CIS^2: A Simplified Commonsense Inference Evaluation for Story Prose

Anonymous CSRR Workshop submission

Abstract

Contextual Commonsense Inference (CCI) is the problem of inferring causal relations between the events of a text, such as a story. Like other commonsense reasoning tasks, CCI is a problem of language understanding, rather than language generation. We show that prior work, in using language generation to perform CCI, trains models that struggle on the CCI task in isolation. This *conflation* of tasks is further exacerbated by evaluating with word-matching based metrics such as BLEU. In order to isolate CCI from language generation, we reframe CCI as a classification problem. Our system, which we call CIS^2, forces the model to focus on CCI directly by providing it the original text of the story to use for understanding while having it generate only the bare minimum: indices to sentences. We look at the GLUCOSE (Mostafazadeh et al., 2020) dataset and compare against their task for predicting CCI between story sentences. We find that models trained on CIS^2 index labels achieve a 4% higher CCI accuracy than those trained for generation, such as in the original GLUCOSE task.

1 Introduction

Transformer-based models (Vaswani et al., 2017) have shown mixed success with story generation (See et al., 2019). Language models (LMs) lose coherence as the output length increases and are prone to meandering, losing the plot over time. This can be largely attributed to the LM generating each token by sampling from a probability distribution, failing to distinguish between statistical correlation (how frequently event A and event B are seen together) and causal reasoning (event A causes event B to occur).

Since causal events across sentences in stories help people understand and retain story information (Trabasso et al., 1984), we posit that the inability of language models to perform commonsense inference leads them to output less coherent long-form text. However, commonsense inference is still an open problem in NLP, especially when the commonsense information is unstructured and provided in the form of natural language. We refer to this task of grounding commonsense inference relations within prose as *contextual commonsense inference (CCI)*, a sub-task within commonsense reasoning. Due to storytelling being deeply intertwined with causal understanding, improving CCI will yield more accurate story generation evaluation metrics and therefore better story generation.

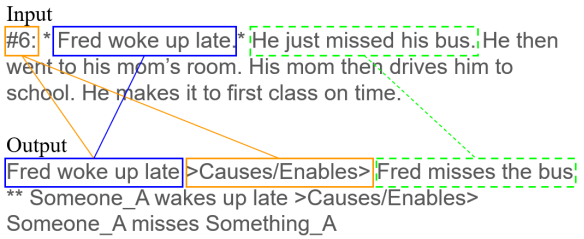

Figure 1: Motivation for CIS^2, illustrating how the original GLUCOSE task conflates commonsense inference and text generation. Input and output are exactly as seen by finetuned T5. Blue: selected sentence X is always paraphrased. Orange: dimension specifies the position of X, and the relation. Green: commonsense inference is needed here to select the other sentence Y.

Our contributions in this paper are twofold. First, we critique existing methods addressing the *contextual commonsense inference* (CCI) task by using the GLUCOSE (Mostafazadeh et al., 2020) dataset and comparing against their associated CCI task formulation. We design several diagnostic tasks which selectively omit sentences of the input and investigate which sentences contribute most to paraphrasing/generation. We replicate their results and finetune T5 models (Raffel et al., 2020) on each of our diagnostic tasks to show significant conflation in the baseline GLUCOSE T5 model.

Second, we propose CIS^2 (Contextual Commonsense Inference in Sentence Selection), a simplified

task for more fairly evaluating commonsense inference in storytelling, which abstracts away the natural language generation component entirely. We develop a heuristic to convert story sentences into CIS^2 tags and show that a language model, when trained on this data, outperforms the original GLUCOSE task formulation on forming correct causal relations between sentences in stories. Our findings reinforce that while the GLUCOSE dataset encodes useful commonsense information, we emphasize that future work should carefully disentangle language generation while performing language understanding tasks.

2 Related Work

Commonsense inference is the ability to use prior knowledge based on real world experiences to infer what has happened or will happen in a text. While lived experiences vary from person to person, there are still significant commonalities throughout our basic interactions with the world around us since we all live in the same physically and temporally-constrained world. Hwang et al. (2021) formalize the *commonsense inference task* (CI) for AI systems as a knowledge three-tuple, to predict the *object* of a relation given the *subject* and *relation*.

2.1 Commonsense Knowledge Graphs

This formulation of commonsense inference easily lends itself to a graph structure, where the subjects and objects (entities) are nodes and the relations are the edges connecting the entities. This line of work on commonsense knowledge graphs (CKGs) explicitly encode the structure of inference relationships between entities. ATOMIC (Sap et al., 2019) is one such CKG dataset that organizes everyday events into if-then relationships. COMET (Bosselut et al., 2019) is a transformer language model designed on top of ATOMIC relations, showing language models can encode and generalize commonsense information.

