# COINS: Dynamically Generating COntextualized Inference Rules for Narrative Story Completion

**Debjit Paul**
Research Training Group AIPHES
Institute for Computational Linguistics
Heidelberg University
paul@cl.uni-heidelberg.de

**Anette Frank**
Research Training Group AIPHES
Institute for Computational Linguistics
Heidelberg University
frank@cl.uni-heidelberg.de

## Abstract

Despite recent successes of large pre-trained language models in solving reasoning tasks, their inference capabilities remain opaque. We posit that such models can be made more interpretable by explicitly generating interim inference rules, and using them to guide the generation of task-specific textual outputs. In this paper we present COINS, a recursive inference framework that i) iteratively reads context sentences, ii) dynamically generates contextualized inference rules, encodes them, and iii) uses them to guide task-specific output generation. We apply COINS to a *Narrative Story Completion* task that asks a model to complete a story with missing sentences, to produce a coherent story with plausible logical connections, causal relationships, and temporal dependencies. By modularizing inference and sentence generation steps in a recurrent model, we aim to make reasoning steps and their effects on next sentence generation transparent. Our automatic and manual evaluations show that the model generates better story sentences than SOTA baselines, especially in terms of coherence. We further demonstrate improved performance over strong pre-trained LMs in generating commonsense inference rules. The recursive nature of COINS holds the potential for controlled generation of longer sequences.

## 1 Introduction

Narrative story understanding, and similarly story generation, requires the ability to construe meaning that is not explicitly stated through commonsense reasoning over events in the story (Rashkin et al., 2018a). Previous work in modeling narrative stories has focused on learning scripts[1] (Schank and Abelson, 1977; Mooney and DeJong, 1985) and learning narrative schemas using corpus statistics (Chambers and Jurafsky, 2009; Balasubramanian et al., 2013; Nguyen et al., 2015). Recently, large pretrained language models (LMs) such as GPT-2 have shown remarkable performance on various generation tasks. While these pretrained LMs learn probabilistic associations between words and sentences, they still have difficulties in modeling causality (Mostafazadeh et al., 2020). Also, in narrative story generation, models need to be consistent with everyday commonsense norms. Hence, to address a story generation task, i) models need to be equipped with suitable knowledge, ii) they need effective knowledge integration and reasoning methods, and ideally iii) we want to be able to make the effectiveness of these methods transparent.

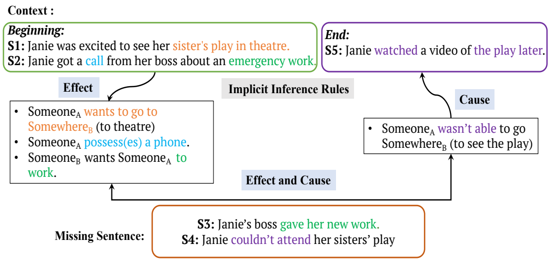

Context :
Beginning:
S1: Janie was excited to see her sister's play in theatre.
S2: Janie got a call from her boss about an emergency work.
End:
S5: Janie watched a video of the play later.
Effect
Implicit Inference Rules
Cause
Someone$_A$ wants to go to Somewhere$_B$ (to theatre)
Someone$_A$ possess(es) a phone.
Someone$_B$ wants Someone$_A$ to work.
Someone$_A$ wasn't able to go Somewhere$_B$ (to see the play)
Effect and Cause
Missing Sentence:
S3: Janie's boss gave her new work.
S4: Janie couldn't attend her sisters' play

Figure 1: An example of the *Narrative Story Completion Task*. Top and bottom boxes show the context (top) and missing sentences (bottom). The chain of implicit inference rules explains the connection between beginning and end, and allows to infer the missing sentences.

In this work we focus on the aspects i) to iii), by investigating new methods that build on pretrained LMs to generate missing sentences from an incomplete narrative story. Specifically, we focus on *Narrative Story Completion (NSC)*, a new task setting for story generation. Given an incomplete story, specified only through its beginning and ending, the task is to generate the missing sentences to complete the story (see Figure 1). Our hypothesis is that in order to obtaining a consistent and coherent

[1] Scripts are structured knowledge about stereotypical event sequences together with their participants.

*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 5086–5099
August 1–6, 2021. ©2021 Association for Computational Linguistics

narrative story, the task requires a model's ability to perform commonsense inference about events and entities in a story. Unlike other existing tasks, NSC requires: *i)* generating *multiple sentences* to complete a story, and *ii) ensuring that* the generated sentences are *coherent* with respect to both *beginning and ending* of the story. Hence, the NSC task offers a challenging setup for investigating the reasoning capacities of a story generation model.

Humans excel in drawing inferences and constructing causal chains that explain the connection between events (Kintsch and Dijk, 1978). Figure 1 illustrates this with an example from our *NSC* task.[2] From *Janie was excited to see her sister's play in theatre*$_{(s_1)}$. *Janie got a call from her boss about new work*$_{(s_2)}$ and the outcome *Janie watched a video of the play later.*$_{(s_5)}$ – we can construct inference rules in forward and backward direction: forward via EFFECT: Someone$_B$ (*boss*) gave work to Someone$_A$ (*Janie*); backward via CAUSE: Someone$_A$ (*Janie*) wasn't able to go Somewhere$_B$ (*to the theatre*). By combining these inferences, we can obtain a representation from which to generate a connection that completes the story, e.g., *Janie's boss wanted her to look after the issue*$_{(s_3)}$. *She missed the theatre play*$_{(s_4)}$.

In this work, we propose COINS: a recursive model that jointly learns to i) *dynamically* generate commonsense inference rules[3] grounded in the context and to ii) perform *controled and coherent* story generation, using the generated inferences as a guide. We hypothesize that jointly learning to generate contextualized inference rules from *dynamically predicted contextualized inference rules* and learning to generate story sentences *incrementally* while taking the inferences into account, will improve the quality of *both* the predicted inference rules and of generated story sentences. Moreover, the recursive nature of the model and the *individuation* of the inference prediction and sentence generation tasks make the process more *interpretable*: the generated inference rules can be viewed as intermediate representations, and can serve as *explanations* of how the dynamically produced inferences influence the quality of generated story sentences.

Our main contributions are as follows:

1) We propose a new setting for a Narrative Story Completion task, which asks a system to complete a narrative story given its beginning and ending, with the aim of examining the reasoning capacities of a model that solves the task.

2) We propose an integrated reasoning and NL generation model, COINS, that based on its current context generates contextualized commonsense inference rules and follow-up sentences, in a stepwise recurrent process.

3) We conduct extensive experiments with automatic and human evaluation. Automatic evaluations show that COINS outperforms strong baselines (+2.2 BLEU score). Human evaluation shows that compared to strong baselines, our model yields better sentence generations with respect to *coherence* ($+50.5\%$) and *grammaticality* ($+20.5\%$).

4) We show that COINS generates better inference rules (+2.3 BLEU score) compared to a *fine-tuned* GPT-2 model, and that jointly learning to generate inferences and story sentences improves the quality of the generated inference rules.

