Using Commonsense Knowledge to Answer Why-Questions

Yash Kumar Lal* Stony Brook University
Niket Tandon Allen Institute for AI
Tanvi Aggarwal Stony Brook University
Horace Liu Stony Brook University

Nathanael Chambers US Naval Academy
Raymond Mooney University of Texas at Austin
Niranjan Balasubramanian Stony Brook University

Abstract

Answering questions in narratives about *why* events happened often requires commonsense knowledge external to the text. What aspects of this knowledge are available in large language models? What aspects can be made accessible via external commonsense resources? We study these questions in the context of answering questions in the TELLMEWHY dataset using COMET as a source of relevant commonsense relations. We analyze the effects of model size (T5 variants and GPT-3) along with methods of injecting knowledge (COMET) into these models. Results show that the largest models, as expected, yield substantial improvements over base models and injecting external knowledge helps models of all sizes. We also find that the format in which knowledge is provided is critical, and that smaller models benefit more from larger amounts of knowledge. Finally, we develop an ontology of knowledge types and analyze the relative coverage of the models across these categories.[1]

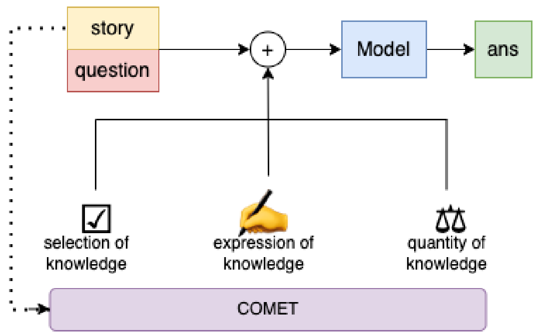

Figure 1: This paper systematically studies how to utilize commonsense knowledge to answer why-questions and their interaction with models M of different sizes.

1 Introduction

Humans reason about events in narratives by making inferences about why those events happen. The recently introduced TELLMEWHY dataset tests for this capability by posing *why* questions over events in simple narratives (Lal et al., 2021). Answering these often requires commonsense knowledge (CSK) that is not explicitly stated as part of the narratives. Indeed, QA models built over standard *base* sized models fare poorly, especially where the answer is not stated in the narrative.

There are two broad avenues for incorporating the necessary commonsense knowledge for this task — using larger language models (e.g. T5-11B (Raffel et al., 2020)) and leveraging external knowledge resources. The former can be seen as an implicit approach, where we tap knowledge that is acquired via language modeling and general QA task pretraining. The latter is an explicit approach where we inject knowledge from a resource as part of the context. We start by asking three questions that can inform future research along these avenues: **(1)** What aspects of commonsense knowledge are already accessible to larger language models? **(2)** What aspects can be made accessible by injecting information from relevant knowledge sources? **(3)** What kinds of knowledge remains inaccessible?

For the TELLMEWHY task, we explore the utility of COMET[2] (Bosselut et al., 2019; Hwang et al., 2021) as a knowledge source. COMET is a transformer-based model that generates commonsense inferences about events that it has learned from ATOMIC (Sap et al., 2019; Hwang et al., 2021) and ConceptNet (Speer et al., 2017). However, the automatically generated knowledge may contain incorrect or irrelevant inferences.

We start by exploring multiple ways of integrating this kind of knowledge into a QA model. First, we experiment with the best way of *selecting* relations from COMET that should be added to model input. We find that adding diverse types

*Corresponding author: ylal@cs.stonybrook.edu

[1]We make the relevant code and data available at https://github.com/StonyBrookNLP/knowwhy

[2]We use COMET 2020 for our experiments.

of relations helps the most. Next, we investigate various ways to integrate this knowledge into a model's input. While COMET relations are usually words or phrases, converting them to sentences using simple verbalization templates works the best. Finally, we analyze the *amount* of external knowledge needed by models of various sizes. Smaller models benefit more from a larger amount of knowledge while larger models do well with less external knowledge.

These findings are used to build models of multiple sizes that can use external commonsense knowledge for the TELLMEWHY task. We use diverse types of information from COMET converted into fluent sentences as part of the model inputs. For small models, we supply more commonsense knowledge to boost model performance while larger models are given less knowledge.

To analyze the relative merits of all these approaches, we manually categorized the Why questions according to the types of knowledge needed to answer them. We find that most questions target Consequence, Goal seeking, Desire, and Reactionary knowledge types. We categorize the rest as Other and analyze the performance of different models across these knowledge categories. Our analysis shows that models seem to particularly lack the ability to understand and utilize 'Goal seeking' knowledge.

In summary, our contributions are:

1. A systematic analysis of different aspects of injecting commonsense knowledge for answering why questions and their interaction with models of different sizes
2. Developing an approach based on this analysis to achieve a new state-of-the-art result on the TELLMEWHY dataset, and an addition of human judgments for answers to it
3. An analysis of types of knowledge that are not adequately captured by current models.

2 Overview: Task and Models

This section gives an overview of the data and evaluation scheme, and defines a formulation to describe the model configurations we investigated.

2.1 Task

TELLMEWHY (Lal et al., 2021) is a dataset of 30k questions and free-form answers concerning why characters in short English narratives perform the actions described. It is built on the ROCStories corpus (Mostafazadeh et al., 2016). The questions are created by applying templates over events described in the narratives, and the answers are crowd-sourced from MTurk. Each question has 3 (possibly different) human answers. The dataset contains both explicit-answer questions (**EXPL**; there is a possible answer to the question in the narrative) and implicit-answer questions (**IMPL**; the answer is not in the narrative, so external knowledge and/or reasoning is needed). Dataset statistics are presented in Table 9, and an example can be found in Table 11.

2.2 Model Setup

For this task, we investigate a variety of model configurations that add commonsense knowledge to the input. The inputs to a given model follow the format:

$$\text{question: } Q \text{ [sep] context: } C \text{ [sep] } G(\text{CSK}^n_\Omega) \quad (1)$$

where Q represents the question and C denotes the context, CSK stands for the external commonsense knowledge being used, the function G indicates the input format of this CSK, n represents the number of CSK statements being used and Ω stands for the way the relevant knowledge is selected from all available knowledge.

