
Sentence: birds can fly.
###
Keywords: water, has property, spicy
Sentence: water isn't spicy.
/* Test data */
Keywords: needles, used for, writing
Sentence: needles are not used for writing.

Table 3: Example prompts of the two probing tasks for in-context learning, which consists of a task instruction at the beginning and several in-context examples. Underlined texts denote the model completion.

A Demonstrations for In-Context Learning

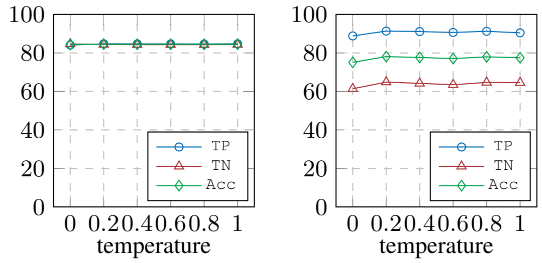
A.1 Manually-written Examples for In-Context Learning

Some of the manually designed examples are shown in Table 6.

A.2 Example Prompts for the Probing Tasks

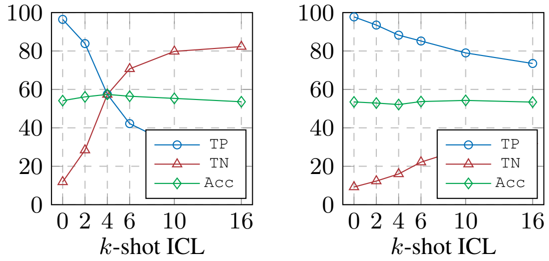
The task inputs to the LLMs are presented in Table 3. Note that *instructions* can be replaced by others. LLMs with in-context learning are known to be sensitive to the wording and examples in the prompts (Min et al., 2022b). Therefore, we manually write 4 interchangeable instructions for each probing tasks. For the QA task, the instructions include:

1. *Answer the commonsense questions with yes or no.*
2. *Choose "yes" or "no" to indicate whether you agree or disagree with the commonsense questions.*
3. *Respond to the questions using "yes" or "no".*
4. *Indicate whether the commonsense questions are correct or incorrect by writing "yes" or "no".*



(a) Results (%) on QA. (b) Results (%) on CG.

Figure 7: Performance change for InstructGPT$_{002}$ on both tasks as the temperature changes.



(a) GPT-3 `davinci` (175B) (b) GPT-3 `curie` (6.7B)

Figure 8: CG results of GPT-3 for the `davinci` (175B) and `curie` (6.7B) variants, where $|E^-| = |E^+|$. Unlike other LLMs, the `TN` of GPT-3 surpasses `TP` when $k \geq 4$.

For the CG task, the instructions include:

1. *Write a short and factual sentence according to commonsense based on the keywords:*
2. *Use the keywords to create a short and factual sentence that accurately reflects commonsense knowledge.*
3. *Create a short, factual sentence based on the keywords and what is generally accepted as true.*
4. *Construct a factual and concise statement based on the provided keywords and commonsense knowledge.*

B Additional Results

B.1 Sensitivity to Temperature Tuning

Figure 7 shows that temperature does not influence much of the performance, thus the findings of this paper are not sensitive to temperature tuning.

B.2 Abnormal Results of GPT-3 (`davinci`)

Different from the trends of other LLMs reported in § 4.2, GPT-3 `davinci` shows a confusing pattern of the results on the CG task. A more de-

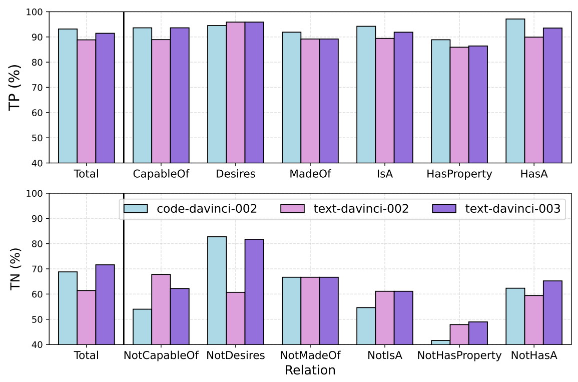


Figure 9: The 10-shot CG results per relation type on CSK-PN. The results are obtained with 10-shot learning. n denotes the triple number per relation.

