TellMeWhy Project Report

1 Overview

Understanding and generating answers to "why" questions in narratives is a complex task in natural language processing (NLP) with significant real-world applications, such as conversational agents, educational tools, and intelligent story comprehension systems. The challenge lies in enabling models to infer implicit causal relationships and apply commonsense reasoning to explain actions in a story. These tasks demand bridging the gap between explicit textual cues and unstated reasoning. Our project focuses on enhancing model capabilities to generate accurate, contextually grounded answers to "why" questions using the TellMeWhy dataset. Specifically, we aim to develop a pipeline that incorporates external commonsense knowledge, reducing overgeneralization and improving response specificity.

While pre-trained transformer-based models like T5 perform well in tasks involving explicit causal relationships, they struggle with implicit reasoning. Existing methods often fail to integrate external knowledge effectively, leading to overgeneralized and contextually vague responses. For instance, models trained on the TellMeWhy dataset frequently default to generic outputs due to the absence of implicit reasoning. Although approaches such as sentence highlighting have shown improvements, they do not fully address the challenges of bridging gaps in reasoning. Therefore, there is a need for strategies that leverage external commonsense knowledge to generate contextually rich, accurate responses.

To address these limitations, we propose a solution that augments the narrative context with external commonsense knowledge. The core components of our approach include the integration of commonsense reasoning generated by pre-trained models, the combination of this knowledge with highlighted sentences from the narrative to enhance contextual focus, and the fine-tuning of a pre-trained T5 model to align outputs with the enriched inputs. This approach is designed to mitigate overgeneralization and improve the model's ability to infer implicit causal relationships, thus producing responses that are more specific and relevant to the narrative context.

The implementation of this approach involves augmenting the input narrative with additional commonsense reasoning generated from pre-trained models such as UnifiedQA. These generated insights complement the narrative by filling gaps in reasoning that are not explicitly stated in the text. Additionally, the enriched input representation includes key sentences from the narrative, annotated in the dataset as being most helpful for answering the questions. This augmented context is then used to fine-tune a T5 model, enabling it to generate outputs that are more grounded in both the narrative and the inferred commonsense reasoning.

To evaluate the proposed approach, we use four key metrics: BLEU, ROUGE, BLEURT, and BERTScore. BLEU measures lexical overlap between predicted and reference answers, while ROUGE evaluates recall and precision of n-grams and longest common subsequences. BLEURT assesses semantic similarity and contextual appropriateness, and BERTScore captures semantic similarity using contextual embeddings. Experiments are conducted on the TellMeWhy dataset, which includes narratives, questions, and annotated answers. The evaluation focuses on both explicit and implicit reasoning questions, with a detailed analysis of the model's performance and its ability to address overgeneralization.

Preliminary results indicate that incorporating commonsense knowledge into the narrative context significantly improves performance on metrics such as BLEU and ROUGE. The pipeline also demonstrates promise in reducing overgeneralization, enabling the generation of contextually specific responses. However, challenges remain in handling highly implicit reasoning, which we aim to address through further experimentation. Overall, the results highlight the potential of combining external knowledge with enhanced input representations to improve the performance of generative models in answering "why" questions.

2 Ideas

2.1 Idea 1

The first idea involved **developing a baseline model** that leverages the existing structure of the TellMeWhy dataset. We fine-tuned the pre-trained T5 transformer model on the dataset, where the input was formatted as question: <question> context: <narrative> and the target was the corresponding answer. The preprocessing included tokenization and removal of non-essential fields to streamline the dataset for training. The model was trained to generate answers directly from the provided narrative and question, focusing on optimizing explicit reasoning.

This approach was essential as it established a benchmark for performance and highlighted the limitations of directly fine-tuning T5 without additional enhancements. While the results showed reasonable performance on explicit questions, the model frequently overgeneralized when addressing implicit reasoning tasks, producing vague answers that lacked contextual specificity.

2.2 Idea 2

To address the overgeneralization observed in Idea 1, the second idea incorporated **highlighted key sentences** from the dataset annotations. These sentences, marked as helpful by annotators, were extracted and appended to the narrative context during preprocessing. This approach emphasized the most relevant parts of the narrative, directing the model's attention to crucial context while answering questions.