For our experiments, we primarily use the T5 family of models (Raffel et al., 2020). T5 is a text-to-text model, which means it can be trained on arbitrary tasks involving textual input and output. T5 has achieved SOTA on many natural language understanding (NLU) tasks, including free-form question answering. We use HuggingFace (Wolf et al., 2020) for our models.

Small models We start with base-sized models, which we refer to as small models. This class of models is the most readily accessible and works with smaller compute resources. Lal et al. (2021) showed that small models struggle with answering why questions about events in narratives. Prior work (Bi et al., 2019; Xu et al., 2021b) has shown that adding relevant knowledge from external sources helps models answer contextual questions. For our investigation, we focus on T5-BASE, which is a 220 million parameter model.

Large models It has been shown that, as the size of the model increases, the ability of these models to perform NLP tasks improves. With the increase in the number of parameters, these models are better endowed with certain types of knowledge due

Question: Why was Kelsi excited to try out bright red hair?
COMET: Kelsi was excited to try out bright red hair. [HasSubEvent] [GEN]
Kelsi was excited to try out bright red hair. [DesireOf] [GEN]
...

Original	Reranked	Diverse
× get a wig (HasSubEvent)	✓ hair color (DesireOf)	✓ to be fashionable (Desires)
✓ hair color (DesireOf)	✓ to be fashionable (Desires)	✓ get red hair dye (HasSubEvent)
✓ to be fashionable (Desires)	✓ get red hair dye (HasSubEvent)	✓ hair dye (DesireOf)
✓ get red hair dye (HasSubEvent)	✓ hair dye (DesireOf)	✓ Kelsi's hair is too dark (HinderedBy)
✓ hair dye (DesireOf)	× get a wig (HasSubEvent)	✓ gets complimented (oEffect)

Table 1: An example of how to extract (potentially noisy) COMET knowledge for a question in TellMeWhy, and two re-ranking approaches to reduce the noise in the generated knowledge. The top part shows how COMET is prompted (its input, including GEN) with an event-centric relation type and the sentence from which the question was created to generate the knowledge. Some resulting phrases are shown in the bottom half of the table according to different ranking methods. These are verbalized according to templates in Table 10 before being fed into the QA model.

to pretraining (Petroni et al., 2019). To investigate the performance of large models, we use the T5-11B model. This 11 billion parameter model requires significant compute resources.

Very large models Brown et al. (2020) showed that very large models have the ability to perform very well on a variety of natural language understanding tasks even in zero- and few-shot settings. Furthermore, PaLM (Chowdhery et al., 2022) and LaMDA (Thoppilan et al., 2022) have shown that these models can achieve comparable performance to state-of-the-art finetuned models even when used in zero- and few-shot settings (Wei et al., 2022). For our experiments, we use the GPT-3 API by OpenAI to run zero-shot experiments. GPT-3 has around 175 billion parameters.

2.3 External Knowledge

We use COMET (Hwang et al., 2021) as our source of external commonsense knowledge to integrate pertinent information into the models. This knowledge is represented through CSK in Equation 1. Such autogenerated knowledge may contain incorrect or irrelevant inferences.

For the sentence in the narrative used to create a question, we generate 3 relation phrases of different types from COMET. We focus on relation types (see Table 10 for full list of relation types used) about people (social interaction) and events (event-centered). COMET also provided a score for each generated relation.

When investigating the best approach to using COMET, we need a ranking of the relations according to their relevance to the associated story and question. We calculate the BertScore (Zhang* et al., 2020) between the output for each relation and all gold answers for a question. The resulting list of relations sorted by the described BertScore value is considered to be the gold ranking. We hypothesize that this is the kind of knowledge the model needs to answer the question correctly.

2.4 Human Evaluation Metric

We use the human evaluation templates and MTurk settings provided by Lal et al. (2021) to collect judgments for models' predicted answers on the hidden test set. We asked the annotators whether, given the story and its associated question, the answer shown to them was valid. Each answer is evaluated by 3 annotators on a 5-point Likert scale (-2 to 2)[3]. We use the average Likert score over all answers as a performance metric (Liddell and Kruschke, 2018). The maximum score possible is 2, and the minimum is -2.

Running human evaluation is expensive and time-consuming. Additionally, slight variants of most large models tend to generate similar answers (when using beam search) for many questions. In order to improve time and cost efficiency, we implement a caching mechanism to re-use previous annotator judgments for the same answer for a question in a particular story. We have built a cache of ~7000 model-generated answers with human evaluations and make it available so that human evaluation on this dataset is easier in the future. More details are in Appendix A.

[3]Integer scores correspond to the labels: strongly disagree, disagree, neutral, agree, strongly agree.

3 Empirical Insights into Knowledge Integration

We instantiate the abstract model formulation described in Equation 1 with various knowledge integration approaches. We ask three questions about injecting external knowledge into models to improve why question answering. Our findings influence the choices we make when building the best possible model. We use the small and large models for our investigations in this section. Examples of each variant are shown in Table 1.

3.1 What Knowledge to Inject?

For each question, COMET is used to retrieve a list of possible commonsense relations across several types. Each relational inference comes with a score provided by COMET, but which of these best aids answering the why-question is an open question. This section investigates how to select which to use (Ω in Equation 1). We thus hard-code $n = 3$ and use ($G = \textit{verbal.}$) to explore Ω. The relations are verbalized according to the templates presented in Table 10. More details about G_{verbal} can be found in §3.2.

Intuitively, we want the external knowledge to help produce human-like answers. To this end, we calculate the BertScore of each COMET inference to human answers and use this as a gold ranking for the external knowledge we want to add.