Our code is made publicly available.[4]

## 2 Related Work

**Sentence-level Commonsense Inference and Beyond.** Recent research in this area has focused on commonsense knowledge acquisition (Sap et al., 2019; Zhang et al., 2020; Speer et al., 2017; Malaviya et al., 2020) and commonsense reasoning (Zellers et al., 2019; Talmor et al., 2018). In our work, we focus on inferential knowledge about events, and entities participating in such events. Rashkin et al. (2018b) introduced a knowledge resource of commonsense inferences regarding people's intents and reactions towards a diverse set of events. With COMET, Bosselut et al. (2019) have shown that pre-trained neural language models can be fine-tuned using large knowledge bases (such as ATOMIC, Sap et al. (2019)) to generate inferences for a given event or sentence. However, the generated knowledge from COMET is non-contextualized and hence, can be inconsistent. Recently, Mostafazadeh et al. (2020) proposed GLUCOSE, a new resource and dataset that offers semi-structured commonsense inference rules that are *grounded* in sentences of specific stories. They show that fine-tuning a pre-trained LM on the GLUCOSE dataset helps the model to better generate inferrable commonsense explanations given a *complete* story. In concurrent work, Gabriel et al. (2021) proposed PARA-COMET, a model that in-

[2] We use the ROCstories dataset to frame the *NSC* task.

[3] In this paper, similar to Mostafazadeh et al. (2020), we will use "inference rule" and "explanation" interchangeably.

[4] `https://github.com/Heidelberg-NLP/COINS`

corporates paragraph-level information to generate coherent commonsense inferences from narratives. In this work, we investigate how well a neural model can generate contextualized commonsense inference rules for an *incomplete* story. Learning to predict iterative inference steps for successive events in a narration using semi-structured knowledge rules is still a difficult and underexplored task. We propose a model that learns to iteratively generate a coherent completion of an incomplete narrative story utilizing semi-structured knowledge as offered by the GLUCOSE framework.

**Commonsense Reasoning in Narrative Stories.** Early work on narrative events focused on *script learning*, by defining stereotypical event sequences together with their participants (Schank and Abelson, 1977). In later works, Chambers and Jurafsky (2008, 2009); Balasubramanian et al. (2013); Nguyen et al. (2015); Pichotta and Mooney (2014) proposed methods to learn *narrative event chains* using a simpler event representation that allows for efficient learning and inference. Chambers and Jurafsky (2009) acquired Narrative Event Schemata from corpora and established the Narrative Cloze Task (Chambers and Jurafsky, 2008) that evaluates script knowledge by predicting a missing event (verb and its arguments) in a sequence of observed events. More recently, Mostafazadeh et al. (2016) proposed the *story cloze task* that selects a plausible (right) over an implausible (wrong) story ending. Bhagavatula et al. (2020) proposed an *abductive reasoning task* to test a model's ability to generate plausible explanations for an incomplete set of observations. Paul and Frank (2020) proposed a multi-head knowledge attention method to dynamically incorporate non-contextualized inferential knowledge to address the *abductive reasoning task*. Qin et al. (2020) proposed an unsupervised decoding algorithm that can flexibly incorporate both the past and future contexts using only off-the-shelf language models to generate plausible explanations. Concurrent to our work, Paul and Frank (2021) presented a method for addressing the *abductive reasoning task* by explicitly learning what events could follow other events in a hypothetical scenario. In our work, we make use of the ROCStories dataset (Mostafazadeh et al., 2016) to build a *Narrative Story Completion* task that tests a model's ability of *generating* missing sentences in a story. We propose a model that aims to produce *coherent* narrative stories by performing iterative commonsense inference steps.

**Narrative Story Generation.** Much existing work on story generation relied on symbolic planning methods (Lebowitz, 1987; PÉrez and Sharples, 2001; Józefowicz et al., 2016). With the advances of Seq2Seq models, several works applied them in automatic story generation tasks (Roemmele, 2016; Jain et al., 2017). Fan et al. (2018) proposed a hierarchical approach to generate short stories from initial prompts. Recently, many works have focused on integrating external commonsense knowledge from large static knowledge bases like ATOMIC (Sap et al., 2019) or ConceptNet (Speer et al., 2017) for different tasks such as story ending generation (Ji et al., 2020; Guan et al., 2019) or story generation (Guan et al., 2020; Xu et al., 2020). In concurrent work, Ammanabrolu et al. (2021) look into causality for a commonsense plot generation task. In our work, we model the assumption that contextualized inference rules provide *inferred* information that can guide a system in generating both *contextually grounded* and *coherent* follow-up sentences in a story generation task.

## 3 Task Definition

We formulate the *Narrative Story Completion task (NSC)* as follows: given an incomplete story ($S$= $s_1, s_2, s_n$) as a sequence of tokens $t = \{t_1, t_2, ..., t_{SEP}, ..., t_m\}$ (with $t_{SEP}$ a mask token delimiting $s_2$ and $s_n$), the goal is to generate the missing sentences ($s_3, ..., s_{n-1}$) as a sequence of tokens $y^{s_i}$=$\{y_1^{s_i}, y_2^{s_i}, ..., y_v^{s_i}\}$ (with $i$ = 3, ..., $n{-}1$ and $v$ the maximum length of each sentence).

In the setting of the NSC task, we expect the completed story to be coherent. That is, the generated sentences should exhibit reasonable logical connections, causal relationships, and temporal dependencies with each other and the given beginning and ending of the story. In this paper, we define a discourse to be coherent if successive sentences that are about the same entities, and the reported events involving them can be construed to reflect common knowledge about how events are typically connected in a temporal sequence or by causal relations. Similar to Hobbs (1985), the criteria to conclude that discourse is coherent include require that there are reflections of causality in the text.

Our take on this task is to incrementally generate contextualized inference rules from the given context, and to make use of this knowledge to generate missing story sentences.

| Relation Type | Dimensions |
|---|---|
| **Cause**<br>(Dim 1-5) | (1) Event that directly causes or enables X; (2) Emotion or basic human drive that motivates X; (3) Location state that enables X; (4) A possession state that enables X; (5) Other attribute that enables X. |
| **Effect**<br>(Dim 6-10) | (6) An event that is directly caused or enabled by X; (7) An emotion that is caused by X; (8) A change of location that X results in; (9) A change of possession that X results in; (10) Other change in attribute that X results in. |

Table 1: Causal Relation types and their mapped relations (Mostafazadeh et al., 2020).

| | |
|---|---|
| Incomplete Story: | $s_1$: Jane loved cooking. $s_2$: Everyone else in her family did too. $s_5$: **Eventually she learned everything there was to teach.** |
| *Gold*: | Someone$_A$ loves Something$_A$ (that is an activity ) >CAUSES/ENABLES> Someone$_A$ learns everything there is to learn. |
| | Jane loves cooking >CAUSES/ENABLES> Jane learns everything there is to learn |
| *COINS*: | Someone$_A$ is a quick learner >CAUSES/ENABLES> Someone$_A$ learns everything there is to learn. |
| | Jane is a quick learner >CAUSES/ENABLES> Jane learns everything there is to learn. |

Table 2: Example of inference rules generated by COINS (compared to *Gold* from GLUCOSE). Grey: context-specific rules (SR); regular: general rules (GR). Bolded sentence $s_5$ is X, CAUSE is the relation type $r$.

| Dataset | Relation Type | Train | Dev | Test |
|---|---|---|---|---|
| **NSC** | | 88,344 | 4,908 | 4,909 |
| **GLUCOSE** | **Effect** | 2949 | 849 | – |
| | **Cause** | 2944 | 916 | – |

Table 3: Dataset Statistics: number of unique stories.

