Counterfactual Story Reasoning and Generation

Lianhui Qin ♠◇   **Antoine Bosselut** ♠◇   **Ari Holtzman** ♠◇   **Chandra Bhagavatula** ◇
Elizabeth Clark ♠   **Yejin Choi** ♠◇

♠Paul G. Allen School of Computer Science & Engineering, University of Washington
◇Allen Institute for Artificial Intelligence
{lianhuiq,antoineb,ahai,eaclark7,yejin}@cs.washington.edu
chandrab@allenai.org

Abstract

Counterfactual reasoning requires predicting how alternative events, contrary to what actually happened, might have resulted in different outcomes. Despite being considered a necessary component of AI-complete systems, few resources have been developed for evaluating counterfactual reasoning in narratives.

In this paper, we propose *Counterfactual Story Rewriting*: given an original story and an intervening counterfactual event, the task is to minimally revise the story to make it compatible with the given counterfactual event. Solving this task will require deep understanding of causal narrative chains and counterfactual invariance, and integration of such story reasoning capabilities into conditional language generation models.

We present TIMETRAVEL, a new dataset of 29,849 counterfactual rewritings, each with the original story, a counterfactual event, and human-generated revision of the original story compatible with the counterfactual event. Additionally, we include 81,407 counterfactual "branches" without a rewritten storyline to support future work on semi- or un-supervised approaches to counterfactual story rewriting.

Finally, we evaluate the counterfactual rewriting capacities of several competitive baselines based on pretrained language models, and assess whether common overlap and model-based automatic metrics for text generation correlate well with human scores for counterfactual rewriting.

1 Introduction

A desired property of AI systems is counterfactual reasoning: the ability to predict causal changes in future events given a counterfactual condition
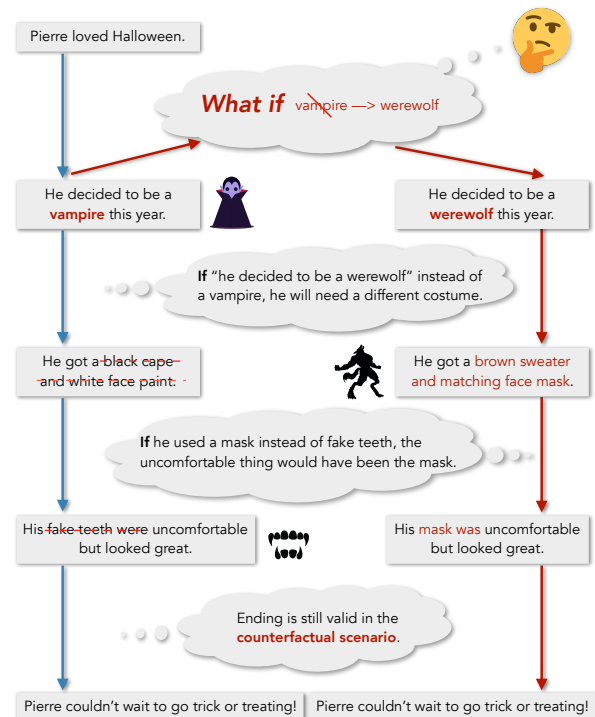

Figure 1: Given a short story (left column) and a *counterfactual context* ("He decided to be a werewolf this year"), the task is to revise the original story with minimal edits to be consistent with both the original premise ("Pierre loved Halloween") and the new counterfactual situation. The modified parts in the new story (right column) are highlighted in red.