- Ω = **COMET (original)** First, we use the scores from COMET in descending order as ranks for the relations. The QA model input is augmented with the top n relations according to these scores. Although using the COMET ranking is the most straightforward way to select relevant information, Table 2 shows that this approach performs poorly on Precision@k metrics.

Ranking model	P@3	P@5
COMET score	0.14	0.23
Pretrained MSMARCO	0.32	0.41
Finetuned MSMARCO	0.45	0.54

Table 2: Precision@k (P@k) scores to compare approaches to rank COMET. Our finetuned reranker significantly outperforms the default COMET ranking.

- Ω = **RERANKED-COMET (pre-trained MSMARCO)** We start by using an off-the-shelf pretrained ranking model: the msmarco-distilbert-dot-v5 model available on HuggingFace. The question and the narrative concatenated with the "[SEP]" symbol is treated as the query, and the associated relational inferences are treated as the documents in this setting. We compute the cosine similarity between the query and the inferences to rank the latter. As shown in Table 2, this significantly improves the P@k performance over ranking just using COMET scores

- Ω = **RERANKED-COMET (fine-tuned)** We finetuned the prior pretrained ranking model to produce "silver" ranked relations as compared to the gold ranking. We use separate query and document encoders, each with frozen embeddings. The word embeddings are mean-pooled to obtained sentence-level representations for both the questions and the relations. We compute the cosine similarity between the query and COMET inferences to rank the latter. We use the pairwise ranking loss function with the aim of optimizing Precision@5 for the ranking. To do this, we generate pairs using positive examples for ranks 1 to 4, and use the other relations to generate negative examples. Table 2 shows that this finetuned ranking model is clearly the best for selecting COMET inferences to augment our QA models. More details about the ranking model are available in Appendix B. Going forward, we use RERANKED-COMET to refer to this model.

- Ω = **DIVERSE-COMET** Table 1 illustrates that the top scoring inferences according to COMET often involve the same relation types. Relational inferences of the same type are often semantically similar. We hypothesize that models would benefit more from diverse knowledge rather than similar, redundant knowledge. Therefore, we filter the list of COMET inferences to retain only the top inference for each relation, according to its COMET score. Finally, we take the top scoring relations from this filtered list.

Finding 1: Using ranked COMET inferences helps. Table 3 shows the results for all possible ways of using COMET inferences. We see that using external knowledge in any form, even ranking by COMET scores, helps compared to models without added knowledge. Furthermore, using

	T5-BASE		T5-11B	
	Full	Impl	Full	Impl
Original COMET	0.75	0.26	1.24	1.01
Diverse COMET	0.84	0.47	1.27	1.04
Reranked COMET	0.88	0.59	1.23	0.99

Table 3: What knowledge to inject? Both Diverse COMET and Reranked COMET yield similar results. We use Diverse COMET for the subsequent experiments since it shows improvement on the large model.

	T5-BASE		T5-11B	
	Full	Impl	Full	Impl
G_{tup}	0.52	0.08	1.24	0.97
G_{tupsep}	0.52	0.1	1.11	0.81
G_{verbal}	0.84	0.47	1.27	1.04

Table 4: How to inject knowledge? The best way to inject COMET inferences is to verbalize relational information as fluent sentences.

the top inferences from the reranking model improves over just using the top inferences according to COMET. We see that selecting diverse relations helps T5-11B the most; but using the reranking model helps T5-BASE the most. However, the *Diverse* COMET and *Reranked* COMET models perform similarly across both model sizes.

3.2 How to Express the Knowledge to Inject?

Task-specific knowledge can be used in different ways (Sahand Sabour, 2021; Xu et al., 2021a). We investigate several ways of integrating the knowledge from COMET ($G(.)$ in Equation 1) into the models for TELLMEWHY. We use $n = 3$ and $\Omega = Diverse$-COMET for these experiments.

- **G_{tup}:** This format uses special tokens (<info> and </info>) for inferences and relation types.
- **G_{tupsep}:** This format adds a "\n" token after each inference and its relation type encapsulated inside <info> tags. Each of these inferences is additionally separated by "\n".
- **G_{verbal}:** Prior work (He et al., 2021; Arabshahi et al., 2021) has shown that it helps to add external information in a fluent natural language. Motivated by this, we verbalize the inferences according to their relation type using the templates presented in Table 10.

See Fig. 5 for examples of these input formats.

Finding 2a: Relations as fluent sentences helps. Table 4 shows that commonsense in any format improves performance. Verbalizing COMET relations helps the most. Models are able to process this extra information better when it is expressed as fluent natural language sentences.

Finding 2b: Separator used is important. Prior work (Khashabi et al., 2020) highlighted the importance of separator tokens. In long texts such as ours, it helps the model distinguish between different portions of the input. We found that a clear separator token ($sep = \backslash n$) informs the model about the input segments and thus improves the performance of both small (T5-BASE performance improves from 0.36 to 0.58) and large models (T5-11B improvement from 0.99 to 1.21). Results are presented in Table 14.

	T5-BASE		T5-11B	
	Full	Impl	Full	Impl
top1	0.88	0.51	1.24	0.95
top3	0.84	0.47	1.27	1.04
top5	0.91	0.56	1.25	1.05

Table 5: How much knowledge to inject? Smaller models need more external knowledge to achieve optimal performance while larger models need less.

3.3 How much Knowledge to Inject?

We also investigate how the amount of knowledge added (n in Eq. 1) affects the performance of the model. We set $\Omega = Diverse$-COMET and use G_{verbal}.

Finding 3: Larger models need less knowledge. Table 5 shows the effects of adding different numbers of relations. Adding 5 relations helps T5-BASE the most, while T5-11B does best with 3 relations.