tailed experiment in Figure 8(a) shows that, when $k < 4$, GPT-3 (davinci) performs similarly with its sibling LLMs, with TP greatly surpasses TN. TN continues to enlarge as k increases, even beating TP. Based on Acc over the whole dataset, GPT-3 does not achieve results as good as other GPT-3 derivatives. However, a smaller version of GPT-3 (*i.e.*, curie, 6.7B) does not express such pattern, according to Figure 8(a). We do not have proper reasons for this finding, but further training on code and instruction tuning (*i.e.*, Codex and InstructGPT) seem to fix this problem.

B.3 Results of Different Relation Types

What types of relations do LLMs find the most difficult to verbalize? As seen in Figure 9, we see LLMs achieve good results in the positive split. On the negative split, LLMs unanimously believe NOTHASPROPERTY to be the most difficult relations.

B.4 *Do LLMs hold concerns about exceptions for commonsense knowledge?*

Commonsense knowledge usually comes with exceptions. Could LLMs answer or generate commonsense knowledge incorrectly be because they are thinking about exceptions? For example, "*birds can fly, but penguins cannot.*" (Allaway et al., 2022). So when asked "*can birds fly?*", LLMs may think of a counterexample and thus arrive at the answer *no*. We rephrase the in-context examples by adding *adverbs of degree* (*e.g.*, *typically*, *generally*, *usually*, *most*, etc.) to make the tasks be about the commonsense instead of exceptions. For instance, we rewrite "*can birds fly?*" into "*can most birds fly?*" or "*can birds generally fly?*", and "*lions don't live in the ocean.*" into "*lions don't usually live in the ocean.*" In this way, we make language explicitly convey uncertainty (Reiter, 2019) and try to rule out exceptions in the tasks.

Model	Exception	Perf. on QA			Perf. on CG		
		TP	TN	Acc	TP	TN	Acc
Codex$_{002}$	-	**88.1**	**81.8**	**84.9**	**93.2**	68.8	81.0
	✓	87.2	79.6	83.4	91.9	**72.2**	**82.1**
Instruct-GPT$_{002}$	-	**84.1**	84.7	84.4	88.9	61.4	75.1
	✓	84.0	**85.4**	**84.7**	**90.9**	**70.1**	**80.5**

Table 4: 10-shot QA and CG results of LLMs when adding adverbs of degree into texts, making them somehow consider exceptions of commonsense knowledge.

Well generated negated sentence		
1	Triple Label Generation	<deer, desires, be shoot by hunter> negative Deer do not desire to be shot by hunters.
Weak negation		
2	Triple Label Generation	<person, desires, eat alone> negative Some people desire to eat alone.
Unfaithful generated sentence		
3	Triple Label Generation	<student, desires, exam> negative Students generally desire to do well on exams.
Wrong data label		
4	Triple Label Generation	<horse, is a, bird> positive horses are not birds.
High co-occurrence words		
5	Triple Label Generation	<worm, capable of, eat bird> negative Worms can eat birds.

Table 5: Examples of the generated texts by InstructGPT$_{002}$ on the CG task.

Based on the results in Table 4, we find that adding adverbs of degree to the texts does improve LLMs' performance on both CG and QA. This suggests that LLMs do hold a certain amount of concerns toward exceptions when dealing with commonsense reasoning, especially for negative knowledge. However, considering exceptions with this trick still does not resolve the belief conflict. Also, this approach could also serve as a useful trick for future commonsense research.

