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

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

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 COMET2 (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

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:

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

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 \(\Omega\) 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. [HAS_SUBEVENT] [GEN]

Kelsi was excited to try out bright red hair. [DESIRE_TO] [GEN]

...
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 ($\Omega$ in Equation 1). We thus hard-code $n = 3$ and use ($G = \text{verbal}$) to explore $\Omega$. The relations are verbalized according to the templates presented in Table 10. More details about $G_{\text{verbal}}$ 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.

- $\Omega = \text{COMET (original)}$ 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.

<table>
<thead>
<tr>
<th>Ranking model</th>
<th>P@3</th>
<th>P@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMET score</td>
<td>0.14</td>
<td>0.23</td>
</tr>
<tr>
<td>Pretrained MSMARCO</td>
<td>0.32</td>
<td>0.41</td>
</tr>
<tr>
<td>Finetuned MSMARCO</td>
<td>0.45</td>
<td>0.54</td>
</tr>
</tbody>
</table>

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

- $\Omega = \text{Reranked-COMET (fine-tuned)}$ 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.

- $\Omega = \text{Diverse-COMET}$ 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...
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.

Table 4: How to inject knowledge? The best way to inject COMET inferences is to verbalize relational information as fluent 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 \((\text{sep} = \backslash n)\) 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.

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.

3.3 How much Knowledge to Inject?
We also investigate how the amount of knowledge added \((n \text{ in Eq. 1)}\) affects the performance of the model. We set \(\Omega = \text{Diverse-COMET}\) and use \(G_{\text{verbal}}\).

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 of 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 = \textbackslash n, \(G_{\text{verbal}}\) and \(\Omega = \text{Diverse-COMET}\). 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.

Additionally, 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 other models are finetuned. Second, it is possible that very large models have a greater capacity to use external information.

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.

<table>
<thead>
<tr>
<th>Likert</th>
<th>T5-BASE</th>
<th>T5-11B</th>
<th>GPT-3</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>0.05</td>
<td>0.01</td>
<td>0.01</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>0.09</td>
<td>0.05</td>
<td>0.02</td>
<td>0.01</td>
</tr>
<tr>
<td>0</td>
<td>0.11</td>
<td>0.08</td>
<td>0.08</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.34</td>
<td>0.36</td>
<td>0.37</td>
<td>0.55</td>
</tr>
<tr>
<td>2</td>
<td>0.41</td>
<td>0.5</td>
<td>0.52</td>
<td>0.44</td>
</tr>
</tbody>
</table>

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.
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, KOnewWhY, 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.
References


Daniel Kathshabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing for-


Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph


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 \( V P_i \) 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 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 \( \sim 7000 \) 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

<table>
<thead>
<tr>
<th>Configuration</th>
<th>P@3</th>
<th>P@5</th>
<th>P@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q: narrative + ques</td>
<td>0.32</td>
<td>0.41</td>
<td>0.57</td>
</tr>
<tr>
<td>D: relation phrase</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q: ques</td>
<td>0.25</td>
<td>0.34</td>
<td>0.50</td>
</tr>
<tr>
<td>D: relation phrase</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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\(^6\).

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

\(^5\)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=f9v5k9RzKntK1e0Uyau0A
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.

<table>
<thead>
<tr>
<th>Split</th>
<th># stories</th>
<th># questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>7558</td>
<td>23964</td>
</tr>
<tr>
<td>Dev</td>
<td>944</td>
<td>2992</td>
</tr>
<tr>
<td>Test</td>
<td>944</td>
<td>3099</td>
</tr>
<tr>
<td>Hidden Test</td>
<td>190</td>
<td>464</td>
</tr>
<tr>
<td>Total</td>
<td>9,636</td>
<td>30,519</td>
</tr>
</tbody>
</table>

Table 9: TELL.ME.WHY Dataset Statistics

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

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 TELL.ME.WHY 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. TELL.ME.WHY was released by its authors at https://stonybrooknlp.github.io/tellmewhy/
Figure 6: Examples of questions associated with each knowledge type in the ontology.

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.

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.
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.

<table>
<thead>
<tr>
<th></th>
<th>Full</th>
<th>Impl</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Small (T5-Base)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o sep.</td>
<td>0.36</td>
<td>-0.27</td>
</tr>
<tr>
<td>w/ sep.</td>
<td>0.58</td>
<td>0.02</td>
</tr>
<tr>
<td><strong>Large (T5-11B)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o sep.</td>
<td>0.99</td>
<td>0.6</td>
</tr>
<tr>
<td>w/ sep.</td>
<td>1.21</td>
<td>0.97</td>
</tr>
</tbody>
</table>