However, Wang et al. (2021) show that language models struggle to perform generalizable commonsense inference across three popular CKG datasets: ConceptNet (Speer et al., 2017), ATOMIC (Sap et al., 2019), and TupleKB (Dalvi Mishra et al., 2017). Through a series of experiments, they found that LMs trained on several CKGs have limited ability to transfer knowledge to unseen CKGs, and that adaptation generalizes well to unseen subjects, but less so on unseen objects.

Although these graphs do well at representing facts and their relations, their statements lack context and would need to be adapted to a textual domain, such as story prose. Using them to generate a story as-is would fail to engage readers since the "story" would simply be a series of facts. Our work goes beyond the explicit structure of CKGs, focusing on finding and leveraging commonsense relations in natural language short stories.

2.2 Commonsense Inference for Storytelling

Applying commonsense reasoning on the events of a story has been proposed as one way to tackle the difficult problem of assessing the quality of machine-generated stories. The Story Cloze Test (Mostafazadeh et al., 2016) formulates story ending generation as a multiple-choice task, having systems look at several possible endings, and predict the one that is most reasonable. Guan et al. (2019) integrated commonsense reasoning directly into their Story Cloze model by building context clues and using implicit knowledge.

Commonsense reasoning can also help story generation with issues in plot coherence. The Commonsense inference Augmented neural StoryTelling (CAST) framework (Peng et al., 2021) introduces commonsense into the generation process, explicitly modeling interactions between multiple characters. The stricter, more explicit generation constraints of CAST produce more coherent and on-topic two-character stories that generating via sampling from a distribution alone.

TellMeWhy (Lal et al., 2021) is a dataset built on top of ROCStories (Mostafazadeh et al., 2016), consisting of 30k questions on why characters perform their actions and the corresponding answers. They found that current state-of-the-art models performed far worse than humans, especially on questions whose answers are external to the narratives. This contrasts with the findings discussed in Mostafazadeh et al. (2020) that language models can approach human performance.

3 The GLUCOSE Dataset and Task

Our work follows from the GLUCOSE (GeneraLized and COntextualized Story Explanations) (Mostafazadeh et al., 2020) dataset and task. This dataset provides us with a set of commonsense reasoning relations between sentences of stories. The stories come from the ROCStories (Mostafazadeh et al., 2016) corpus, a collec-

#	Description	Relation Text
1	Event that directly causes or enables X	>Causes/Enables>
2	Emotion/basic human drive that motivates X	>Motivates>
3	Location state that enables X	>Enables>
4	Possession state that enables X	>Enables>
5	Other attributes enabling X	>Enables>
6	Event that X directly causes or enables	>Causes/Enables>
7	An emotion that is caused by X	>Causes>
8	A change in location that X results in	>Results in>
9	A change of possession that X results in	>Results in>
10	Other changes in property that X results in	>Results in>

Table 1: The ten GLUCOSE dimensions and the corresponding textual "connective" that the model learns (Mostafazadeh et al., 2020).

Parameter	Text
Story	Fred woke up late. He just missed his bus. He then went to his mom's room. His mom then drives him to school. He makes it to first class on time.
Selected Sentence (X)	Fred woke up late.
Dimension	6
Specific Rule	Fred wakes up late >Causes/Enables> Fred misses his bus
General Rule	Someone_A wakes up late >Causes/Enables> Someone_A misses Something_A

Table 2: Example GLUCOSE entry (Mostafazadeh et al., 2020). Top three rows (story, X, and dimension) are input, bottom two rows (specific rule and general rule) are output.

tion of crowdsourced five-sentence everyday stories built for the Story Cloze Test (see Section 2.2).

Mostafazadeh et al. (2020) designed a multi-stage crowdsourcing platform to collect 670K annotated and quality-controlled causal reasoning relations between sentences within each story. They structured the data they collected around ten different dimensions of causal relations, inspired by cognitive psychology.

The ten dimensions are given in Table 1. They capture causal explanations between a selected sentence X from the story and another statement Y, which can either be another story sentence or external commonsense knowledge. The relationship between these statements can be formalized as:

$$statement_1 \ REL \ statement_2, \quad (1)$$

X can be in either statement position, depending on the particular dimension chosen.

The dimensions can be divided into two categories: dimensions 1-5, specifying the events that *caused* X (i.e. X is $statement_2$[1]), and dimensions 6-10, specify the events *caused by* X (i.e. X is $statement_1$). The dimensions mirror each other across the two categories. For example, dimension 1 mirrors dimension 6, depending on whether statement Y (which is an event in this example) is affecting X (1) or being affected by X (6).