applied to the original chain of events (Goodman, 1947; Bottou et al., 2013). For example, given an original story shown in the left chain in Figure 1, where "Pierre loved Halloween. He decided to be a *vampire* this year. He got a *black cape and white face paint*..." and a counterfactual condition, "what if Pierre decided to be a *werewolf* instead of a *vampire*?", an intelligent system should be able to revise the subsequent events in the story appropriately, for example, that a *brown sweater* would be more appropriate than a *black cape*.
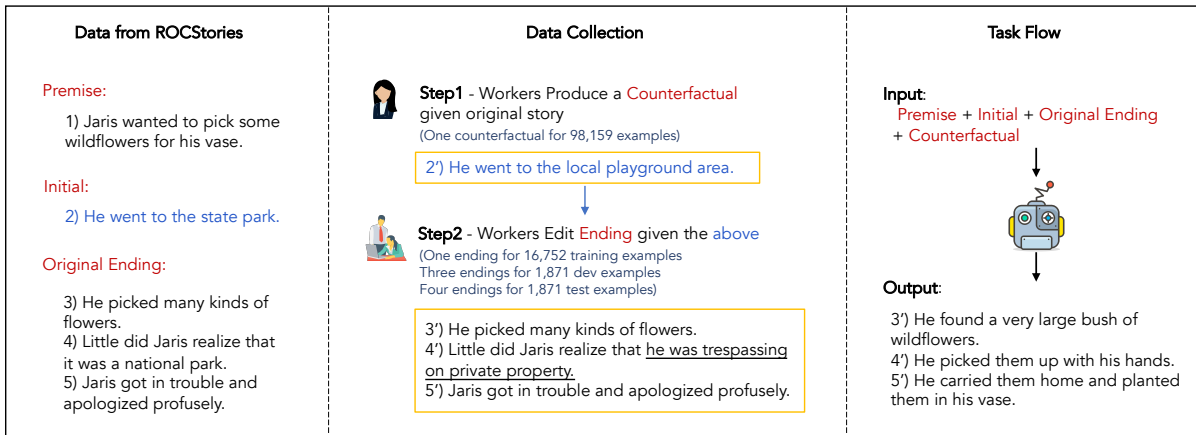
5043

Figure 2: Data annotation process for the TIMETRAVEL dataset. Given a story from the ROCStories corpus, crowdworkers write a counterfactual sentence w.r.t the second sentence of the story. The counterfactual sentence and the original story are then presented to other workers to rewrite the story ending. Models for the task are expected to generate a rewritten ending given the original story and counterfactual sentence.

This notion of counterfactuals has become increasingly relevant in several recent benchmarks such as ROC story cloze (Mostafazadeh et al., 2016), COPA (Roemmele et al., 2011), and HellaSwag (Zellers et al., 2019), where the negative responses in multiple-choice problems implicitly construct counterfactual narratives. However, no existing benchmark to date has been designed to explicitly evaluate counterfactual narrative reasoning and revision as its principal focus, where a system is evaluated on its ability to make modifications to future events based on a counterfactual condition, as illustrated in Figure 1.

In this paper, we introduce *Counterfactual Story Rewriting* as a new challenge to story understanding and generation. Given an original story and a counterfactual condition, the task is to re-write the story to regain narrative consistency through counterfactual reasoning. An important challenge in counterfactual reasoning is *causal invariance*, namely, the aspects of future events that are invariant under the counterfactual conditions. This is necessary to accurately reason about the new consequences with minimal edits to the original sequence of events, instead of being confounded by spurious correlations (Woodward, 2002; Bottou, 2019). Therefore, a key measure of the task besides consistency is that the rewriting must perform *minimal edits* to the original story. This challenges the system to reason about causal invariance, which in turn, challenges the system to reason more carefully about the causal chains of how the story unfolds.

We introduce TIMETRAVEL, a new dataset with 29,849 counterfactual revisions to support research on counterfactual narrative reasoning and revision. In addition, our dataset provides 81,407 counterfactual *branches* without rewritten storylines to support potential future work on semi- or un-supervised approaches. Figure 2 illustrates (1) the structure of the original stories, (2) the counterfactual data construction process, and (3) the final task definition.

We establish baseline performances of state-of-the-art neural language models on this task, such as GPT (Radford et al., 2018) and GPT-2 (Radford et al., 2019), evaluated in zero-shot, unsupervised, and supervised learning settings. Empirical results indicate that while these models are able to capture certain instances of counterfactual reasoning, they generally struggle with rewriting endings with full consistency. Our results suggest that current neural language models operate based primarily on frequent patterns in language without true understanding of the causal chains in narratives, thus requiring more focused future research to integrate reasoning capabilities to neural language models.