## 4 Discourse-Aware Inference Rules

This section details how we construct training data for the NSC task, by enriching stories with automatically predicted contextualized inferences.[5] We utilize the GLUCOSE (Mostafazadeh et al., 2020) dataset, which contains implicit commonsense knowledge in form of semi-structured general and specific inference rules[6] (cf. Table 1) that are grounded in the context of individual stories from ROCStories. In GLUCOSE, given a story $S$ and a selected sentence $X$ from the story, the authors define ten dimensions $d$ of commonsense causal explanations related to $X$, inspired by human cognitive psychology. Only a small part of ROCStories is annotated with GLUCOSE inferences (Table 3).

Given the amount of commonsense knowledge needed for real-world tasks, a static knowledge resource is always incomplete. Thus, we *fine-tune* a pre-trained GPT-2 model on the annotated part of GLUCOSE to *dynamically* generate inference rules for each sentence $X_i$ of each story $S_i$ from the underlying ROCStories data. We *fine-tune* two separate language models $CSI_{gen}$ and $CSI_{spec}$ for general and specific rules, respectively (Table 2).

The 10 dimensions $d$ in GLUCOSE cover implicit *causes* and *effects* of a sentence $X$ in a given story. In our work, we are interested in inference rules that explain a sentence's causes and effects, to study the impact of such inferences on narrative story completion. We therefore cluster all dimensions $d$ into the two categories EFFECT vs. CAUSE (Table 1) and aggregate all rules from the respective categories (preserving their dimensions). Once our models ($CSI_{gen}$, $CSI_{spec}$) are trained, we apply them to our *NSC* task training data, to enrich it with inference rules for each sentence and story.

## 5 COINS: COntextualized Inference and Narrative Story Completion Model

In this section we introduce a recursively operating reasoning and sentence generation model: COINS. An overview is given in Figure 2. In each iteration, the model applies two consecutive steps:
(1) *Inference Step*: Given an *incomplete story context* $S'$= $X \oplus S_i$ and relation $r$, an *inference model* $CSI$ ($gen$ or $spec$) generates COntextualized inference rules of type $r$.
(2) *Generation Step*: a *sentence generator* reads the generated inference rules concatenated with the current context $S'$ and generates the next story sentence $s_{i+1}$. The context $S'$ is updated with $s_{i+1}$ and steps (1) and (2) are repeated (cf. Algorithm 1).

This formulation allows us to i) examine inference and generation capabilities separately from each other, ii) helps determine the impact of inferential knowledge on story generation, and iii) can give us insight into how knowledge can guide story generation in a recursive inference framework.

**Inference Step.** We define the initial story context $S'$ = $\{s_1, s_2$,[SEP], $s_n\}$, a selected sentence as $s_i$, and relation type $r \in \{$EFFECT, CAUSE$\}$, where $i \in [2, \ldots n$-1], $s_i$=$\{w_1^{s_i}, .., w_v^{s_i}\}$. We adopt a pretrained GPT-2 (base) (Radford et al., 2019) transformer model with multiple Transformer blocks of multi-head self-attention and fully connected layers. During training, in each iteration the input to the model is a concatenation of the current source ($S', s_i, r$) and target sequence i.e., the inference

[5] For testing we rely on GLUCOSE's manually validated inference rules on a small subset of the ROCStories corpus.

[6] *Specific* means rules grounded in a given context and *general* corresponds to rules that are applicable to other contexts.

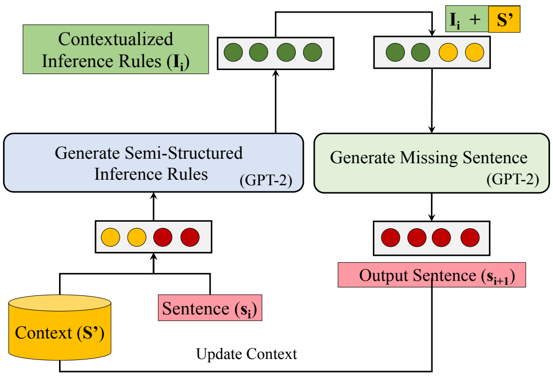

Contextualized Inference Rules (I_i)
I_i + S'
Generate Semi-Structured Inference Rules (GPT-2)
Generate Missing Sentence (GPT-2)
Output Sentence (s_{i+1})
Sentence (s_i)
Context (S')
Update Context

Figure 2: Architecture of the COINS model.

**Algorithm 1** COINS

```
Input: Initial Context ($S' = \{s_1, s_2, [SEP], s_n\}$)
1: $Mem_{\mathcal{IR}} \leftarrow$ empty
2: $\mathcal{G}en\mathcal{S} \leftarrow$ empty list
3: for $i \leftarrow 2$ to $n-1$ do
4:     $E_i$ = GenInferenceRules($S'$, $s_i$, EFFECT)
5:     $C_i$ = GenInferenceRules($S'$, $s_n$, CAUSE)
6:     $I_i = E_i \oplus C_i$
7:     $s_{i+1}$ = GenNewSentence($I_i$, $S'$)
8:     $\mathcal{G}en\mathcal{S}$ := $\mathcal{G}en\mathcal{S}$ + $s_{i+1}$
9:     $Mem_{\mathcal{IR}}$ := $Mem_{\mathcal{IR}} \oplus I_i$
10:    $\mathcal{L}_{\mathcal{S}}$ += $-log_{p_{(\theta)}}(s_{i+1}|I_i, S')$ $-log_{p_{(\beta)}}(I_i|S')$
11:    $\mathcal{L}_{\mathcal{IR}}$ += $-log_{p_{(\theta)}}(s_{i+1}|I_i, S')$ $-log_{p_{(\beta)}}(I_i|S')$
12:    $S' := \{s_1, s_2, s_{i+1}, [SEP], s_n\}$
13: end for
14: return $\mathcal{G}en\mathcal{S}$, $Mem_{\mathcal{IR}}$
```

rules ($E_i$ or $C_i$). Eq. (1) defines the inference rule ($\mathcal{IR}$) generation model:

$$\begin{aligned} h_p^0 &= e_p + P_p, \\ h_p^l &= block(h_{<p}^{l-1}), l \in [1, L] \\ p(y_p|y_{<p}, p) &= softmax(h_p^L W^T) \end{aligned} \quad (1)$$

where $h_p^0$ is a summation of token embedding $e_p$ and position embedding $P_p$ for the $p$-th token; $h_p^l$ is the $l$-th layer's output at position $p$, computed through transformer blocks with the masked multi-head self attention mechanism; $h_p^L$ is the final layer's hidden state and $y_{<p}$ indicates the left context of position $p$. The softmax layer defines the model to output the most probable target sequence: the most likely inference rules ($E_i$ and $C_i$) for each relation type (cf. Algorithm Line 4-5).

During training, we minimize the objective (2)

$$\begin{aligned} \mathcal{L}_I(\beta) = &- \sum_{k=m}^{m+N} log\ p(E_i^k|S', s_i, \textsc{Effect}) \\ &- \sum_{k=m}^{m+N} log\ p(C_i^k|S', s_n, \textsc{Cause}) \end{aligned} \quad (2)$$

where $m, N$ denote the number of tokens in the source $(S', s_i, r)$ and target sequence (inference rules) respectively; $\beta$ refers to model parameters.