3.1 Contextual Commonsense Inference Task

GLUCOSE addresses the *contextual commonsense inference* (CCI) task of predicting relationship(s) between statements explicitly or implicitly expressed within a text. The entries in their dataset are organized to reflect this and are formalized as a paired input-output tuples, with an input tuple

$$\langle \text{story } S, \text{ selected sentence } X, \text{ dimension } D \rangle, \quad (2)$$

and an output tuple

$$\langle \text{specific rule } R_S, \text{ general rule } R_G \rangle, \quad (3)$$

where a story S consists of five sentences [s_0, s_1, s_2, s_3, s_4], the selected sentence X is the sentence on which we center our rule, the number dimension D is one of the ten dimensions from Table 1, the specific rule R_S is the relation between X and Y. Y can be either (1) another sentence in the story or (2) an implicit statement from outside the text[2]. and the general rule R_G is the same relation but using generalized tags for named entities (e.g., Someone_A instead of Fred). An example entry can be found in Table 2. To summarize, the GLUCOSE task is: given S, X, and D, predict R_S and R_G.

In the original experiment, Mostafazadeh et al. (2020) curated a test set of 500 "unambiguous examples with clear gold answers" and preserved the rest of the 670K examples for training. After training several models, they ran an automated evaluation using SacreBLEU (Post, 2018), where their best-performing model was a pretrained T5 model (Raffel et al., 2020) further fine-tuned on GLUCOSE data. It achieved a 71.26 average

[1] This ordering is nearly always true in our observations.

[2] Mostafazadeh et al. (2020) gives a rough proportion of how frequently the sentence is internal or external to the story. In a manual analysis of 100 samples, they found that 63-66% of entries had "statements that contained inferences with non-story content".

Task	Input	Output
ORIGINAL	1: My mother told me to fix the car. I was unable to do this right away. * **I could not find my tools.** * I looked everywhere for them. It turns out they were stolen the night before.	They were stolen the night before >Causes/Enables> **I could not find my tools** ** Something$_A$ is stolen >Causes/Enables> Someone$_A$ cannot find Something$_A$
HISTORY	1: My mother told me to fix the car. I was unable to do this right away.	They were stolen the night before >Causes/Enables> **I could not find my tools** ** Something$_A$ is stolen >Causes/Enables> Someone$_A$ cannot find Something$_A$
MASK X	My mother told me to fix the car. I was unable to do this right away. `<masked>` I looked everywhere for them. It turns out they were stolen the night before.	They were stolen the night before >Causes/Enables> **I could not find my tools** ** Something$_A$ is stolen >Causes/Enables> Someone$_A$ cannot find Something$_A$
HISTORY+X	1: My mother told me to fix the car. I was unable to do this right away. * **I could not find my tools.** *	They were stolen the night before >Causes/Enables> **I could not find my tools** ** Something$_A$ is stolen >Causes/Enables> Someone$_A$ cannot find Something$_A$
CIS^2	1: My mother told me to fix the car. I was unable to do this right away. * **I could not find my tools.** * I looked everywhere for them. It turns out they were stolen the night before.	`<s4> >Causes/Enables> <s2>`

Table 3: Task formulations of the same GLUCOSE entry. The output is split into a specific rule and a general rule by "**", and the selected sentence *X* ("I could not find my tools") is surrounded by asterisks. In this table, we also **bolded** the selected sentence, and special tokens are `monospace`. The "1:" at the beginning of the input specifies the GLUCOSE dimension; "1" corresponds to the Causes/Enables relation. HISTORY, MASK X, and HISTORY+X are variations on the original task, ORIGINAL. CIS^2 is our proposed task.

BLEU across the 10 dimensions on predicting general rules and a 75.65 average for the specific rules[3]. They performed a human evaluation using a custom MTurk task, where workers rated model outputs (the specific and general rules) for correctness on a 4-point Likert scale[4] mapped to a numerical scale of 0-3. Their T5 model scored an average 2.5/3 across all 10 dimensions in the human evaluation (compared to 2.8/3 for gold standard answers). For context, their closest baseline got a 2.21/3 average.

3.2 Issues with the GLUCOSE Task for CCI

We take no issue with the GLUCOSE dataset, which is well-designed and of good annotation quality. However, we find issue with the GLUCOSE task, which asks a model to perform two tasks simultaneously: commonsense inference and language generation. Due to this *conflation* of tasks, the model, in generating its output, would rely heavily on the already-good language generation ability of T5 and neglect learning enough CCI.

T5 (Raffel et al., 2020) and other transformer LMs were designed to perform language *generation* tasks, such as summarization or translation. Therefore, even with the GLUCOSE task, T5 will focus on generating paraphrases of, or even directly copying, story sentences. Though our work focuses on the specific rules since it performed the best, we argue that the same issue applies to the general rule, given that only the entities are replaced with tags and the majority of the grammar remains the same.