B.5 Case Study

Table 5 presents some examples of generated by InstructGPT$_{002}$ (10-shot). In the 1st case, the model correctly generated negative commonsense sentences. The 2nd one suffers from the problem of weak negation, *i.e.*, for negative triple, the model sometimes use “may” or “some” for weak negation, which is not detected by the negation cue detector metric. The 3rd one suffers from unfaithful generation to the constraints, where the model generates information outside the input triples to avoid generating negation. The 4th one is wrong due to the noise in the dataset. The 5th one is probably due to the high co-occurrence of the concept *worms* and *birds*, the model finally generates a positive sentence.

		Examples for Positive Commonsense Knowledge
1	Triple	<birds, capable of, fly>
	Sentence	Birds can fly.
	Question	Can birds fly?
	Deduction	Things with lightweight bodies and strong wing muscles can usually fly. Birds have these physical characteristics.
	Fact	Birds have wings.
2	Triple	<playing tennis, causes, feeling relaxed>
	Sentence	Playing tennis makes one feel relaxed.
	Question	Does playing tennis cause someone to feel relaxed?
	Deduction	Sport can make people feel relaxed. Tennis is a kind of sport.
	Fact	Tennis is a kind of sport.
3	Triple	<basketball players, desires, winning>
	Sentence	Basketball players want to win.
	Question	Do basketball players want to win?
	Deduction	Winning is an important goal for many athletes. Basketball players are athletes.
	Fact	Athletes usually desire winning in competitions.
4	Triple	<people, desires, relax after work>
	Sentence	People want to relax after work.
	Question	Do people want to relaxed after work?
	Deduction	Tired people want to relax. Work makes people tired.
	Fact	People will be tired after work.
5	Triple	<sheepskin, used for, writing>
	Sentence	Sheepskin can used for writing.
	Question	can sheepskin be used for writing?
	Deduction	Things with a smooth and consistent surface can be used for writing. Sheepskins have that texture.
	Fact	Sheepskin is the hide of a sheep.
		Examples for Negative Commonsense Knowledge
1	Triple	<shoes, has a, sleeves>
	Sentence	Shoes have no sleeve.
	Question	Do shoes have sleeves?
	Deduction	Sleeves are parts of garments that cover the arms. Shoes are not garments.
	Fact	Shoe is a type of footwear.
2	Triple	<banana, is a, tree>
	Sentence	Bananas are not trees.
	Question	Are bananas a kind of trees?
	Deduction	If something is a tree, then it has an elongated trunk. Bananas do not have elongated trunks.
	Fact	bananas are a type of fruit.
3	Triple	<computer, is a, intelligent being>
	Sentence	Computers aren't intelligent beings.
	Question	Is a computer an intelligent being?
	Deduction	Intelligent beings have the ability to think. Computers cannot think like humans do.
	Fact	Computer is a type of electronic device.
4	Triple	<guns, used for, healing>
	Sentence	Guns can't be used for healing.
	Question	Are guns used for healing?
	Deduction	Healing instruments are tools that are used to treat injuries or illnesses. Guns are not tools that are used to treat injuries or illnesses.
	Fact	Guns are used for killing.
5	Triple	<elephant, capable of, jump>
	Sentence	Elephants cannot jump.
	Question	Can elephants jump?
	Deduction	Jumping needs sufficient force to overcome the effects of gravity. Elephants are too heavy to overcome gravity.
	Fact	elephants can walk slowly.

Table 6: Some of the manually written examples used in in-context learning.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations

☑ A2. Did you discuss any potential risks of your work?
Ethical Statement

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☑ A4. Have you used AI writing assistants when working on this paper?
Grammarly, Quillbot. For grammar check and writing polish.

B ☑ Did you use or create scientific artifacts?

Section 3.1

☑ B1. Did you cite the creators of artifacts you used?
Section 3.1

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Open-sourced resource.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 3.1

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section 3.1

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Open-sourced resource.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3.1

C ☑ Did you run computational experiments?

Section 4, Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4.1

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3.2

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.