In this work, we focus on the NSC task, which requires our model to capture temporal dependencies and causal relationships between events. While we designed our sentence generation model in such a way that it can utilize inference rules from both forward and backward directions for each sentence, we here trigger the generation of CAUSE inference rules for $s_n$, since we expect that *events*, *motivations* or *attributes* that **cause** $s_n$ will be relevant for generating the preceding sentences $[s_3, \ldots s_{n-1}]$. Similarly, we generate EFFECT relations for $s_i$, assuming that an *event*, changes of *emotion* or changes of *attribute* that are possible **effects** caused by $s_i$ will be most relevant for generating the missing follow-up sentences. In principle, however, for NSC and other story generation tasks, we may consider CAUSE and EFFECT relations for all sentences, letting the model freely choose from the full space of inferences.

We concatenate the generated inference rules ($I_i = E_i \oplus C_i$)[7] and store the last hidden representation in $Mem_{\mathcal{IR}} \in \mathbb{R}^{N \times L \times H}$, where $N$ is the number of sentences, $L$ the maximum inference sequence length and $H$ the hidden state dimensions. $Mem_{\mathcal{IR}}$ is updated with the hidden representations of inference rules in each iteration. Hence, $Mem_{\mathcal{IR}}$ could act as an intermediate representation, and as a basis for providing *explanations* for observed story sentence generations. $Mem_{\mathcal{IR}}$ may also be used as a memory for long-form text generation tasks, to keep track of implicit knowledge *triggered by* previously generated text, and could support flexible discourse serialization patterns.[8]

**Generation Step.** Given the generated inference rules $I_i$ (in form of tokens) and the incomplete story context $S'$, we aim to generate the next missing sentence. We pass the input through another pretrained GPT-2 (base) model (cf. Equation 1). The loss function for the sentence generator is

$$\mathcal{L}_S(\theta) = -\sum_{k=1}^{v} log\ P(y_k^{s_{i+1}}|I_i, [EOK], S') \quad (3)$$

where $y_k$ denotes the $k$-th token and $v$ the maximum length of the generated sentence;

[7] We use $[SEP]$ token to delimit the individual E$_i$ and C$_i$ when concatenating them.

[8] We leave such extensions to future work.

$i \in [2, n-1]$ ; $[EOK]$ denotes the end of knowledge rule tokens, and $\theta$ refers to model parameters.

**Update Story Context.** In the final step we update the story context by inserting the generated sentence $s_{i+1}$ into the previous story context (cf. Algorithm 1, line 12).

**Training and Inference.** We add the losses $\mathcal{L}_\mathcal{I}$ for inference generation and $\mathcal{L}_\mathcal{S}$ for sentence generation to make the models dependent on each other (Algorithm 1, line. 10-11). For both the inference and the generation step model, we minimize the negative log likelihood loss of the respective target sequence.

## 6 Experiments

### 6.1 Dataset

We apply COINS to the *NSC* and the *Story Ending Generation* tasks.[9] For data statistics see Table 3.

**Narrative Story Completion.** We follow the task definition as introduced in §3.

*Data Collection.* We construct the *NSC* dataset on the basis of the ROCStories corpus (Mostafazadeh et al., 2016), which contains 98,162 five-sentence stories with a clear beginning and ending, thus making it a good choice for this task. We choose the first two sentences ($s_1, s_2$) as beginning rather than just $s_1$ because the first sentence ($s_1$) tends to be short in length, and usually introduces characters or sets the scene (Mostafazadeh et al., 2016), wherease the second sentence ($s_2$) provides more information about the initial story.

### 6.2 Hyperparameter Details

*Parameter size.* For GPT-2 we use the GPT-2 small checkpoint (117M parameters) based on the implementation of HuggingFace (Wolf et al., 2020).

*Decoding Strategy.* In the inference stage, we adopt beam search decoding with a beam size of 5 for all our models and all baselines we produce.

We used the following set of hyperparameters for our COINS model: batch size: $\{2, 4\}$; epochs: $\{3, 5\}$; learning rate: $\{1e\text{-}5, 5e\text{-}6\}$. We use Adam Optimizer, and dropout rate = 0.1. We ran our experiments with GPU sizes of 11GB and 24GB.

### 6.3 Baselines

We compare our COINS model to the following baselines:

(a) **GPT-2** (Radford et al., 2018) (with 12-layer, 768-hidden, 12-heads), trained with an objective to predict the next word. The input to the GPT-2 model is the concatenation of the source and the target story sequence. We follow the standard procedure to fine-tune GPT-2 on the NSC task during training and minimize the loss function:

$$-log(s_3, s_4|[SOS]s_1, s_2, [SEP], s_5[EOS]) \quad (4)$$

(b) **Knowledge-Enhanced GPT-2** (**KE**) (Guan et al., 2020) is the current SOTA for ROCStories generation. It first fine-tunes a pre-trained GPT-2 (small) model with knowledge triples from commonsense datasets (ConceptNet [CN] Speer et al. (2017) and ATOMIC [AT] Sap et al. (2020)). The knowledge triples were converted to sentences using templates. A multitask learning framework further fine-tunes this model on both the *Story Ending Generation task* and classifying corrupted stories from real ones. As our baseline we choose the version without multi-tasking, since the corrupted story setting is not applicable for the *NSC* task.

(c) **GRF** (Ji et al., 2020) is the current SOTA for the *Abductive Reasoning* and the *Story Ending Generation* tasks. GRF enables pre-trained models (GPT-2 small) with dynamic multi-hop reasoning on multi-relational paths extracted from the external ConceptNet commonsense knowledge graph.

(d) **GLUCOSE-GPT-2** Similar to Guan et al. (2020), we *fine-tune* pretrained GPT-2 (small) on the GLUCOSE dataset using *general rules* (GR). We follow the same procedure as Guan et al. (2020) and (i) first fine-tune a pre-trained GPT-2 , but here on the GLUCOSE dataset, with the following loss:

$$-log(I_i|S, s_i, r), \quad (5)$$

where r: CAUSE/EFFECT, $I_i$: Inference rules. (ii) Then we fine-tune the above model again on the NSC dataset with the following loss:

$$-log(s_3, s_4|[SOS]s_1, s_2, [SEP], s_5[EOS]) \quad (6)$$

The main difference between GLUCOSE-GPT-2 and COINS is: **COINS** explicitly learns to generate (contextualized) inference rules *on the fly* during the inference step and incorporates them in the story generation step.

### 6.4 Automatic Evaluation Metric

For automatic evaluation in the *NSC* task we use as metrics Perplexity (indicates fluency of text generation), BLEU-1/2 (Papineni et al., 2002) and ROUGE-L (Lin, 2004). We report performance on the test

[9] The results for *Story Ending Generation* will corroborate our results for *NSC*. All details are given in the *Appendix*.

| **Model** | **PPL** (↓) | **BLEU-1/2** (↑) | **ROUGE-L** (↑) |
|---|---|---|---|
| **GPT-2** | 11.56 | 16.66/6.8 | 17.2 |
| **KE** [CN, AT] | 12.61 | 17.55/7.6 | 17.9 |
| **GLUCOSE-GPT-2** | 12.7 | 17.9/7.8 | 17.5 |
| **GRF** [CN] | 12.18 | 20.8/8.2 | 17.6 |
| **COINS (SR)** | **6.7** | 22.53/10.10 | 18.9 |
| **COINS (GR)** | 6.9 | **22.82/10.52** | **19.4** |
| **COINS Oracle (SR)** (Test-only) | – | 30.75/22.76 | 32.5 |
| **COINS Oracle (GR)** (Test-only) | – | 26.37/17.01 | 27.38 |
| **Human** | – | 24.53/12.10 | 20.2 |

Table 4: Automatic evaluation results for Story Completion. Best performance highlighted in **bold**; used Inference Rule types: specific (SR), general (GR).

sets by averaging results obtained for 5 different seeds. All improvements across all model variants are statistically significant at $p < 0.05$).