Furthermore, there are several one-to-one correspondences between parts of the input and output in the original GLUCOSE task (illustrated in Figure 1). For example, for all GLUCOSE entries, the target output contains at least one sentence paraphrased from the input sequence. Conflation with paraphrasing worsens with BLEU as the evaluation metric, since BLEU measures word overlap, meaning even incorrect commonsense inferences can score partial credit.

4 Diagnostic Tests

In this section, we describe our three diagnostic tests—variations on the original GLUCOSE task with altered input—to isolate different factors that influence T5's generation. Through these tests, we investigate the extent to which language models rely on paraphrasing to generate the commonsense rule output for GLUCOSE.

For each of the following diagnostic tests, we finetune the same T5 (Raffel et al., 2020) model, a

[3]Our best-effort replication of their experiments achieves slightly lower BLEU scores (66.2 & 70.7, respectively) due to resource limitations (detailed in Appendix A.4).

[4]Likert scale: completely incorrect, almost incorrect, almost correct, completely correct

pretrained model using the same hyperparameters as in the GLUCOSE paper, to generate the same output as in Equation 3. The diagnostic tests differ only in the format of the input. The purpose of these tests was to assess how reliant the model is on language generation when performing CCI. More detailed training setup and hyperparameters for these models can be found in Appendix A.5.

Because these tasks are measured by BLEU score, conflation between CCI and language generation will always occur. But by deleting different parts of the input, these diagnostic tasks analyze which sentences contribute most to performance, and thus result in more conflation. Conflation always occurs for *X*, since this is known from the input, and is also worse in cases where an incorrect statement *Y* was generated, but it contains tokens that match the correct statement.

An overview of the tests' different data formats can be found in rows 2, 3, and 4 of Table 3. We describe them in this section using the following terminology for brevity:
Dimension (dim): the causal dimension
Pre-context: sentences before selected sentence X
Selected sentence (X): the story sentence of interest
Post-context: sentences after selected sentence X

ORIGINAL. This experiment is the same as in (Mostafazadeh et al., 2020), which we described in Section 3.1. We report results on our own replication of the finetuned T5 model, implemented with the `transformers` package (Wolf et al., 2019).

HISTORY. This experiment gives as input only the pre-context (the sentences before sentence *X*) and the dimension. This model must generate the output without knowing the target sentence *X*, nor the events happening afterwards. Here, we test the model's ability to generate two (specific) statements given only what happened before. This difficult task serves as a lower bound to contextual commonsense inference performance. Conflation with language generation is absent.

For all dimensions, the model must first speculate what *X* might be given the pre-context. Based on this predicted X, it generates a statement *Y* that follows from the causal relationship: either a paraphrase from the input or an implied statement.

Masked Selected Sentence (MASK X). This experiment gives as input the pre-context, post-context, and the dimension. The selected sentence is replaced with a token `<masked>`. Here, we test the commonsense ability to generate two (specific) statements given most of the story – 4 out of 5 sentences – but not the selected *X* sentence. This setup partially avoids language generation conflation.

As with HISTORY, for all dimensions, the model must first predict *X*, then generate a paraphrased or implied statement *Y* that is causally consistent.

History and Selected Sentence (HISTORY+X). This experiment gives as input the pre-context, selected sentence, and dimension. This is used as a direct comparison to HISTORY except with selected sentence *X* given as part of the input. Statement *Y* is generated as it is in HISTORY.

For these diagnostic tests, we drop entries in which the modifications result in input identical to the original task. For example, for HISTORY+X, we omit those entries where *X* is the last sentence.

model	spec	spec1-5	spec6-10	gen	gen1-5	gen6-10
ORIGINAL	70.7	67.1	74.4	66.2	62.3	70.0
HISTORY	35.9	36.9	34.9	50.4	50.1	50.7
MASK X	41.6	38.8	44.4	49.6	50.4	48.8
HISTORY+X	68.3	66.2	70.4	65.5	61.8	69.3

Table 4: Test SacreBLEU scores for diagnostic tasks. We expect that ORIGINAL performs the best, as it can access the entire input. But as we keep the output and underlying T5 LM consistent, the results' trends demonstrate how omitting different parts of the input affect BLEU scores.

4.1 Diagnostic Task Results

Table 4 compares the results of T5 models trained on the diagnostic tasks. We report test set results on the averaged dimensions 1-10, as well as averaged dimensions 1-5 (*X* is the second statement), and 6-10 (*X* is the first). Following Mostafazadeh et al. (2020), we use SacreBLEU (Post, 2018) with equal weights up to 4-grams. We report results for both specific and general rules, but focus on specific.