2 Background

Counterfactual reasoning is the ability to consider *alternative possibilities* that diverge from current observed narratives. Due to their prevalence in common reasoning situations, counterfactuals have been studied in a wide range of disciplines, including psychology (Epstude and Roese, 2008),

cognitive science (Byrne, 2002), as well as natural language processing (Hobbs, 2005; Lawrence and Riezler, 2018; Son et al., 2017).

Meanwhile, despite the progress made in NLU tasks by adapting pretrained language representations such as BERT (Devlin et al., 2018) or GPT (Radford et al., 2018), models still have trouble discriminating between reasonable and unreasonable counterfactuals, as shown in (Zellers et al., 2019). Moreover, success in tasks linked to discrimination of reasonable alternatives often results in models learning to exploit latent artifacts of the dataset (Niven and Kao, 2019; Zellers et al., 2018), rather than learning to robustly reason about counterfactuals. In response to this, we hypothesize that learning to *generate the result* of counterfactual prompts will encourage models to learn to understand the underlying dynamics of a given situation, whereas discrimination between two alternatives is more likely to take advantage of dataset biases.

This goal shares many similarities with script learning (Pichotta and Mooney, 2014; Chambers, 2013), which attempts to canonicalize stereotypical event sequences for learning causal structure of narratives. However, because it is often difficult to capture the richness of causal dependencies with templatized structures (Sap et al., 2019), we instead study counterfactual reasoning in unstructured text directly and also require the model to *generate* the consequences of the counterfactual reasoning.

The "counterfactual event" in our task can be viewed as a causal intervention (Pearl, 2000) in the latent chain of events of the story. Such interventions demand changes to the written narrative in order to abide by the shared background knowledge that human readers have about how the world works. This neatly embeds the problem of causal reasoning in a space that laymen with no knowledge of formalized causality can understand. It also allows us to evaluate the capabilities and limitations of the recent advances in neural language models in the context of counterfactual reasoning.

Similar issues arise in the area of controllable language generation (e.g., Hu et al., 2017), which involves preserving the content of text while changing it along a single or multiple dimensions, such as theme (Koncel-Kedziorski et al., 2016), style (Lample et al., 2019), and sentiment (Shen et al., 2017). Reasoning in these tasks is limited to discrete axes (e.g., sentiment), which are often categorized with a closed label set ({positive, negative}). Because of controllability motivations, these axes and labels are generally known *a priori*. In contrast, counterfactual rewriting focuses on the causes and effects of a story, dimensions that can require more complex and diverse, yet potentially subtle, changes to accommodate the counterfactual event. Additionally, we put no restrictions on the nature of counterfactual events, yielding no clear set of discrete axes along which the story can change and no closed set of labels for them.

3 Counterfactual Story Rewriting

3.1 Task

We now formally introduce the task and establish the notation used in the paper. Each example consists of a five-sentence story $S = (s_1, \ldots, s_5)$ with a general structure where the first sentence s_1 sets up the *premise*, the second sentence s_2 provides more information of the *initial context*, and the last three sentences $s_{3:5}$ are the *original ending* of story. We are further given an additional sentence s_2', which is counterfactual to the initial context s_2. That is, s_2' states something contrary to that in s_2, which in turn can make the original ending $s_{3:5}$ no longer valid. Thus, the goal of the task is to rewrite the ending, such that the *edited ending* $s_{3:5}'$ minimally modifies the original one and regains narrative coherency to the new counterfactual context.

The minimum edit goal differentiates our task from previous story ending studies, which have mostly focused on consistency in a given context. To achieve consistency with minimal edits, a model must understand the key mechanisms that drive the story's narrative so that it can filter out spurious correlations and capture counterfactual invariance. We thus consider the new task as a suitable testbed for studying counterfactual reasoning in combination with language generation.