## 7 Results

Our experimental results are summarised in Tables 4 and 6.

**NSC task.** Table 4 shows the results for the models described in §6.3 and evaluated as per §6.4. We observe the following: (i) COINS outperforms all strong baseline models that utilize pre-trained language models and incorporate external commonsense knowledge with respect to all automatic evaluation metrics. Note that **GLUCOSE-GPT2** and **COINS** are using the same knowledge resource, hence the clear performance increase of COINS (+4.92 BLEU score) indicates that jointly learning to generate contextualized inferences rules and missing sentences in a recursive manner can enhance generation quality.[10] (ii) Similar to Ji et al. (2020) we observe that fine-tuning GPT-2 over knowledge triples ([CN], [AT]OMIC or [GL]UCOSE) doesn't improve the overall performance by much (Table 4, line 2: [CN+AT] vs. line 3: [GL] vs. line 1: [no CSK]). (iii) For COINS, *general rules* (GR) boost performance more than specific rules, indicating that the sentence generation model generalizes well. (iv) In the oracle settings at inference time we provide the model with the silver inference rules (generated as per §4) that use the complete story context as background. The result indicates that SR performs better than GR when the model sees the full story context.

| **Input** | **PPL** (↓) | **BLEU-1/2** (↑) | **ROUGE-L** (↑) |
|---|---|---|---|
| **IR only** (GR) | 13.05 | 10.65/4.01 | 6.31 |
| **IR only** (SR) | 8.01 | 15.65/6.08 | 15.31 |
| **No IR + w/oSE** | 11.5 | 15.12/5.95 | 12.47 |
| **IR (GR) + w/oSE** | 7.49 | 21.50/9.78 | 18.07 |

Table 5: Impact of different inputs to COINS for Story Completion, SR: specific rules, GR: general rules, IR: inference rules, **w/oSE**: w/o the story ending ($s_n$).

In general we observe that story generation benefits from higher-quality, contextualized inference rules from GLUCOSE (for COINS).[11] The improvement of COINS over GLUCOSE-GPT-2 indicates that our model is well able to utilize and profit from the inference rules. In the oracle setting, SR performs much better than GR. This is expected, since oracle rules with access to the full context will deliver more contextually-relevant inferences, while GR rules may diverge more from the story context. However, in the realistic NSC task setting (Table 4, lines 5,6) GR outperforms SR, which again underlines the generalization capacities of COINS.

**Impact of different inputs for the Generation Step.** In Table 5 we investigate the performance of COINS with different inputs to the sentence generation component *at inference time*: (i) When only inference rules (from the inference step) are given to the model without any story context ($S'$ = $\{s_1, s_2$,[SEP], $s_n\}$) (**IR only**), sentence generation benefits when specific rules are used. This is expected since the specific rules contain statements with concrete character names and paraphrased events from the story. (ii) When only the story beginning ($s_{1,2}$) is provided to the sentence generation model *without* the ending sentence $s_n$ (**w/oSE**) nor inference rules (**w/oIR**) we observe that the performance drops compared to models given the full incomplete context ($S'$), indicating that knowing the story ending helps the model to generate missing sentences that are coherent with the story. However, (iii) when adding inference rules **IR** (from the inference step i.e., E$_i$ + C$_i$) to the context ($s_{1,2}$) without ending sentence (**w/oSE**), performance again improves (+5.85 BLEU scores). Note that the inference rule contains the CAUSE relation for $s_n$. This indicates that the model is able to utilize inference rules for story generation.[12]

[10] Since **GRF**'s architecture is specific for ConceptNet, we cannot exclude that the better performance of COINS (+2.2 BLEU) is in part due to differences in the used knowledge.

[11] Automatic (silver) GLUCOSE inference rules (cf. §4) of type GR yield 60.8 BLEU score i.e., performance of $CSI_{gen}$ (avg. of both relation types).

[12] Here, we report the results with generalized rules as GR works better than SR when context is given (cf. Table. 4).

|  | Full Context |  | 1-Missing Sentence |  | 2-Missing Sentence |  |
|---|---|---|---|---|---|---|
| **Model** | **E** | **C** | **E** | **C** | **E** | **C** |
| **GPT-2**$^{\dagger}$ | 58.3 | **63.3** | 56.5 | 58.3 | 55.4 | 53.9 |
| **COINS** | **59.9** | 62.9 | **58.6** | **60.3** | **57.5** | **56.8** |
| **GPT-2**$^{\dagger}$ | 57.7 | 59.5 | 55.5 | 55.3 | 53.4 | 51.4 |
| **COINS** | 57.8 | **60.1** | **56.3** | **58.2** | **55.1** | **55.2** |

Table 6: Automatic evaluation of the quality of inference rules in different context settings. Best results in **bold**. Metric: BLEU-1 scores, **E**: EFFECT, **C**: CAUSE, Grey: context-specific rules (SR); regular: general rules (GR), $^{\dagger}$: *fine-tuned* on GLUCOSE dataset.

**Performance of inference rule generation.** We now investigate how difficult it is to generate contextualized inference rules (specific and general) when multiple sentences are missing from a story. For this we compare COINS to a GPT-2 model fine-tuned on GLUCOSE data to generate inference rules (cf. §4). We study the impact of jointly and dynamically learning sentence and inference rule generation (in COINS) on the inference generation task – while the fine-tuned GPT-2 model only learns to generate inference rules conditioned on the static story context. We specifically examine the difficulty of generating inference rules *for two consecutive sentences* ($s_3$ and $s_4$) in a 5-sentence context, as opposed to shorter sequences, in three different scenarios: i) when the *complete story context* $S$ is given; ii) when *the incomplete context* $S'$ (i.e., $s_1, s_2$ and $s_5$) is given, plus either $s_3$ or $s_4$ (**1-missing sentence**), and iii) when $S'$ is given, but neither of the intermediate sentences $s_3$ and $s_4$ (**2-missing sentences**). In each setting, we generate EFFECT and CAUSE rules for the targeted sentences $s_3$, $s_4$, and compare their quality. The results are reported in Table 6. We observe that in the **2-missing sentences** setting, COINS outperforms GPT-2 (by $+2.3$ BLEU score on average). This indicates that learning to perform inference rule generation jointly with sentence generation is beneficial for filling-in multiple story sentences. Interestingly, for increasing numbers of missing sentences, performance drops drastically for CAUSE (as opposed to EFFECT), but less so for COINS as opposed to GPT-2. A possible reason for this may be the conditional, uni-directional nature of the underlying GPT-2 language model, which is trained to predict follow-up words in forward direction. This may favor future-directed EFFECT rules – as opposed to CAUSE relations. The milder effect on COINS could indicate that the concurrent inference model supports the sentence generation model to overcome this weakness.[13]

## 8 Manual Evaluation

Automatic metrics can give us some indication of NLG quality, however, these metrics do not necessarily reflect the coherence of generated story sentences. We thus conduct a human evaluation focusing on the grammaticality and coherence of the generated sentences in their story context. We conduct pairwise comparisons for randomly sampled 100 instances of our best model, i.e., COINS with GR (according to automatic metrics) with four strong baseline models (GPT-2, GLUCOSE-GPT-2, GRF, KE). For each pair of instances (one from COINS, the other from a baseline model), we present the generated sentences in their story context, and asked three annotators to give a *preference rating* (*win*, *tie*, *lose*) according to the criteria *grammaticality* and *coherence*. For grammaticality, we present each sentence in isolation and ask the annotators to rate which sentence is more fluent, readable, and compliant with the English standard usage. For coherence, we ask the annotators to assess which of the two generated sentences are more logically coherent with each other and the story beginning and ending, in terms of causal and temporal dependencies. We applied majority voting among the three annotators to obtain final decisions. More details about the annotation are given in *Appendix*.