ORIGINAL, of course, performs the best as its input has the most available information. HISTORY and MASK X perform similarly to each other and far worse than the other diagnostic tasks. HISTORY, with only the pre-context, has a a 35-point BLEU gap for specific rules (16 for general) compared to ORIGINAL averaged across all dimensions.

Adding to HISTORY multiple sentences of the post-context gives MASK X, and modest score gains (35.9 vs 41.6 specific). However, adding to HISTORY just the one selected sentence *X* gives HISTORY+X, which performs very closely

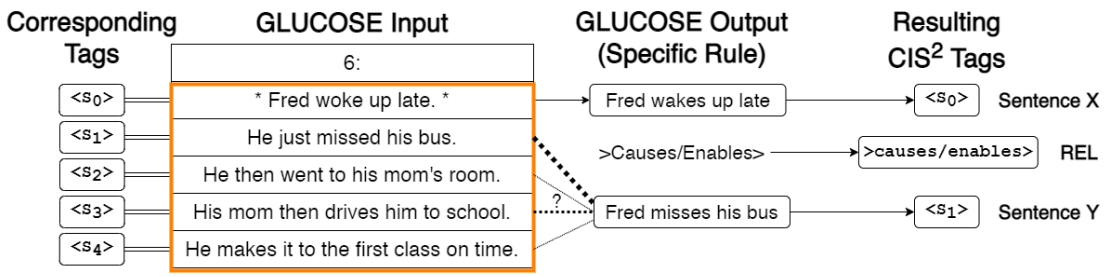

Figure 2: Generation of CIS^2 labels from a GLUCOSE entry. Each sentence in an input story (highlighted in orange) will correspond to tags <s$_0$> to <s$_4$>, depending on their position in the story. To get tags for the CIS^2 output (like in Equation 4), we do the following: (1) Find selected sentence X from the input since it always be denoted as the sentence with the asterisks surrounding it (e.g., *Fred woke up late.*, which is the first sentence, so the tag becomes <s$_0$>); (2) Get the relation REL from the output directly; and (3) Calculate the similarity of "other" sentence Y from the output to every other sentence in the input. The dashed lines represent sentence similarity comparisons with the darkest line being the highest similarity, and so <s$_1$> is selected as our sentence Y tag. The order of Sentence X and Sentence Y depends on the dimension number.

to ORIGINAL for both specific and general rules (70.7 vs 68.3 specific). Furthermore, comparing trends between dimensions 1-5 and 6-10, we find that 6-10 scores are mostly higher, for both general and specific, than 1-5.

These results and their trends show that BLEU scores are highly contingent on having X as input over all other sentences. The fine-tuned T5 models perform some CCI, but BLEU scores are hard to interpret and look unreliable. Does ~35.9 BLEU on specific rules for HISTORY mean it is half as good at CCI than ORIGINAL, with 70.7 BLEU specific? This is unlikely to be the case.

Specific vs. General Rule Performance Table 4 shows that both ORIGINAL and HISTORY+X perform better for specific rules than general. This matches the results seen in Mostafazadeh et al. (2020), but is still interesting as one might expect that general would be easier – we don't need to copy and memorize from the original input.

However, for HISTORY and MASK X, which both omit X, the opposite trend occurs. General is higher than specific, as would be intuitively expected. This shows that copying and paraphrasing from the original text is in fact a conflating factor in the LM's BLEU performance.

5 Contextual Commonsense Inference in Sentence Selection (CIS^2)

Given the extensive paraphrasing present in both the GLUCOSE task and the evaluation method, we design the Contextual Commonsense Inference in Sentence Selection (CIS^2) task to abstract away language generation. We recast the task as a classification problem, with the same 3 inputs as in ORIGINAL (Equation 2), while the output becomes

$$\langle \texttt{<s}_a\texttt{>}\ \texttt{REL}\ \texttt{<s}_b\texttt{>} \rangle \tag{4}$$

where <s$_a$> and <s$_b$> are tags corresponding to sentences from the original story, a and b are indices from $[0, 4]$ and $a \neq b$. The output sequence comes from a limited vocabulary of 5 sentence index tokens, 5 causal dimension tokens[5], and the sentence index token corresponding to the selected sentence X can be before or after the REL token, depending on what causal dimension is being used. the The classification task is to choose the correct sequence of 100 possible output sequences[6].

The abstracted output avoids the prior conflation issue since there are no partial matches within tokens of statements. Furthermore, there is no explicit correspondence between input and output. Note that CIS^2 does not distinguish between specific and general rules.