3.2 Dataset: TIMETRAVEL

Our dataset is built on top of the ROCStories corpus (Mostafazadeh et al., 2016), which contains 98,159 five-sentences stories in the training set, along with 3,742 stories in the evaluation sets. Each story was written by

Premise	Alec's daughter wanted more blocks to play with.
Initial	Alec figured that blocks would develop her scientific mind.
Original Ending	Alec bought blocks with letters on them. Alec's daughter made words with them rather than structures. Alec was happy to see his daughter developing her verbal ability.
Counterfactual	Alec couldn't afford to buy new blocks for his daughter.
Edited Ending	Alec decided to make blocks with letters on them instead. Alec's daughter made words with the blocks. Alec was happy to see his daughter developing her verbal ability.
Premise	Ana had just had a baby girl.
Initial	She wanted her girl to have pierced ears.
Original Ending	She took her baby to the studio and had her ears pierced. Then she fastened tiny diamond studs into the piercings. Ana loved the earrings.
Counterfactual	She didn't like the idea of having her ears pierced.
Edited Ending	She decided not to take her baby to the studio to get her ears pierced. So she took tiny diamond stickers and stuck them to her ear. Ana loved the fake earrings.

Table 1: Examples from TIMETRAVEL

crowdworkers. To collect counterfactual events and new story continuations for TIMETRAVEL, we employ workers from Amazon Mechanical Turk (AMT) for a two-step task, which we describe in detail below.

3.3 Data Collection

Counterfactual Event Collection We present workers with an original five-sentence story $S = (s_1, s_2, \ldots, s_5)$ and ask them to produce a counterfactual event s_2' based on s_2. Workers are instructed to produce counterfactual sentences s_2' that are:
(1) Topically related to the original context sentence s_2, rather than a completely new sentence.
(2) Relevant to the original premise sentence, s_1, allowing for a coherent story continuation.
(3) Influential to the subsequent storyline, such that at least one of the original ending's sentences, $\{s_3, s_4, s_5\}$ is no longer appropriate given s_1 and s_2', necessitating a rewritten story ending.

Continuation Rewriting Once a counterfactual sentence s_2' is provided, we present it to a new set of workers with the original story $S = (s_1, s_2, \ldots, s_5)$. Now that s_2' invalidates the original storyline, workers are instructed to make minimal edits to $s_{3:5}$, such that the narrative is coherent again. Before beginning, workers are instructed to validate whether the counterfactual event satisfies the requirements from the previous stage of the pipeline. If not, we ask them to rewrite the counterfactual again, and the continuation rewriting step is reassigned to a new worker.

Summary We summarize the scale of TIMETRAVEL in Table 2. Overall, we collect

	Train	Valid	Test
ROCStories data:			
# Stories	98,159	1,871	1,871
TIMETRAVEL:			
# Counterfactual Context	98,159	5,613	7,484
# Edited Ending	16,752	5,613	7,484

Table 2: Dataset statistics

16,752 training examples of a counterfactual context and a rewritten ending. We also collect an additional 81,407 counterfactual contexts for the training set with no rewritten ending to support future work in unsupervised learning on this task. For the development and test sets, we gather multiple counterfactual contexts and rewritten endings for *each* example (3 new endings for development and 4 for test). Information regarding quality control and cost are provided in Appendix A.

4 Learning a Counterfactual Rewriter

Recent work in constructing large-scale generative language models based on transformers (Radford et al., 2018, 2019) has led to considerable improvements in natural language generation tasks. Due to their current prominence, we use them as baselines to study the extent to which the current neural text generation systems can perform and fail counterfactual narrative reasoning and revision. We focus on the family of GPT models, including GPT (Radford et al., 2018) and the latest small- (GPT2-S) and medium-sized (GPT2-M) transformer models from Radford et al. (2019). For each of the three pretrained language models, we fine-tune with multiple objectives, leading to 14 different model variants for the task, which we

describe in more detail below.