The human evaluation results are presented in Table 7.[14] The results show that our model produces more coherent and more grammatically correct sentences compared to all baselines. This indicates that with support of learned contextualized inference rules based on GLUCOSE knowledge, our model generates more coherent story sentences that are causally and temporally well connected.

**Relevance of Generated Inferences Rules.** We further conduct human evaluation to validate the effectiveness and relevance of the generated inference rules. We randomly select 50 instances from the NSC dev set. We asked three annotators to evaluate the (GR) inference rules[15]. We define an inference rule to be relevant if (a) it captures im-

[13] In future work, we will test the above hypothesis by experimenting with a bi-directional transformer generation model.

[14] We report inter-annotator agreement scores calculated with Fless' kappa $\kappa$ (Fleiss, 1971), calculated for each comparison. We find moderate or fair agreement.

[15] We report only COINS (GR), our best model according to automatic metrics.

| Models | Knowledge of Base Model | Coherence Win(%) | Tie(%) | Loss(%) | $\kappa$ | Grammaticality Win(%) | Tie(%) | Loss(%) | $\kappa$ |
|---|---|---|---|---|---|---|---|---|---|
| **COINS** vs **GPT-2** | None | 54.7 | 32.0 | 13.3 | 0.52 | 45.7 | 41.3 | 13.0 | 0.49 |
| **COINS** vs **GLUC.-GPT-2** | GLUCOSE | 52.0 | 33.0 | 15.0 | 0.43 | 31.7 | 54.3 | 14.0 | 0.45 |
| **COINS** vs **KE** | CN + ATOMIC | 50.0 | 32.0 | 18.0 | 0.44 | 21.3 | 69.7 | 9.0 | 0.37 |
| **COINS** vs **GRF** | CN | 50.5 | 30.5 | 19.0 | 0.48 | 20.5 | 70.0 | 9.5 | 0.35 |

Table 7: Manual evaluation of sentence generation quality of COINS (GR) for 100 stories. Scores are percentages of *Win, Loss,* or *Tie* when comparing COINS to baselines. Fleiss' kappa $\kappa$: fair agreement or moderate agreement.

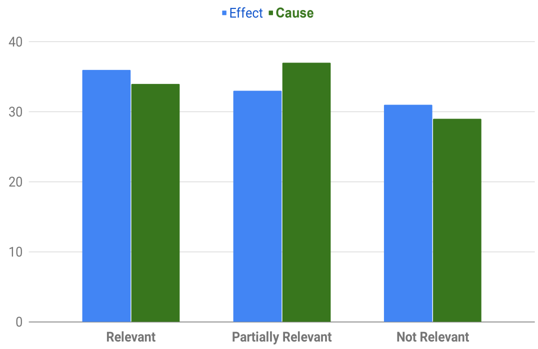


Figure 3: Human evaluation of the relevance of Inference Rules generated by COINS.

plicit causes and effects of a selected sentence $X$ given an incomplete story $S'$, and (b) it is providing useful explanations for the incomplete story $S'$. The result for this evaluation is shown in Fig.3, for EFFECT and CAUSE relations. We find that in 36% and 34% of cases for effects and causes, respectively (computed on the basis of majority agreement), our algorithm was able to generate relevant inference rules. Our annotations yielded fair inter-annotator agreement of Fleiss' $\kappa = 0.45$.

**Case Study.** We provide an example from NSC with different generation outputs (Table 8). Note that the generated sentences are grounded to the inference rules obtained from the inference step. Hence, the rules provide both an intermediate representation and explanations for how knowledge can guide or influence story generation. We provide more qualitative examples in the Appendix.

| | |
|---|---|
| Incomplete Story: | $s_1$: Ken was driving around in the snow. $s_2$: He needed to get home from work. $s_5$: His tires lost traction and he hit a tree. |
| Missing Sentences: | $s_3$: He was driving slowly to avoid accidents. $s_4$: Unfortunately the roads were too slick and Ken lost control. |
| COINS (I$_{GR}$) | Someone$_A$ is going Somewhere$_B$ $\succ$Cause/Enables$\succ$ Someone$_A$ is at Somewhere$_B$, Someone$_A$ **is driving Something$_A$ fast** $\succ$Cause/Enables$\succ$ Something$_A$ hits Something$_B$ (that is a tree), Someone$_A$ possess(es) Something$_A$ (that is **a car** ) $\succ$Enables$\succ$> Something$_A$ (tires) lost Something$_B$ (traction) |
| COINS (I$_{SR}$) | He posses(es) a car $\succ$result in$\succ$ His tires lost traction, He needed to get home $\succ$Enables$\succ$ He drove home, He was **driving on ice** $\succ$ Causes/Enables $\succ$ His tires lost traction, He was driving on ice $\succ$Causes/Enables$\succ$ He **lost control** of his vehicle. |
| COINS(MS$_{GR}$) | He was **driving too fast**. He lost control of his **car** . |
| COINS(MS$_{SR}$) | He was **driving on ice**. He **lost control of his vehicle** . |
| GPT-2 | He stopped at a gas station. He filled his tank. |
| GPT-2 GLUCOSE | When he got to the house he realized he was stuck. Ken had to pull over to get help. |
| KE | When he got home, he noticed his tires were flat. He decided to pull over. |
| GRF | He pulled over to see what was wrong. He saw that his car was stuck in the snow. |
| Human | He was going very fast. The street was slippery from the snow. |

Table 8: Examples: inference rules and missing sentences generated by COINS (compared to *Gold* from GLUCOSE, Green), as well as baseline model generations. Gray: COINS (SR); Regular: COINS (GR); MS: missing sentences, I: inference rules

## 9 Conclusion

We addressed a Narrative Story Completion task that allows us to probe the coherence capabilities of a neural generation model. We proposed COINS, a model that iteratively generates commonsense inference rules grounded in the context and generates story sentences, using the generated inferences as a guide. Human and automatic evaluations show that the model outperforms strong commonsense knowledge-based generation models. By individuating the inference rule and sentence generation steps, COINS can make the contribution of commonsense knowledge on story generation transparent. The recursive nature of the inference-driven generation model holds potential for knowledge-driven control in the generation of longer sequences. In future work we will explore how an enhanced memory of generated inferences can realize more complex narrative patterns that diverge from strictly ordered narrative sequences.

## Acknowledgements

This work has been supported by the German Research Foundation as part of the Research Training Group "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES) under grant No. GRK 1994/1. We thank our annotators for their valuable annotations. We also thank NVIDIA Corporation for donating GPUs used in this research.

## References

Prithviraj Ammanabrolu, Wesley Cheung, William Broniec, and Mark O. Riedl. 2021. Automated storytelling via causal, commonsense plot ordering. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Niranjan Balasubramanian, Stephen Soderland, Mausam, and Oren Etzioni. 2013. Generating coherent event schemas at scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1721–1731, Seattle, Washington, USA. Association for Computational Linguistics.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

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

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610, Suntec, Singapore. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

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

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

Jian Guan, Yansen Wang, and Minlie Huang. 2019. Story ending generation with incremental encoding and commonsense knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6473–6480.