Finetuned CIS^2 models are forced to only learn the commonsense inference task. The input is kept the same, so the models see the same information as with the original task formulation. Therefore, we argue that CIS^2 is a simpler and fairer measurement of commonsense inference performance.

5.1 GLUCOSE Entries to CIS^2 Tag Heuristic Conversion

To evaluate the CIS^2 formulation, we need to convert story sentences into CIS^2 output labels (as

[5] >Causes/Enables>, >Causes>, >Enables>, >Results in>, >Motivates>

[6] 5P2 = 20 * 5 relation texts = 100 possibilities

in Equation 4). This method is illustrated in Figure 2. Two of the three CIS^2 tokens are immediately given from the input. The dimension specifies REL, as well as the position of sentence X—whether it comes first (corresponding to <s_a>) or last (<s_b>). Where X is in the input story specifies the index of one of the tokens. Sentence X is easy to find since it is a direct match across the input and output of the original GLUCOSE task.

To find the remaining token, we look at the specific rule from the original GLUCOSE task output, which consists of two statements separated by relation REL. We will call them P_0 and P_1. Suppose X corresponds to P_0, and we need to find which sentence Y corresponds to P_1. We do this by iterating over the sentences (excluding X), for each calculating its similarity with P_1. We take the index of the sentence with the highest similarity to P_1 as <s_b>. We experimented with several sentence similarity metrics, and found Sentence-BERT (Reimers and Gurevych, 2019) the most effective.

Being a heuristic approach, generated CIS^2 labels are not guaranteed to be perfect. However, a cursory manual inspection finds most labels are reasonable for GLUCOSE entries that have an explicit Y (found within the story). CIS^2 labels are likely incorrect for implicit relationships, which are a minority of the entries. Therefore we do not attempt to filter them out. Future work remains to distinguish implicit and explicit GLUCOSE relationships, and either filter implicit ones out, or handle them separately.

We run the conversion heuristic on the GLUCOSE train set and train a T5 model using the same hyperparameters used for our other models with the task of generating the three-token CIS^2 label, given the GLUCOSE input. We refer to this model as CIS^2-TRAIN.

5.2 CIS^2 Results

To compare ORIGINAL and the GLUCOSE diagnostic models to CIS^2, we run the conversion method from Section 5.1 on each model's specific rule output to obtain its predicted CIS^2-like labels. We do the same conversion on the GLUCOSE test set to obtain the CIS^2 test set[7]. This enables us to do an exact-match comparison between the model labels and the test set labels, and removes the associated issues with evaluating generated text.

With all outputs converted to CIS^2-like labels

[7]We will obtain ground-truth test labels via crowdsourcing.

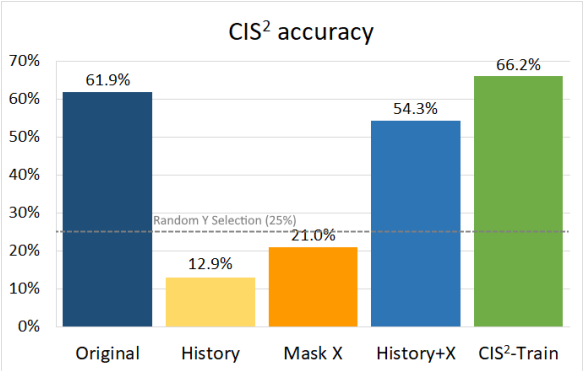

Figure 3: CIS^2 accuracy results for Original and diagnostic GLUCOSE task models, and CIS^2-TRAIN. The dashed line shows Random Y Selection, a baseline that derives X and the relation text from the input, and randomly selects Y.

we calculate the accuracy for how often the correct sentence Y is actually being chosen. Results for this evaluation are shown in Figure 3.

Results are compared to a simple model that infers the index of X and REL from the input randomly select one of the 4 other story sentences for the index of Y – achieving a 25% accuracy. CIS^2-TRAIN achieves the highest score at 66.2%, while ORIGINAL is not far behind at 61.9%. As for the diagnostic tasks, we see the same score ordering of models with BLEU evaluation. HISTORY+X scores 8% lower than ORIGINAL. HISTORY and MASK X perform even worse than random, indicating that their BLEU performance was almost entirely due to partial token matches.

The best GLUCOSE model ORIGINAL achieves 70.7 specific BLEU, but only 61.9% CIS^2 accuracy. Although we cannot directly compare BLEU and CIS^2 exact match numbers, we can see that CIS^2 provides a fairer estimate of the CCI performance of these fine-tuned T5 models by removing language generation from evaluation. These CCI results are promising, and yet there is still much room for improvement.