4.1 Unsupervised Training

Constructing large-scale counterfactual revision dataset is costly. Therefore, an ideal system must learn to reason without direct supervision. Toward this goal, we examine how unsupervised approaches to counterfactual story rewriting perform on our evaluation task. We devise the following unsupervised settings for models to learn to generate counterfactual story endings.

Zero-shot (ZS) In our simplest setting, we evaluate the counterfactual reasoning abilities already learned by these models due to pretraining on large corpora: the BooksCorpus dataset (Zhu et al., 2015) for GPT and the WebText corpus for GPT-2 (Radford et al., 2019). In this setting, models are not trained on any portion of the training data from TIMETRAVEL and must instead produce counterfactual rewritten stories for the evaluation set using only the representations learned from pretraining. At test time, the model receives the premise and the counterfactual context (s_1, s_2') as input and generates the tokens that constitute the rewritten counterfactual outcome.

Fine-tuning (FT) Because the domains on which both the GPT and GPT2 models were trained are broad and more complex than the domain of ROCStories, we investigate whether adapting the language model to the data distribution of ROCStories is helpful for learning to reason about counterfactuals. In this setting, the model is further fine-tuned to maximize the log-likelihood of the stories in the ROCStories corpus:

$$\mathcal{L}^{ft}(\boldsymbol{\theta}) = \log p_\theta(S), \qquad (1)$$

where p_θ is the language model with parameters $\boldsymbol{\theta}$, and S is the original story as defined in Section 3.1. This fine-tuning step encourages the model to generate text with the same consistent style of the stories. Similar to the zero-shot setting, the premise and the counterfactual sentence (s_1, s_2') are provided as input to the model.

Fine-tuning + Counterfactual (FT + CF) The above training loss, however, does not make use of the additional 81,407 counterfactual training sentences for fine-tuning. To inform the model

with a larger set of possible counterfactual narratives in the training data, we propose an additional loss function that fits the model to the counterfactual sentences given the premise sentence:

$$\mathcal{L}^{cf}(\boldsymbol{\theta}) = \log p_\theta(s_2'|s_1), \qquad (2)$$

where $p_\theta(s_2'|s_1)$ denotes that the language model first reads the premise s_1 and maximizes the log-likelihood of counterfactual sentence s_2'. The model is fine-tuned with both objectives in Eqs (1) and (2):

$$\mathcal{L}^{ft+cf}(\boldsymbol{\theta}) = \mathcal{L}^{ft} + \mathcal{L}^{cf}, \qquad (3)$$

and receives inputs in the same format as the zero-shot and fine-tuned models at test time.

Reconstruction + Counterfactual (RC + CF) One issue with the above training procedures is that models are not explicitly trained to retain as much text of the original outcome $x_{3:5}$ as possible (i.e., minimum edits). If these models are to learn to "rewrite" the original story ending given the counterfactual sentence, rather than learning to produce a completely new plot, they must be able to condition on the original ending during generation. Motivated by this requirement and following the goal of developing unsupervised methods for counterfactual rewriting, we design a reconstruction objective for learning a noisy reproduction of the original ending. Specifically, we provide the model with both the original story and a masked context as input $(S, [s], s_1, [mask])$ and train the model to reconstruct the original ending $s_{3:5}$:

$$\mathcal{L}^{rc}(\boldsymbol{\theta}) = \log p_\theta(s_{3:5}|S, [s], s_1, [mask]), \qquad (4)$$

where $[s]$ denotes a separator token and $[mask]$ is a special mask token. In this setting, the model first reads the original story S followed by the separation token $[s]$, and then reads the premise x_1 again, followed by the mask token $[mask]$, which serves as a placeholder sentence for the counterfactual sentence. This objective encourages the model to reproduce the original ending $s_{3:5}$ in the general case where the second sentence is not specified, thereby encouraging generations similar to the original ending regardless of the counterfactual provided. At test time, we replace $[mask]$ in the input

Premise	Rob loved to play soccer and was on his middle school team.
Initial	They had just started their season and were playing their first game.
Original Ending	Rob accidentally tripped over another player during the game. The other player was mad, but Rob apologized and kept his composure. Rob ended up playing the best game of his life!
Counterfactual	When he graduated from middle school, he started learning to play the guitar.