Jerry R Hobbs. 1985. On the coherence and structure of discourse.

Parag Jain, Priyanka Agrawal, A. Mishra, M. Sukhwani, Anirban Laha, and K. Sankaranarayanan. 2017. Story generation from sequence of independent short descriptions. abs/1707.05501.

Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. 2020. Language generation with multi-hop reasoning on commonsense knowledge graph. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 725–736, Online. Association for Computational Linguistics.

R. Józefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Y. Wu. 2016. Exploring the limits of language modeling. *ArXiv*, abs/1602.02410.

W. Kintsch and T. A. Dijk. 1978. Toward a model of text comprehension and production. *Psychological Review*, 85:363–394.

Michael Lebowitz. 1987. Planning stories. In *Proceedings of the 9th annual conference of the cognitive science society*, pages 234–242.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2020. Commonsense knowledge base completion with structural and semantic context. *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.

Raymond J Mooney and Gerald DeJong. 1985. Learning schemata for natural language processing. In *IJCAI*, pages 681–687.

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.

Kiem-Hieu Nguyen, Xavier Tannier, Olivier Ferret, and Romaric Besançon. 2015. Generative event schema induction with entity disambiguation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 188–197.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Debjit Paul and Anette Frank. 2020. Social commonsense reasoning with multi-head knowledge attention. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2969–2980, Online. Association for Computational Linguistics.

Debjit Paul and Anette Frank. 2021. Generating hypothetical events for abductive inference. In *Proceedings of the Tenth Joint Conference on Lexical and Computational Semantics*, Online. Association for Computational Linguistics.

Karl Pichotta and Raymond Mooney. 2014. Statistical script learning with multi-argument events. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 220–229.

Rafael PÉrez Ý PÉrez and Mike Sharples. 2001. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence*, 13(2):119–139.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D. Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 794–805, Online. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Hannah Rashkin, Antoine Bosselut, Maarten Sap, Kevin Knight, and Yejin Choi. 2018a. Modeling naive psychology of characters in simple commonsense stories. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2289–2299.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A. Smith, and Yejin Choi. 2018b. Event2Mind: Commonsense inference on events, intents, and reactions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 463–473, Melbourne, Australia. Association for Computational Linguistics.

Melissa Roemmele. 2016. Writing stories with help from recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

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. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019.*, pages 3027–3035.

Maarten Sap, Vered Shwartz, Antoine Bosselut, Yejin Choi, and Dan Roth. 2020. Commonsense reasoning for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 27–33, Online. Association for Computational Linguistics.

Roger C. Schank and Robert P. Abelson. 1977. *Scripts, plans, goals, and understanding : an inquiry into human knowledge structures*. Hillsdale, N.J. : Lawrence Erlbaum Associates.

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*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *NAACL-HLT*.

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

Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro. 2020. MEGATRON-CNTRL: Controllable story generation with external knowledge using large-scale language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2831–2845, Online. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. 2020. Aser: A large-scale eventuality knowledge graph. In *Proceedings of The Web Conference 2020*, pages 201–211.

## A Supplementary

### A.1 Manual Evaluation.

We perform an error analysis to better understand the generation quality. We ask our annotators to assess whether the generated text contains any pieces of information that are contradicting the given incomplete story or not. Our annotations were performed by three annotators with a *linguistic* background. Figure 5, shows a screenshot of the annotation guidelines. Figure 4 depicts the result, we observe the that our COINS models produce less contradicting missing sentences compare to other baselines.

### A.2 Hyperparameter Details

**Parameter size.** For GPT-2 we use the GPT-2 small checkpoint (117M parameters) based on the implementation of HuggingFace (Wolf et al., 2020) at: `https://github.com/huggingface/transformers/tree/master/src/transformers/models/gpt2`

**Decoding Strategy.** In the inference stage, we adopt beam search decoding with a beam size of 5 for all our models and all baselines we produce. We used the following set of hyperparameters for our COINS model: batch size: $\{2, 4\}$; epochs: $\{3, 5\}$; learning rate: $\{1e\text{-}5, 5e\text{-}6\}$. We use Adam Optimizer, and dropout rate = 0.1. We ran our experiments with GPU sizes of 11GB and 24GB.

**Training Details.** Our training time is $\approx$24 hours. The original ROCStories Corpus can be found at: `https://cs.rochester.edu/nlp/rocstories/`

### A.3 Story Ending Generation Task

**Data.** This task is to generate a reasonable ending given a four-sentence story context (Guan et al., 2019). The stories are from ROCStories (Mostafazadeh et al., 2016). We use the same data splits as Guan et al. (2019).

**SEG task.** We also investigate how COINS performs when applied to the task of generating a story ending when given a 4-sentence story (SEG). In this task our model takes only one iteration step to generate the story ending, where in the inference step it generates EFFECT inference rules for sentence ($s_4$). As seen in Table 9, the COINS model outperforms all previous strong baselines, including GPT2-GLUCOSE that uses the same knowledge resource. Interestingly, we also observe that fine-tuning on GLUCOSE or ConceptNet knowledge improves the text generation diversity, indicating that the models leverage concepts and event knowledge during generation (cf. Table 9 line.4-8).

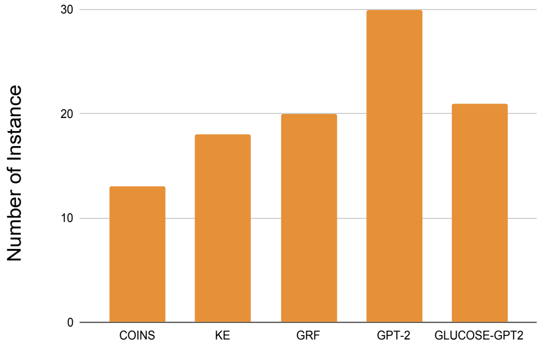


Figure 4: Human evaluation on Contradiction

| Model | BLEU-1/2 (↑) | Distinct-2/3 (↑) |
|---|---|---|
| **Seq2Seq**$^{\dagger}$ | 19.1 / 5.5 | 0.181 / 0.360 |
| **IE+GA**$^{\dagger}$ | 20.8 / 6.4 | 0.140 / 0.280 |
| **GPT**$^{\dagger}$ | 25.5 / 10.2 | 0.304 / 0.505 |
| **GPT2-OMCS**$^{\dagger}$ | 25.5 / 10.4 | 0.352 / 0.589 |
| **GPT2-GLUCOSE** | 25.6 / 10.2 | 0.361 / 0.609 |
| **GRF**$^{\dagger}$ | 26.1 / 11.0 | 0.378 / 0.622 |
| **COINS (GR)** | **27.4 / 12.3** | **0.428 / 0.724** |
| **COINS (Oracle)** | 41.80/28.40 | 0.479/0.786 |