3.4 Injecting knowledge with GPT-3 prompts

To extend upon the insights of Finding 3, we also experiment with a very large model (GPT-3), which performs well on many NLP problems but may still exhibit a lack commonsense (Bender and Koller, 2020). With the right *prompts*, very large models have been shown to work well even in a zero-shot setting (Ouyang et al., 2022) because they may already encode much of the information needed to perform the task. We prompt GPT-3 with the narrative context (N), the question (Q) and the knowledge (CSK) and the model autoregressively generates a sequence. Our use of knowledge in the prompt is a form of "prompt engineering", where GPT-3's behavior is modified by enhancing the prompt (Le Scao and Rush, 2021). We enhance the input by simply injecting commonsense that nudges the model towards the correct answer. See Table 12 for examples of different prompts.

Unlike finetuning, in a zero-shot setting, the model has no opportunity to learn when and how to apply CSK. Therefore, it is imperative to inject CSK into the prompt in the best possible manner. We experimented with providing CSK before N (prefix), after N (postfix), and finally by inserting CSK after the sentence from which Q was created (infix). Infix injection works best because it allows the model to encode the sentence of interest with a richer context. Postfix injection forces the autoregressive model to pay more attention to potentially noisy CSK, and prefix injection leads to the knowledge being often ignored perhaps due to the distance from the question. Thus, we chose infix injection as the preferred prompting approach.

4 Distilling the Empirical Insights: The KnowWhy Approach

Finally, we combine our findings to create the KnowWhy approach. We use it to build the best possible models of all sizes — small, large and very large — for the TellMeWhy task. From Findings 1-2, we use sep $=$ \n, G_{verbal} and $\Omega =$ *Diverse*-COMET. From Finding 3, we use $n = 5$ for the small models, $n = 3$ for large models, and $n = 1$ for very large models.

Injecting knowledge helps. For each scale of model under investigation, we compare versions with and without external knowledge. Table 6 shows the overall human evaluation numbers on the hidden test set of the TellMeWhy dataset as calculated according to §2.4.

Injecting external knowledge helps the small models the most. While overall performance improves, the biggest improvement is on implicit questions, where the answer is not available in the narrative. This shows that such external knowledge can significantly fill gaps in small models.

Additonally, we find that external knowledge improves the performance of very large models (GPT-3) more than it does for large models (T5-11B). This can be due to various reasons. First, GPT-3 is used in a zero-shot setting while the others are finetuned. Second, it is possible that very large models have a greater capacity to use external information.

Size	Setting	Avg Likert		Binary Accuracy	
		Full	Impl	Full	Impl
Small	w/o knowl	0.58	0.02	0.61	0.42
	w/ knowl	0.91	0.56	0.73	0.61
Large	w/o knowl	1.21	0.97	0.84	0.75
	w/ knowl	1.27	1.04	0.85	0.77
V. large	w/o knowl	1.17	1.1	0.83	0.8
	w/ knowl	**1.32**	**1.24**	**0.87**	**0.85**
Humans		1.35	1.28	0.99	0.97

Table 6: KnowWhy approach on TellMeWhy dataset achieves a new SOTA. Judiciously adding knowledge helps across model sizes.

Scale matters. Table 6 indicates that just increasing the scale of the model results in a large performance boost (5x higher than the previous SOTA Lal et al. (2021), which achieves 0.36 on Full and -0.27 on IMPL on the Avg Likert metric §2.4). Judiciously adding knowledge helps across all model sizes. Large models outperform small models, even when small models are augmented with external knowledge. Interestingly, adding external, relevant commonsense knowledge still significantly helps large and very large models correctly answer questions. T5-11B and GPT-3 augmented with knowledge achieve the best performance on this dataset and come very close to human performance on the Avg Likert metric.

4.1 Have models actually reached human performance?

To investigate this, we compare scores for humans and models on the spectrum of the Likert scale.

Likert	T5-Base	T5-11B	GPT-3	Human
-2	0.05	0.01	0.01	0
-1	0.09	0.05	0.02	0.01
0	0.11	0.08	0.08	0
1	0.34	0.36	0.37	0.55
2	0.41	0.5	0.52	0.44

Table 7: Comparison of models by percentage of scores of different Likert values for their answers. Larger models get a higher percentage of strong agreement (Likert=2) scores than humans. Humans maintain near-perfect consistency of correct answers (Likert>0).

Table 7 suggests that (very) large models are unable to maintain peak performance consistently.