Model	Generated Ending
GPT-M + ZS	He played in bands for about a year, then got into touring with a band called The Muffs. He wrote the music for a few of The Muffs' songs. He was a very talented singer-songwriter and guitarist.
GPT-M + FT + CF	He took classes on the guitar for years to get better. He took a solo lesson and had a great time. Rob now is a famous guitarist.
GPT-M + Supervised	He accidentally tripped over the guitar during the song. The other person was mad, but Rob apologized and kept his composure. Rob ended up playing the best song of his life.

Premise	I called Carly to ask her on a date.
Initial	I got her voicemail.
Original Ending	I texted her that we should talk soon. She texted me back, asking why we should talk soon. I told her I liked her.
Counterfactual	She said that we should just stay friends which made me really sad.

Model	Generated Ending
GPT-M + ZS	She said that she went out and did something really stupid and that was all I could think about. I did my best to be nice to Carly by telling her how much I loved her and that I would be her best friend no matter what. There were nights when I felt like crying because I couldn't get out of bed.
GPT-M + FT + CF	I decided to watch a movie instead. After that I decided I should see a movie I had never seen before. I went to the theater and bought a ticket.
GPT-M + Sup	I texted her that we should just stay friends soon. She texted me back, asking why we should just stay friends soon. I told her I liked her.

Table 3: Model Outputs

with the counterfactual sentence s_2', and the model must generate the continuation of $(S, [s], s_1, s_2')$. We also use the objective from Eq (2) above to inform the model with counterfactual information during training.

4.2 Supervised Training (Sup)

Our dataset also provides 16,752 training instances that include human annotated rewritten endings for supervised learning. To assess whether being able to train directly on alternative endings is helpful for learning counterfactual narrative understanding, we train models on this portion of data in a supervised manner. More concretely, the input to the model contains the full information $(S, [s], s_1, s_2')$, and we train the model to maximize the log-likelihood of ground-truth rewritten endings:

$$\mathcal{L}^s(\boldsymbol{\theta}) = \log p_\theta(s_{3:5}'|S, [s], s_1, s_2'). \quad (5)$$

where [s] denotes a separator token.

4.3 Hyperparameters

We largely follow the same training and inference setups as in Radford et al. (2018) for the GPT model and Radford et al. (2019) for the GPT2 variants. Experiments are implemented with the

Model	Pre (1)	Plot (2)	CF (3)
GPT + ZS	1.945	1.290	1.555
GPT2-S + ZS	1.945	1.335	1.475
GPT2-M + ZS	2.435	1.615	2.045
GPT + FT	2.485	1.750	2.005
GPT2-S + FT	2.365	1.645	1.895
GPT2-M + FT	2.580	1.790	**2.070**
GPT + FT + CF	2.310	1.595	1.925
GPT2-S + FT + CF	2.310	1.640	1.850
GPT2-M + FT + CF	2.395	1.650	1.945
GPT2-S + RC + CF	2.240	2.090	1.500
GPT2-M + RC + CF	**2.780**	2.595	1.660
GPT + Sup	2.630	**2.690**	1.460
GPT2-S + Sup	2.705	2.650	1.625
GPT2-M + Sup	2.750	2.620	1.820
Human	2.830	2.545	2.520

Table 4: Likert scale scores for different models. The top performing model for each question is **bolded**.

text generation toolkit Texar (Hu et al., 2019). We provide more details in Appendix B.

5 Human Study of Rewritten Sentences

To assess the quality of rewritten endings, we conduct two sets of human evaluation. To give a sense of the model generation, Table 3 presents example outputs by a subset of representative models on two test cases.

5.1 Rewritten Sentence Scoring

Setup In this setting, workers from Amazon Mechanical Turk were presented 100 outputs from 14 different models. For each example, two workers were presented the original premise sentence, the original ending, the counterfactual sentence, and the rewritten ending, and asked to answer the following three questions on a 3-point Likert scale:

(1) Does the rewritten ending keep in mind details of the original premise sentence?