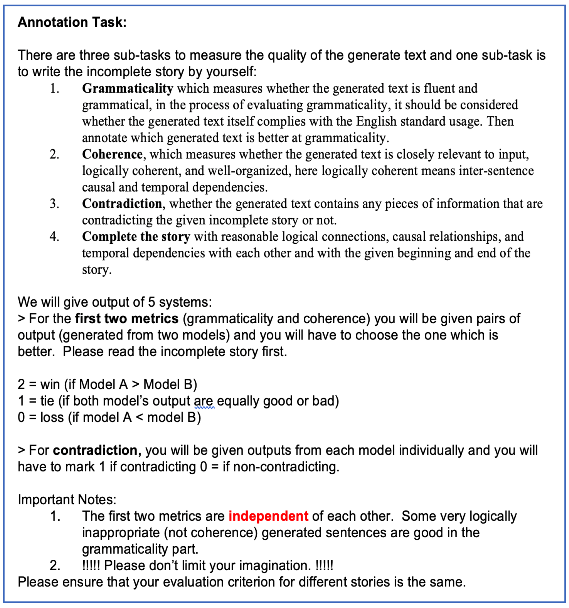
Table 9: Result: Automatic evaluation results on the Story Ending Generation Task, $^{\dagger}$ (Ji et al., 2020)

| Dataset | Train | Dev | Test |
|---|---|---|---|
| **SEG** | 90,000 | 4,080 | 4,081 |

Table 10: Dataset Statistics: nb. of unique stories

**Automatic Metrics.** For Story Ending Generation (SEG) we follow the metrics used in Guan et al. (2019); Ji et al. (2020): they use BLEU-1/2 to measure n-gram overlap between generated and human-written story endings, and Distinct-n (Li et al., 2016) to measure the generation diversity using maximum mutual information.

**Baselines.** For the *Story Ending Generation task*, we compare COINS to the **IE+GA** model (Guan et al., 2019). It is based on incremental encoding and multi-source graph attention (Guan et al.,



**Annotation Task:**

There are three sub-tasks to measure the quality of the generate text and one sub-task is to write the incomplete story by yourself:

1. **Grammaticality** which measures whether the generated text is fluent and grammatical, in the process of evaluating grammaticality, it should be considered whether the generated text itself complies with the English standard usage. Then annotate which generated text is better at grammaticality.
2. **Coherence**, which measures whether the generated text is closely relevant to input, logically coherent, and well-organized, here logically coherent means inter-sentence causal and temporal dependencies.
3. **Contradiction**, whether the generated text contains any pieces of information that are contradicting the given incomplete story or not.
4. **Complete the story** with reasonable logical connections, causal relationships, and temporal dependencies with each other and with the given beginning and end of the story.

We will give output of 5 systems:
> For the **first two metrics** (grammaticality and coherence) you will be given pairs of output (generated from two models) and you will have to choose the one which is better. Please read the incomplete story first.

2 = win (if Model A > Model B)
1 = tie (if both model's output are equally good or bad)
0 = loss (if model A < model B)

> For **contradiction,** you will be given outputs from each model individually and you will have to mark 1 if contradicting 0 = if non-contradicting.

Important Notes:

1. The first two metrics are **independent** of each other. Some very logically inappropriate (not coherence) generated sentences are good in the grammaticality part.
2. !!!!! Please don't limit your imagination. !!!!!

Please ensure that your evaluation criterion for different stories is the same.

Figure 5: A screenshot of the annotation guidelines for manual evaluation.

| | |
|---|---|
| Incomplete Story: | $s_1$: Danielle dreamed of living in California. $s_2$: After college she had to decide where to live. [mask] $s_5$: She loved it there. |
| Missing Sentences: | $s_3$: She could move back home or move to California. $s_4$: Danielle decided to take a leap and move to California. |
| COINS ($I_{GR}$) | Someone$_A$ decide Something$_A$ (where to live) >Causes/Enables> Someone$_A$ decides to live in Somewhere$_A$. |
| COINS ($I_{SR}$) | She had to decide where to live >Causes/Enables> She chose to **live in California**. |
| COINS($MS_{GR}$) | She decided to live in California. She settled in California. |
| COINS($MS_{SR}$) | She decided to **live in California**. She went to the beach. |
| GPT-2 | She finally settled in California. She loved it there. |
| GPT-2 GLUCOSE | She decided to move to NH. She found a nice apartment there. |
| KE | When he got home, he noticed his tires were flat. He decided to pull over. |
| GRF | She decided to move to California. She found a great place to live. |

Table 11: Example1: Generated Inference rules and Missing Sentences

2019). We also compare to a Seq2Seq model (Luong et al., 2015) based on gated recurrent units (GRU) and attention mechanism.

| | |
|---|---|
| Incomplete Story: | $s_1$: Her favorite glasses were ruined. $s_2$: The pink dye had gotten all over them. $s_5$: She chose pink, and they both laughed at the irony. |
| Missing Sentences: | $s_3$: Her mother took her to get a new prescription. $s_4$: It was time to order a new pair. |
| COINS($MS_{GR}$) | She took her friend to get a new one. She took it and it was pink. |
| GPT-2 | She bought a new pair of glasses. She wore them to school. |
| GPT-2 GLUCOSE | She couldn't decide between two colors. She finally decided on pink. |
| KE | She was sad that she couldn't see anymore. Her boyfriend came over to help. |
| GRF | She decided to dye them pink instead. She went to the store and bought a pink one. |

Table 12: Example2: Generated Missing Sentences

| | |
|---|---|
| Incomplete Story: | $s_1$: Susy was writing an essay by hand for class. $s_2$: She handed it in and thought she would do well. $s_5$: the teacher could not even grade it. |
| Missing Sentences: | $s_3$: But unfortunately the teacher could not even read it. $s_4$: Susy was humiliated. |
| COINS($MS_{GR}$) | But she could not. Teacher didn't read the essay. |
| GPT-2 | Suddenly, her hand slipped. She fell and broke her wrist. |
| GPT-2 GLUCOSE | But all the sudden she got an F. Susy was so embarrassed. |
| KE | When she got her paper back she realized she had tylenol. She had written the entire essay by hand. |
| GRF | Susy was very nervous about the essay. The teacher told her she was not allowed to write. |

Table 13: Example3: An example where all the models failed to generated coherent sentences

| | |
|---|---|
| Incomplete Story: | $s_1$: Seth was at a party with his friends. $s_2$: **Someone dared a kid to climb on a wall.** $s_5$: He immediately began screaming that his leg was broken. |
| Missing Sentences: | $s_3$: The kid climbed to the top and everyone cheered. $s_4$: Suddenly he slipped and fell to the ground. |
| *Gold*: | Some People$_A$ (who should not be there) start daring a Someone$_C$ to climb a Something$_C$ (without safety gear) >Causes/Enables> Someone$_C$ (who should not be there makes it to the top then falls down and Someone$_C$ (who is acting like monkey)). |
| | The kids start daring a kid to climb the wall >Causes/Enables> He makes it to the top then falls down and breaks his leg. |
| *Fine-tuned GPT-2*: | Some People$_B$ start daring a Someone$_A$ to climb a Something$_C$ >Causes/Enables> Someone$_A$ quickly shouted that his leg was broken. |
| | Someone start daring a kid to climb the wall >Causes/Enables> He shouted that his leg was broken. |
| *COINS*: | Some People$_B$ start daring a Someone$_A$ to climb a Something$_C$ >Causes/Enables> Someone$_A$ is on top of Somewhere$_A$ |
| | Someone start daring a kid to climb the wall >Causes/Enables> He climbed at the top. |

Table 14: Example of inference rules generated by COINS and *Fine-tuned* GPT-2 when **2-sentences** are missing (compared to *Gold* from GLUCOSE). Grey: context-specific rules (SR); regular: general rules (GR). Bolded sentence $s_2$ is $X$, EFFECT is the relation type $r$.