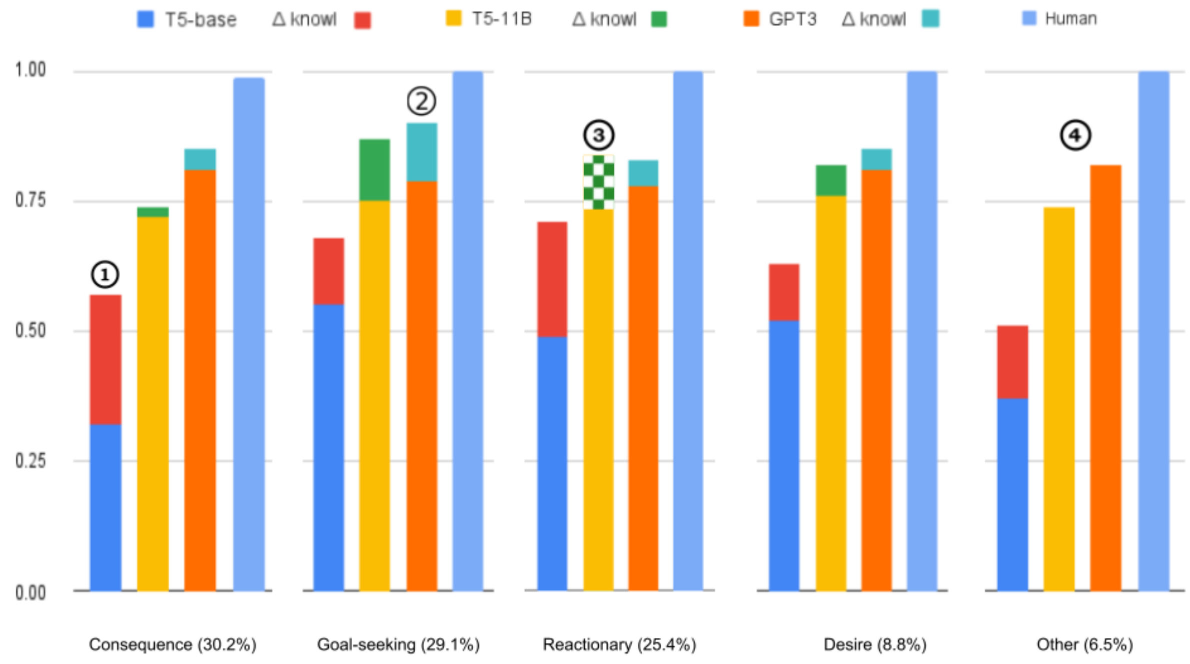

Figure 2: Binary accuracy of models by question type for IMPL. The checkered pattern indicates that there is a drop in performance when external knowledge is added. **1.** T5-BASE benefits most from consequence knowledge. **2.** GPT-3 and T5-11B get largest gains from Goal-seeking knowledge. **3.** The only case where knowledge hurts is for T5-11B with Reactionary type. **4.** Other knowledge categories don't help T5-11B and GPT-3.

For a Likert score of +2, they outperform humans: 0.44 vs. 0.52 for GPT-3. Figure 3 shows an example where the model answer is judged by humans to be better than a human answer. However, unlike humans, this performance is inconsistent. Models generate more answers given scores 0, -1, and -2. Figure 3 also shows an example where the model generates a terrible answer that is rated -2, a score that no human answer is ever given. This is in line with Bender and Koller (2020): large models are on topic, but can be unclear or fail to make sense.

To make the comparison with human performance clearer, similar to (Lal et al., 2021), we collapse the 5-point average Likert into a binary measure of accuracy (only scores of 1 and 2 are counted as correct). On this binary accuracy metric, as shown in Table 6, humans are almost perfect, with an accuracy of 99%. However, the best model (GPT-3 with knowledge) only achieves 87%, indicating that there is still significant room for improvement on this task.

5 Analysis

To better understand the strengths and weaknesses of these models, we defined an ontology for the types of knowledge that are required to answer TellMeWhy questions. We identified five categories of questions, and then labeled the CaTeRs subset of TellMeWhy, for which the gold answers already have human evaluation judgments.

Matt and Sarah were pregnant. They wanted to announce it in a fun way. They wrote it on a cake. The, they invited their friends over. When their friends saw the cake, they were excited.

Question: Why were Matt and Sarah pregnant?

Human Answer: The two had previously been intimate together.

Model Answer: Matt and Sarah were pregnant because they wanted to have a baby.

Maggie was drinking some green juice. She left the cup out awhile. When she went to get another sip it tasted odd. She realized that it had separated weirdly. She threw the juice out.

Question: Why did she leave the cup?

Human Answer: There was something else Maggie had to attend to for a bit.

Model Answer: Maggie left the cup because it was too heavy.

Figure 3: Examples of cases where the model comes up with really good (upper box) and really bad (lower box) answers. In the upper box, when augmented with knowledge, GPT-3 generates an answer that achieves a +2 Likert score, while the human gold answer itself only got a +1. In the lower box, GPT-3's answer gets a Likert score of -2 while the human gold answer is +1.

5.1 Question distribution in IMPL subset

The categories are:

- **Consequence (30.2%)**: an event happened as a consequence of another event.
- **Goal-seeking (29.1%)**: an agent performed an action as an intermediate step to a goal.
- **Reactionary (25.4%)**: an agent performed an action as a reaction to another event.
- **Desire (8.8%)**: an agent performed an action to accomplish an inherent goal.
- **Other (6.5%)**: types of knowledge that do not fall into the categories above.

Examples of each type can be found in Figure 6.

Since implicit answer questions (IMPL) require knowledge outside the text, we analyze them to study the gaps in the models' understanding and identify possible areas for improvement. Figure 2 presents binary accuracy of all models across question types.

To quantify the differences across models, we compute a failure probability for each category, i.e., the probability of an incorrectly answered question (Avg 5pt Likert score of the model answer to the question < 1) belonging to a given category. We compute this by dividing the number of incorrectly answered questions of that knowledge type by the total number of wrong answers. We measure the differences in these failure probability distributions across models using Jensen-Shannon Divergence (JSD).

5.2 Reasoning that Models Lack

Figure 2 shows that small models are unable to reason adequately about all knowledge types, and adding external knowledge boosts performance across the board, particularly for 'Consequence' questions. As the model size increases, it's understanding of each type also increases. However, without knowledge, there is a huge gap in the performance of even the largest models when compared to humans across all categories, showing that understanding all of the aspects of an event needed to answer why questions is hard.

The JSD of the failure probability distributions for T5-BASE and T5-11B across categories is only 0.13 and T5-11B and GPT-3 is 0.14. This suggests only a small difference in the knowledge types these models fail to capture.

Matt and Sarah were pregnant. They wanted to announce it in a fun way. They wrote it on a cake. The, they invited their friends over. When their friends saw the cake, they were excited.

Question: Why did they write it?

Model Answer (no CSK): They wrote it on a cake.

Human Answer: Matt and Sarah wanted to surprise their friends with something unexpected.

Model Answer (w/ CSK): To let their friends know that they were expecting a baby.

Figure 4: An example where knowledge helps GPT-3. The model answer without CSK achieved a Likert score of 0, the human answer had a Likert score of +1, and the model answer with CSK scored +2.

5.3 Where Knowledge Helps

Figure 2 shows how adding knowledge helps for questions of different categories. External knowledge consistently helps models of all sizes, except that it hurts significantly for 'Reactionary' questions. For 'Consequence' questions, adding CSK pushes T5-BASE (a 110M parameter model) to close to T5-11B (11B parameter model) even though the latter is finetuned without CSK. Adding CSK improves GPT-3 the most on 'Goal-seeking' questions. Figure 4 shows an example where knowledge helps.