6 Discussion

The diagnostic tasks we discussed in the paper investigated the extent to which the original GLUCOSE task conflates language generation and contextual commonsense inference (CCI). We found that the most significant sentence of the input is the selected sentence X, and if omitted, BLEU scores drop significantly compared to omitting other story sentences. This shows that the language model is

relying on X, as it should, for CCI. We have shown that the T5 model trained on the GLUCOSE task (to maximize BLEU on the specific and general rules) performs only 4.3% worse on the CIS^2 than one trained directly on CIS^2 labels. This shows that T5 can still learn significant CCI from the GLUCOSE data, and can further improve performance with CIS^2 converted labels, abstracting away with language generation.

We have also shown evidence for extensive copying and paraphrasing as seen from the higher performance on specific rules relative to general rules for ORIGINAL and HISTORY+X. These trends hold for CIS^2 evaluation as well, but are even more marked (since there is no inflation from matching tokens).

It is worth discussing how "fair" it is to remove X – after all, without X, LMs have little to condition their predictions on. While this is true, we emphasize that our diagnostic tasks are intended to be taken together to analyze the extent of conflation. The main takeaway is that by including X, trained models will rely on good language generation instead of good commonsense inference.

Future work can further explore utilizing GLUCOSE and related datasets for story generation tasks. We believe our diagnostic tasks can be easily be extended to allow using GLUCOSE in various story generation settings. HISTORY and HISTORY+X models for continuing stories, and MASK X models for story infilling (Ippolito et al., 2019).

We especially resonate with the question-answering based approach to commonsense inference for stories (also on ROCStories) of Lal et al. (2021). They trained large language models on their dataset, finding that they only perform well when the answers are present in the narrative. This finding goes hand in hand with our finding that the original GLUCOSE task formulation allows for easy paraphrasing and thus inflated performance.

7 Conclusion

This work investigated the extent to which language models learn contextual commonsense inference (CCI), utilizing the GLUCOSE (Mostafazadeh et al., 2020) dataset and the T5 (Raffel et al., 2020) language model as case studies. We showed how the original GLUCOSE task conflates language generation and CCI tasks, causing over-estimation of true CCI performance. We then formulated diagnostic tasks by permuting the original task and found that LMs rely on paraphrasing the selected sentence and context in making their predictions.

We proposed CIS^2 as an alternative task to structure and evaluate language models for CCI. CIS^2 evaluation is a simplified, fairer measurement of CCI performance over BLEU. By finetuning a T5 model on our CIS^2 task, it correctly selects the causal statement 4.3% more than a model trained on the original GLUCOSE task. We note this is using heuristically converted CIS^2 labels, and collecting ground-truth CIS^2 labels for training would lead to even better performance.

Overall, we found that GLUCOSE indeed encodes contextual commonsense information, and T5 has capacity to learn this. Therefore, the challenge for future researchers is to leverage GLUCOSE and other contextual commonsense inference datasets' knowledge representations appropriately and avoid conflation of language generation.

References

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. page 4762–4779.

Bhavana Dalvi Mishra, Niket Tandon, and Peter Clark. 2017. Domain-targeted, high precision knowledge extraction. *Transactions of the Association for Computational Linguistics*, 5:233–246.

Jian Guan, Yansen Wang, and Minlie Huang. 2019. Story Ending Generation with Incremental Encoding and Commonsense Knowledge. In *AAAI Conference on Artificial Intelligence*, pages 6473–6480.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):6384–6392.

Daphne Ippolito, David Grangier, Chris Callison-Burch, and Douglas Eck. 2019. Unsupervised hierarchical story infilling. In *Proceedings of the First Workshop on Narrative Understanding*, pages 37–43, Minneapolis, Minnesota. 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.

Lori Moon, Lauren Berkowitz, and Nasrin Chu-Carroll, Jennifer Mostafazadeh. 2020. Details of data collection and crowd management for glucose.

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.

Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. GLUCOSE: GeneraLized and COntextualized story explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4569–4586, Online. Association for Computational Linguistics.

Xiangyu Peng, Siyan Li, Sarah Wiegreffe, and Mark Riedl. 2021. Inferring the Reader: Guiding Automated Story Generation with Commonsense Reasoning. *arXiv preprint arXiv:2105.01311*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. 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*, 21:140:1–140:67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 3982–3992.

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.

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. Do massively pretrained language models make better storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861, Hong Kong, China. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Thirty-first AAAI Conference on Artificial Intelligence (AAAI)*, page 4444–4451.

Tom Trabasso, Tom Secco, and Paul van den Broek. 1984. Causal cohesion and story coherence.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *Conference on Advances in Neural Information Processing Systems (NeurIPS)*, pages 1–15.