(2) Is the plot of the rewritten ending relevant to the plot of the original ending?

(3) Does the rewritten ending respect the changes induced by the counterfactual sentence?

In addition to evaluating the 14 models, we also provided gold human annotated counterfactual endings for the same 100 test examples to compute an expected upper bound for how models should perform. We present the results from this study in Table 4 and share key observations below.

Model Size and Pretraining Data We observe that models with more parameters are better at the counterfactual rewriting task than smaller models. The GPT2-M variants consistently outperform the GPT and GPT2-S models, regardless of the objective on which the model was trained. Interestingly, however, the GPT model appears to generally outperform the GPT2-S model on the counterfactual question **(3)**, indicating that the domain on which models are pretrained does affect how adaptable their representations are to the story rewriting task.

Domain Adaptation Another pattern we notice is that fine-tuning on the ROCStories data (FT) is always helpful for increasing performance on counterfactual relevance (**CF (3)** in Table 4), indicating adapting to the ROCStories-style language distribution helps the model learn to produce relevant rewrites for counterfactuals, especially for models with fewer parameters. The **Plot (2)** question in Table 4 indicates why this might be the case, as the zero-shot models tend to produce more creative rewritings that are not at all tied to the original story. Interestingly, however, fine-tuning with the larger set of counterfactuals (CF loss) does not seem to help in rewriting endings that relate to the counterfactuals well.

Supervised vs. Unsupervised Learning A surprising observation is that using the dataset

COUNTERFACTUAL - Human Judges Preferred			
Best model	Neutral	Comparator	
M+Sup 20.0	7.0	**29.5**	M+FT+CF
M+Sup 19.0	3.0	**38.5**	M+FT
M+Sup **23.5**	14.0	4.5	M+Recon+ CF
M+Sup 26.5	5.0	**33.5**	M+ zero-shot
M+Sup **14.0**	18.5	6.0	S+Sup
M+Sup **18.5**	20.0	8.0	GPT + Sup
M+Sup 10.0	15.0	**52.0**	Human

PLOT - Human Judges Preferred			
Best model	Neutral	Comparator	
M+Sup **57.5**	14.5	13.5	M+FT+CF
M+Sup **58.5**	16.5	12.5	M+FT
M+Sup 11.5	60.0	**16.5**	M+Recon+CF
M+Sup **63.0**	14.5	11.0	M+zero-shot
M+Sup 11.5	62.5	**12.5**	S+Sup
M+Sup 14.5	61.0	**15.0**	GPT+Sup
M+Sup 22.0	47.5	**25.0**	Human

PREMISE - Human Judges Preferred			
Best model	Neutral	Comparator	
M+Sup **35.5**	31.0	16.5	M+FT+CF
M+Sup **32.5**	39.5	14.0	M+FT
M+Sup 10.5	65.0	9.0	M+Recon+CF
M+Sup **46.5**	29.5	13.0	M+zero-shot
M+Sup **8.5**	71.0	7.5	S+Sup
M+Sup **12.0**	68.0	7.5	GPT+Sup
M+Sup 12.5	59.0	**22.5**	Human

Table 5: Pairwise human comparison between the best model (GPT2-M + Sup) and comparison models on all three questions. "Neutral" means both are "equally good". Percentage of "equally bad" are omitted.

of labeled rewritten endings for training does not seem to help the language models learn to rewrite endings better. While the supervised models are generally able to adhere to the plot better than unsupervised methods, their new endings do not score well on question **(3)**, indicating that they may be copying the original ending or learning to paraphrase the original story ending without acknowledging the counterfactual sentence. This points to the fact that this task cannot be trivially solved by adding more paired data, since adding more data merely simplifies to having more stories in the dataset, without necessarily learning to handle counterfactuals more effectively.

5.2 Pairwise Model Preference

Setup We conduct a pairwise comparison between the best model (GPT2-M + Sup) with other models along the same three dimensions as in the first evaluation setting (section 5.1).