6 Related Work

6.1 Knowledge Bases

Knowledge bases (KBs) are a reliable source of world facts and relationships between common concepts. They can be constructed through semi-automated extraction over text (Speer et al., 2017; Tandon et al., 2017) or through crowdsourcing (Sap et al., 2019).

Petroni et al. (2019) show that, instead of these approaches, pretraining language models on text already endows them with certain types of factual knowledge that helps them do well on QA tasks. More recently, a popular approach is to fine-tune a language model on existing KBs, to generalize their knowledge and pay attention to the context, e.g., COMET (Bosselut et al., 2019; Hwang et al., 2021) generates context-relevant commonsense knowledge. It is a fine-tuned language model over ATOMIC and ConceptNet KBs. Similarly, ParaCOMET (Gabriel et al., 2021) is a language model fine-tuned for discourse knowledge by fine-tuning over ROCStories, thus it generates relations consistent with an input narrative.

6.2 Incorporating External Knowledge

Model outputs have been improved through commonsense injection using regularization at training time (Guan et al., 2020) or simply by appending to the input (Lewis et al., 2020; Talmor et al., 2020).

There are two key challenges in using external sources. The first is figuring out what knowledge to use and the second is determining how to effectively integrate it into the end task.

Some recent research injects triples into sentences in order to create domain-specific knowledge (Liu et al., 2020; Wang et al., 2020). Huang et al. (2019) incorporate commonsense knowledge directly into training data. Feng et al. (2020) leverage relations from ConceptNet using structured relational attention to perform multi-hop QA. However, there is still uncertainty on the proper way to use external knowledge to solve commonsense reasoning problems (Zhang et al., 2020).

ERNIE (Zhang et al., 2019) is an enhanced language representation model trained using large-scale corpora and knowledge graphs that shows significant improvements on various knowledge-driven tasks. Xiong et al. (2020) propose a weakly supervised pretraining objective, which explicitly forces the model to incorporate knowledge about real-world entities to perform entity-related QA tasks. KGLM (Logan et al., 2019) is a neural language model with mechanisms for selecting and copying facts from a knowledge graph that are relevant to the context.

KagNet (Lin et al., 2019) grounds a QA pair in CommonsenseQA (Talmor et al., 2019) from the semantic space to the knowledge-based symbolic space as a schema graph, uses a KG-aware module to focus on it, and scores answers with graph representations. Lv et al. (2020) propose a graph-based contextual representation learning and inference module to better use graph information for commonsense QA. Shwartz et al. (2020) generate and integrate background knowledge from pretrained LMs to develop an unsupervised framework for multiple-choice commonsense tasks. Generated knowledge prompting elicits and integrates knowledge from language models using task-specific, human-written, few-shot demonstrations so as to improve performance on commonsense reasoning tasks (Liu et al., 2021).

7 Conclusion

Answering why questions requires several forms of commonsense knowledge. This paper investigates different aspects of incorporating external knowledge to improve this process. We discover several empirical insights on how to incorporate external knowledge. By incorporating these insights, our approach, KNOWWHY, successfully uses external knowledge to help models of all sizes (small, large, and very large) answer why questions better; but they still fall short of human performance. Questions that involve implicit inferences are harder for all the models, and require modeling innovations. Our investigation opens up interesting questions, such as learning when and how to add external knowledge, in order to further close the gap with humans.

8 Limitations

Reproducing our experiments for T5-11B requires extensive compute resources, including TPUs, while running zero-shot experiments with GPT-3 requires access to the paid OpenAI API.

Few-shot prompting and in-context learning are also popular ways of using models like GPT-3 for various tasks. We leave the exploration of such methods for TELLMEWHY for future work.

Our investigation into adding external commonsense to help NLP models perform better is limited to just one dataset that focuses on why question answering. Indeed, such knowledge is also required to enhance other reasoning capabilities of a model. It would be interesting to see how our findings would transfer to other tasks.

COMET as a source of knowledge is limited in the quality and relevance of information it can provide. Studying other sources of commonsense knowledge would be another productive area for future work.

Finally, human evaluation of free-form model answers is expensive and time-consuming. Even though our current answer cache is fairly sizeable, there is non-trivial time and expense involved in following the evaluation suggested by Lal et al. (2021).

Acknowledgements

This material is based on research that is supported in part by the Air Force Research Laboratory (AFRL), DARPA, for the KAIROS program under agreement number FA8750-19-2-1003 and in part by the National Science Foundation under the award IIS #2007290. The authors would like to thank the anonymous reviewers and the area chair for their feedback on this work. We would also like to thank Jierui Li for her suggestions on the camera ready version.

References

Forough Arabshahi, Jennifer Lee, Antoine Bosselut, Yejin Choi, and Tom Mitchell. 2021. Conversational multi-hop reasoning with neural commonsense knowledge and symbolic logic rules. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7404–7418, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Bin Bi, Chen Wu, Ming Yan, Wei Wang, Jiangnan Xia, and Chenliang Li. 2019. Incorporating external knowledge into machine reading for generative question answering. 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 2521–2530, Hong Kong, China. Association for Computational Linguistics.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. 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.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309, Online. Association for Computational Linguistics.

Saadia Gabriel, Chandra Bhagavatula, Vered Shwartz, Ronan Le Bras, Maxwell Forbes, and Yejin Choi. 2021. Paragraph-level commonsense transformers with recurrent memory. In *AAAI*.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108.

H. He, Hua Lu, Siqi Bao, Fan Wang, Hua Wu, Zhengyu Niu, and Haifeng Wang. 2021. Learning to select external knowledge with multi-scale negative sampling. *ArXiv*, abs/2102.02096.