Peifeng Wang, Filip Ilievski, Muhao Chen, and Xiang Ren. 2021. Do Language Models Perform Generalizable Commonsense Inference? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, page 3681–3688.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

A Appendix

A.1 Acknowledgements

We thank the authors of GLUCOSE, in particular Or Biran and Lori Moon, for their helpful assistance in working with the GLUCOSE dataset and codebase.

A.2 Ethical Considerations and Broader Impacts

The methods used in our paper build in large part upon work by prior researchers. The T5 (Raffel et al., 2020) language model we used was pre-trained on a massive dataset for many days. Despite the energy usage, T5 has proved be a valuable tool that can be used for countless downstream NLP applications, ours included. As for our own trained models, we note that we further fine-tuned T5 on an array of diagnostic and custom tasks. During development, we made sure to pilot any experiments on smaller datasets, and we carefully managed our GPU and CPU usage throughout.

As for the data used, the ROCStories (Mostafazadeh et al., 2016) and GLUCOSE (Mostafazadeh et al., 2020) datasets involved a great deal of careful task design and work with crowd-source workers. We thank these researchers for their ethical treatment of their crowd-source workers, with fair pay and two-way communication (Moon et al., 2020).

We will publicly release all our code, from data preprocessing, to model training, to final evaluation, to ensure that our work is fully reproducible.

The broader impacts of our work outside its immediate subject are several. First, our work takes a step towards analyzing stories, which are something fundamentally human, and that machines have yet to master. Second, we have encouraged NLP researchers in general to think more carefully about the structure of a task, before defaulting to the latest state-of-the-art language model. For example, we found that our CIS^2 task, which is simpler and thus requires less training resources than the language generation task, performs better on capturing contextual commonsense inference.

A.3 Reproducing Our Work

We make our code publicly available at a Github link. The codebase includes complete preprocessing, training, and evaluation scripts, to take the raw GLUCOSE CSVs and T5 checkpoints, and train both diagnostic and CIS^2 models. We will also release the final trained checkpoints.

We also include our code to reproduce the original GLUCOSE experiments. We model this closely to the original GLUCOSE paper, starting from their provided code repository.

A.4 Reproduction Results

We report the results we obtained on the original GLUCOSE task in Table 5. We report per-dimension BLEU, as was done prior, as well as the weighted average BLEU across all dimensions. We find that the reported numbers from (Mostafazadeh et al., 2020) and their provided Tensorflow checkpoint are essentially consistent.

Our replication results (done with the `transformers` package (Wolf et al., 2019)) achieve 4-5 BLEU points lower, due to resource limitations and slight differences in experimental setup (i.e. we had far less GPU resources and and training time). For consistency's sake all of our experiments use the same setup as replicated t5-large (termed Original in the main text), and thus use this as the baseline.

We report results on the test set, but choose to evaluate BLEU on only the first of the three provided references for each test set entry. This is because the GLUCOSE train set only has one reference per entry, not 3, and we carved a small development set out of the train set, since no train/development split was provided. We evaluate our custom development and the original test set the same way, with 1 reference per entry.

A.5 Training Setup and Hyperparameters

We trained our models on 2 NVIDIA Quadro RTX 6000 GPUs, with 24 GB vRAM each. We train up to 10 epochs, early stopping after 10 checkpoints without improvement on the validation set. Depending on the task, the models finish training between 6 to 34 hours. The GLUCOSE authors trained their model far more – for 72 hours on 8 TPUs – which can explain our lower BLEU scores.

We use the exact same hyperparameters as in Raffel et al. (2020), following Mostafazadeh et al. (2020), with one major exception: we use a learning rate of 1e-4 instead of 1e-3, which we found to converge too quickly.

Model	Level	avg	1	2	3	4	5	6	7	8	9	10
(Mostafazadeh et al., 2020)	Specific	N/A	72.5	73.8	70.5	81.1	71.7	73.9	79.3	80.2	86.6	66.9
(Mostafazadeh et al., 2020)	General	N/A	66.4	68.5	69.8	76.8	68.6	67.6	73.0	77.0	86.8	57.5
GLUCOSE TF-checkpoint	Specific	75.7	71.9	69.8	75.8	75.9	73.3	75.2	79.8	80.2	85.5	69.9
GLUCOSE TF checkpoint	General	70.1	66.4	66.4	70.1	72.1	70.0	69.2	71.6	72.4	82.0	61.0
replicated t5-large	Specific	70.7	65.9	60.4	63.8	76.5	69.0	66.7	72.6	74.0	82.4	76.0
replicated t5-large	General	66.2	61.3	59.9	60.4	68.8	61.3	60.5	65.0	68.1	75.8	80.4

Table 5: Test Set Results for the original GLUCOSE task. The first rows are the original results, the second are decoded by us using the provided GLUCOSE TF checkpoint, and the third are our best-effort replications.