For implementation, the preprocessing function combined the narrative and highlighted sentences, formatting the input as question: <question> context: <narrative> <highlighted sentences>. The model was then fine-tuned on this enriched input format. This modification improved the model's ability to produce more specific answers for both explicit and implicit questions. However, the model still struggled with deeply implicit reasoning, as the highlighted sentences alone could not bridge the reasoning gap fully.

2.3 Idea 3

Building upon the progress from Ideas 1 and 2, the third idea introduced **external commonsense knowledge** into the model pipeline. Pre-trained commonsense reasoning models such as UnifiedQA were used to generate additional insights about the narrative. These generated commonsense insights were appended to the narrative context to create an augmented input representation. The enriched format was structured as question: <question> context: <narrative> Commonsense: <commonsense knowledge>.

This approach requires integrating an external model to generate commonsense knowledge dynamically during preprocessing. The augmented inputs, combining narrative, highlighted sentences, and commonsense reasoning, were used to fine-tune the T5 model. The goal was to enable the model to infer implicit causal relationships more effectively and generate contextually grounded, specific answers.

3 Experimental Setup

3.1 Models

The model used for this project is a transformer-based architecture, specifically a pre-trained T5-small model from the Hugging Face Transformers library. T5 is a sequence-to-sequence model designed to handle a wide range of NLP tasks by unifying them under a text-to-text framework. It consists of an encoder-decoder architecture, with 6 layers each in the encoder and decoder, making it efficient for fine-tuning on medium-sized datasets.

The training configuration is summarized in the following table:

Model	**Type**	**Pre-trained**	**Fine-Tuning**	**Epochs**	**Batch Size**	**Optimizer**	**Learning Rate**	**AMP Enabled**	**GPU Used**
T5-small	Transformer	Yes	Yes, on TellMeWhy	3	16	AdamW	5e-5	Yes	NVIDIA T4 GPU

3.2 Dataset

The dataset used for this project is the **TellMeWhy** dataset, which comprises narratives, associated "why" questions, and their corresponding answers.Each instance in the dataset includes the following fields:

Split	**Number of Examples**
Training	71,892
Validation	8,976
Test	10,689

Narrative: The full story from which questions are generated.
Question: A "why" question derived from the narrative.
Answer: The corresponding explanation for the question, either implicit or explicit.
Helpful Sentences: Annotated sentences deemed most helpful for answering the question.
is_ques_answerable: A binary flag indicating whether the question has an answer in the story.

3.3 Evaluation Metrics

To evaluate the fine-tuned model, four automatic metrics were employed. These metrics ensure a comprehensive assessment of the model’s performance across lexical, semantic, and contextual dimensions. These metrics were applied to both **explicit** and **implicit** questions to analyze the model's ability to handle various reasoning scenarios and reduce overgeneralization.

Metric	Focus
BLEU	Lexical similarity
ROUGE	Key content similarity
BLEURT	Context and meaning
BERTScore	Semantic relevance

4 Results

The results highlight the effectiveness of integrating external knowledge and contextual enhancements in generating meaningful and precise responses. BLEU and ROUGE-L scores reflect how closely the generated answers align with reference answers at the lexical level. BLEURT and BERTScore further capture semantic similarity and contextual appropriateness, providing a deeper evaluation of the model's reasoning abilities.

Overall Results

Metric	**Idea 1**	**Idea 2**	**Idea 3**
BLEU	10.81	13.55	11.15
ROUGE-L(F1)	23.47%	26.26%	24.23%
BERTScore (F1)	40.20%	90.49%	90.14%
BLEURT (Avg)	-0.7950	-0.7370	-0.8437

Observations:

- **Idea 1** demonstrated a baseline capability, with moderate lexical overlap and semantic alignment, but struggled with nuanced reasoning and contextually rich outputs, as reflected in the low BLEURT and BERTScore.
- **Idea 2** outperformed other approaches across all metrics, indicating that sentence highlighting combined with external commonsense knowledge significantly improves

both lexical alignment and semantic depth. This suggests that enhancing the context with relevant information enables the model to generate more precise and contextually appropriate responses.