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

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. In *AAAI*.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Yash Kumar Lal, Nathanael Chambers, Raymond Mooney, and Niranjan Balasubramanian. 2021. TellMeWhy: A dataset for answering why-questions in narratives. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 596–610, Online. Association for Computational Linguistics.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Torrin M. Liddell and John K. Kruschke. 2018. Analyzing ordinal data with metric models: What could possibly go wrong? *Journal of Experimental Social Psychology*, 79:328–348.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2829–2839, Hong Kong, China. Association for Computational Linguistics.

Jiachen Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2021. Generated knowledge prompting for commonsense reasoning. *ArXiv*, abs/2110.08387.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. *ArXiv*, abs/1909.07606.

Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971, Florence, Italy. Association for Computational Linguistics.

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. 2020. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, USA*, pages 8449–8456. AAAI Press.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

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.

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.

Minlie Huang Sahand Sabour, Chujie Zheng. 2021. Cem: Commonsense-aware empathetic response generation. *arXiv preprint arXiv:2109.05739*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3027–3035.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4615–4629, Online. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *ArXiv*, abs/1811.00937.

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. *Advances in Neural Information Processing Systems*, 33:20227–20237.

Niket Tandon, Gerard de Melo, and Gerhard Weikum. 2017. Webchild 2.0 : Fine-grained commonsense knowledge distillation. In *ACL*.

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, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. 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 H. Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. *CoRR*, abs/2201.08239.

Peifeng Wang, Nanyun Peng, Filip Ilievski, Pedro Szekely, and Xiang Ren. 2020. Connecting the dots: A knowledgeable path generator for commonsense question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4129–4140, Online. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

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

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *ArXiv*, abs/1912.09637.

Yichong Xu, Chenguang Zhu, Shuohang Wang, Siqi Sun, Hao Cheng, Xiaodong Liu, Jianfeng Gao, Pengcheng He, Michael Zeng, and Xuedong Huang. 2021a. Human parity on commonsenseqa: Augmenting self-attention with external attention. *CoRR*, abs/2112.03254.

Yichong Xu, Chenguang Zhu, Ruochen Xu, Yang Liu, Michael Zeng, and Xuedong Huang. 2021b. Fusing context into knowledge graph for commonsense question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1201–1207, Online. Association for Computational Linguistics.

Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2031–2043, Online. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Model	Input Format	Output Format
sep = ' '	question: Q context: C	A
sep = '\n'	question: Q \n context: C	A
G_{tup}	question: Q context: C <info> relation: R1 phrase: P1 </info> <info> relation: R2 phrase: P2 </info> <info> relation: R3 phrase: P3 </info>	A
G_{tupsep}	question: Q \n context: C \n <info> relation: R1 \n phrase: P1 </info> \n <info> relation: R2 \n phrase: P2 </info> \n <info> relation: R3 \n phrase: P3 </info> \n	A
$G_{verbalized}$	question: Q \n context: C \n V1 \n V2 \n V3 \n	A

Figure 5: Example inputs and target outputs for different models of the T5 family. Q represents the question, C denotes the context and A denotes the answer. P_i denotes a relation from COMET, R_i denotes its type and VP_i denotes its verbalized form according to Table 10. Here $i = 1, 2, 3$.

A Caching for Human Evaluation

In order to improve time and cost efficiency, we implement a caching mechanism to re-use previous annotator judgments for the same answer for a question in a particular story. For this purpose, we save all the human judgments for a (question, answer, story) triple. For all model predictions, we first check if a (question, answer, story) triple[4] is already present in the cache. If it is, we use the old judgments for it. If not, we gather validity annotations for it using human evaluation and add them to the cache for future use. We have built up a cache of ∼7000 model-generated answers and will release it so that it becomes easier to perform human evaluation on this dataset in the future.

B Building the Ranking Model

Configuration	P@3	P@5	P@10
Q: narrative + ques D: relation phrase	0.32	0.41	0.57
Q: ques D: relation phrase	0.25	0.34	0.50

Table 8: Pretrained msmarco-distilbert-dot-v5 model ranking precision scores for different query (Q) and document (D) configurations

Here, we provide details into our experiments with finetuning the pretrained msmarco-distilbert-dot-v5 model to rank the COMET relations associated with a context and question.

We experimented with different formulations of queries and documents. Table 8 shows the ranking precision scores of the pretrained model for different configurations. We select the first one for finetuning as it has a higher P@k suggesting that adding the context after a question is most helpful.

We used Adam optimizer with a learning rate of 1e-05 and weight decay of 1e-04. The batch size was 1 and we used Precision@5 scores to select the best finetuned model.

C Hyperparameters

C.1 T5-Base

For T5-Base, we train the model with batch size 16, learning rate 5e-5 and maximum answer length 30. We vary the source length from 75 to 450 according to the amount of external knowledge being injected into the input context. The model is trained until the dev loss fails to improve for 3 iterations. Training usually takes 7-8 hr on 1 Titan Xp GPU.

C.2 T5-11B

For training the T5-Base model, we followed a default set of hyperparameters that are recommended in (Raffel et al., 2020).[5]

T5-Base model has 110M parameters with 24-layers, 1024-hidden-state, 4096 feed-forward hidden-state, and 16 attention heads. T5-11B model has 11B parameters with 24-layers, 1024-hidden-state, 65,536 feed-forward hidden-state, 128 attention heads. We use TPU (v3-8) on google cloud platform. It takes 6 hours in average to train the model.

C.3 GPT-3

We used a temperature of 0.0 for all the experiments to select the most likely token at each step, as this setting allow for reproducibility[6].

```
import os
import openai

openai.api_key = os.getenv("OPENAI_API_KEY")
```

[4] All text is lowercased and answer is also stripped of punctuation.

[5] https://github.com/google-research/text-to-text-transfer-transformer

[6] We note that some researchers have shown that even this setting might not make it completely reproducible: https://twitter.com/ofirpress/status/1542610741668093952?s=46&t=f9v5k9RzVKnTK1e0UyauOA

```
response = openai.Completion.create(
            engine="text-davinci-002",
            prompt=prompt,
          temperature=0.0, # for reproducibility.
            max_tokens=40,
            top_p=1,
            frequency_penalty=0.1,
            presence_penalty=0
)
```

The frequency penalty penalizes new tokens based on existing frequency in text so far, while the presence penalty sets the model's likelihood to talk about novel topics.

Split	# stories	# questions
Train	7558	23964
Dev	944	2992
Test	944	3099
Hidden Test	190	464
Total	9,636	30,519

Table 9: TELLMEWHY Dataset Statistics

Relation Type	Verbalization
Causes	causes
CausesDesires	makes someone want
DesireOf	is a desire of
Desires	desires
HasFirstSubevent	begins with
HasLastSubevent	ends with
HasPrerequisite	to do this, one requires
HasSubevent	includes
HinderedBy	can be hindered by
MotivatedByGoal	is a step towards accomplishing
oEffect	as a result, they will
oReact	as a result, they feel
oWant	as a result, they want
xEffect	as a result, she will
xIntent	because she wanted
xNeed	but before, she needed
xReact	as a result, she feels
xReason	because
xWant	as a result, she wants

Table 10: Fluent natural language templates used to verbalize each relation according to its type. To prepare the external knowledge for G_{verbal} , the sentence best-aligned to the question precedes the verbalization and the relation succeeds it.

D Automatic Metrics

Table 13 shows the values for various automatic metrics for different models we built. We adapt the evaluation script released by Lal et al. (2021) to obtain these numbers.

Story: Sandra got a job at the zoo. She loved coming to work and seeing all of the animals. Sandra went to look at the polar bears during her lunch break. She watched them eat fish and jump in and out of the water. She took pictures and shared them with her friends. **Question:** Why did Sandra go to look at the polar bears during her lunch break? **Ans:** she wanted to take some pictures of them.
Story: Cam ordered a pizza and took it home. He opened the box to take out a slice. Cam discovered that the store did not cut the pizza for him. He looked for his pizza cutter but did not find it. He had to use his chef knife to cut a slice. **Question:** Why did Cam order a pizza? **Ans:** Cam was hungry.

Table 11: Examples from the TELLMEWHY dataset. The first is answerable directly from text in the story, but the second requires external knowledge. We only show one out of three available answers here. TELLMEWHY was released by its authors at https://stonybrooknlp.github.io/tellmewhy/

Knowledge Type	Narrative	Question
Consequence	Marissa had just finished a scary movie with her boyfriend. She was terrified after watching it. She tried to go to bed but was too scared to sleep. Instead, she asked her boyfriend to stay up with her. Her boyfriend stayed up with her until she fell asleep.	Why did She try to go to bed?
Reactionary	I was watching the game furious. This Referee had no idea what he was doing. He kept making bad calls. I yelled at him. Then he had the nerve to kick me out of the game.	Why did I yell at him?
Goal-seeking	Grandma woke Lucy up at 6 on Sunday morning. Lucy was groggy and confused. Lucy was to get bathed and put on the dress on her bed. Lucy didn't understand what was going on. She realized they were going to church when she saw grandpa's suit.	Why did Grandma wake Lucy?
Desire	Lucy awoke planning to go outside and play. But when she sat up she could hear the rain on her window. She looked outside and saw the storm clouds were rolling in. Upset she couldn't go outside she went to her bookshelf instead. Lucy lay on her bed and read books the whole day.	Why did Lucy awake planning to go outside and play?
Other	I was watching the game furious. This Referee had no idea what he was doing. He kept making bad calls. I yelled at him. Then he had the nerve to kick me out of the game.	Why did He keep making bad calls?

Figure 6: Examples of questions associated with each knowledge type in the ontology.

		Narrative (N): Rudy was convinced that bottled waters all tasted the same. He went to the store and bought several popular brands. He went back home and set them all on a table. He spent several hours tasting them one by one. He came to the conclusion that they actually did taste different. Question (Q): Why did He go back home?
No knowledge	Prompt Answer	S1 S2 S3 Q The correct reason is: Rudy went back home to compare the different brands of water side by side.
Prefix	Prompt Answer	Note: CSK S1 S2 S3 Q The correct reason is: to taste them one by one
Postfix	Prompt Answer	S1 S2 S3 Note: CSK Q The correct reason is: He went back home to taste the waters.
Infix	Prompt Answer	S1 S2 CSK S3 Q The correct reason is: He went back home to test the waters.

Table 12: This table the different types of prompt formats we tried for GPT-3, using an example. Each narrative can be represented as a sequence of sentences S1, S2, S3. The external knowledge is denoted as CSK.

Size	**Setting**	**BertScore**		**ROUGE-L F1**		**BLEU**		**BLEURT**	
		Full	Impl	Full	Impl	Full	Impl	Full	Impl
Small	w/o knowl	0.45	0.38	0.24	0.17	20.49	14.61	-0.36	-0.59
	w/ knowl	0.45	0.36	0.23	0.18	19.85	13.7	-0.38	-0.62
Large	w/o knowl	0.42	0.35	0.22	0.17	17.3	12.6	-0.22	-0.42
	w/ knowl	0.43	0.35	0.23	0.18	17.1	12.4	-0.22	-0.44
Very large	w/o knowl	0.28	0.26	0.15	0.14	8.63	7.66	-0.78	-0.83
	w/ knowl	0.39	0.36	0.24	0.21	17.32	15.24	-0.50	-0.50

Table 13: Scores of various models using automatic metrics for the free-form, open-ended TellMeWhy answer generation task. We use the same logic followed by Lal et al. (2021). The trends for none of these metrics match any trends observed in Table 6.

		Full	Impl
Small (T5-BASE)	w/o sep.	0.36	-0.27
	w/ sep.	0.58	0.02
Large (T5-11B)	w/o sep.	0.99	0.6
	w/ sep.	1.21	0.97

Table 14: Performance of both small and large models on Avg Likert score improves significantly when adding a clear separator token ($sep = \backslash n$) to the original T5 format specified in Raffel et al. (2020) Appendix D.3.