- **Idea 3** maintained strong semantic performance, as indicated by the BERTScore, but showed a slight decline in BLEU and ROUGE-L compared to Idea 2. This suggests that while the approach captured semantic meaning effectively, it faced challenges in achieving high lexical alignment.

Explicit vs. Implicit Results

Metric	Idea 2		Idea 3	
	Explicit	Implicit	Explicit	Implicit
BLEU	15.95	6.95	11.14	11.12
ROUGE-L(F1)	27.26%	16.72%	24.20%	24.29%
BERTScore (F1)	46.02%	38.23%	90.14%	90.13%
BLEURT (Avg)	-0.6446	-0.9513	-0.8437	-0.8388

- **Idea 2** shows a strong performance gap between explicit and implicit reasoning, indicating that while the contextual highlighting and commonsense augmentation improved handling of explicit causal relationships, implicit reasoning remains a challenge.
- **Idea 3** demonstrates balanced performance between explicit and implicit questions across all metrics, indicating its robustness in addressing both types of questions. However, it slightly underperforms compared to Idea 2 in handling explicit questions.

5 Analysis and Discussion

5.1 Failures

After analyzing outputs from Ideas 1, 2, and 3, the models exhibited three key types of failures:

- **Repetition of Context:** The models often repeated parts of the input narrative as predictions instead of providing meaningful answers.

- **Lack of Implicit Reasoning:** The models struggled with questions requiring implicit reasoning. Despite commonsense augmentation in Idea 3, implicit connections were often overlooked.
- **Underutilization of Commonsense Knowledge:** Idea 3 failed to effectively leverage external commonsense inputs, defaulting to context-based answers.

Idea	Idea 1	Idea 2	Idea 3
Question	Why did Howard buy the Zelda game?	"Why did Howard forget to eat?"	"Why did Howard love his job?”
Gold Answer	"Howard loved the Zelda franchise."	"Howard was so caught up in his new video game."	"The projects Howard worked on were fascinating.”
Predicted Answer	"Howard bought the Zelda game"	"He forgot to eat"	"He had a job"

5.2 Hypotheses

1. **Proximity to Relevant Context Affects Accuracy:** Placing relevant information closer to the question improves predictions.
   - **Example:** Rearranging the context to move "Howard loved the Zelda franchise" closer to the question improved the prediction from "Howard bought the Zelda game" to the correct answer.
2. **Explicit Causal Links Enhance Performance:** The models perform better when causal relationships are explicitly stated in the narrative.
   - **Example:** Adding "Howard forgot to eat because he was caught up in playing Zelda" improved predictions for implicit reasoning questions.
3. **Simplified Commonsense Inputs Improve Results:** Simplifying commonsense inputs to align directly with the question improved Idea 3's predictions.

- **Example:** Changing commonsense from "generate commonsense for" to "Howard loved his job because the projects he worked on were fascinating" yielded accurate responses.

5.3 Conclusion

The models succeed in straightforward tasks but struggle with implicit reasoning and fail to fully utilize external knowledge. Adjusting context proximity, adding explicit causal links, and refining commonsense inputs improved predictions, highlighting areas for enhancing reasoning and external knowledge integration. Future work should focus on training models to handle implicit reasoning and better utilize commonsense knowledge.

6 Code

Google Drive Link: CSE354_Project

The complete project code and outputs are structured around the three main ideas.

Files and Structure:

Notebook: cse_354_project_final_version.ipynb

Environment Setup → Load Dataset → Generate Commonsense Knowledge → Tokenize the Data → Fine-Tune Model → Evaluate and Analyze (Explicit vs. Implicit)

Trained Models and Data: The Drive folder contains preprocessed data, tokenized datasets, trained T5 models for all ideas, and evaluation outputs.

Software Requirements: Python 3.10+, PyTorch 2.0+, Transformers 4.31+, Datasets 2.4+.

7 Learning Outcomes

- Learned to fine-tune transformer models like T5 for NLP tasks.
- Gained skills in evaluating models using BLEU, ROUGE-L, BLEURT, and BERTScore.
- Explored integrating commonsense knowledge to improve model reasoning.
- Improved error analysis and debugging techniques.

8 Contributions

All aspects of the project, including design, implementation, evaluation, and reporting, were completed independently.