Metric	(1) Prem	(2) Plot	(3) CF
BLEU-4	**.2623**	**.6792**	**-.1804**
ROUGE-L	**.3187**	**.7484**	**-.1423**
WMS	**.2713**	**.5809**	-.0343
S+WMS	**.2789**	**.6075**	-.0538
BERT	**.2124**	**.1929**	**.1067**
BERT-FT	**.2408**	**.1847**	**.0995**

Table 6: Pearson correlation between automatic metrics and human scores. **Bolded** numbers are statistically significant at p < 0.05.

Specifically, crowdworkers were presented outputs of a pair of systems, and asked to choose which one is better, or "equally good" or "equally bad", in terms of each of the three criteria. As in section 5.1, we evaluate 100 outputs of each model.

Results In Table 5, we present the human preference results, showing that the best model outperforms the comparison baselines in terms of consistency with premise, while being less consistently better with regards to the other two questions. Interestingly, a model that performs better on one of the evaluated dimensions often performs worse for another question, indicating plenty of room for future work in counterfactual reasoning for story rewriting.

6 Challenges for Automatic Metrics

While human scores provide the clearest insight into how models are able to complete the counterfactual rewriting task, their associated cost makes them difficult to scale to larger evaluation sets. To provide further insight into the performance of candidate models, we explore how different automatic metrics evaluate the produced generations.

6.1 Metrics

Overlap Metrics The most common metrics used in evaluating text generation are based on textual overlap between a candidate generated sequence and set of reference sequences provided by the dataset. **BLEU** (Papineni et al., 2002) is perhaps the most widely used metric in text generation, which computes the number of overlapping n-grams between the generated and reference sequences. Another commonly used metric in text generation (though originally designed for extractive summarization) is **ROUGE-L** (Lin, 2004), which measures the

length of the longest common subsequence (LCS) between a candidate generation and reference. We report the performance of all models on both of these metrics.

Model-based Metrics Although BLEU and ROUGE are widely used in text generation, they use exact string matching, and thus fail to robustly match paraphrases and capture semantically-critical ordering changes. Recently, there has been a growing body of work in producing model-based metrics (Lowe et al., 2017) that use trained models and embeddings to score a sequence.

Kusner et al. (2015) proposed Word Mover's Distance, which defines the distance between two texts as the minimal cost of transforming one sequence's word embeddings to the other's. The measure finds a matching between the two texts that minimizes the total Euclidean distance between the matched word embeddings. Following Kilickaya et al. (2017), we take the negative exponential of this distance to get **Word Mover's Similarity (WMS)**. More recently, Clark et al. (2019) proposed **Sentence + Word Mover's Similarity (S+WMS)** to extend WMS for longer multi-sentence texts by using sentence representations in the minimum distance calculation in addition to word embeddings.[1]

Other recent methods use contextualized embeddings (Devlin et al., 2018) to compute similarity between sequences. We use **BERTScore** (Zhang et al., 2019), which computes cosine similarity between two sentences using BERT encodings. Zhang et al. show that BERTScore correlates better with human judgments than existing metrics such as BLEU, ROUGE, and other learning-based metrics. To adapt BERTScore to our task, we finetune BERT on ROCStories using the same training framework from Devlin et al. (2018) and compute **BERT-FT** the same way as before.

6.2 Human Correlation with Metrics

Recent work in text generation (Wiseman et al., 2017) and dialogue (Liu et al., 2016) have explored the limitations of automatic metrics for text production tasks. Due to the highly semantic nature of the counterfactual rewriting task and the need to recognize subtle changes in

---

[1]We follow previous work and use GloVe embeddings (Pennington et al., 2014) to represent words and the averaged word embeddings to represent sentences.

	BLEU-4	ROUGE-L	BERT	BERT-FT	WMS	W+SMS
Training: Pretrained Only				*Input:* $s_1 s_2'$		
GPT + zero-shot	1.25	18.26	59.50	58.28	0.30	0.97
GPT2-S + zero-shot	1.28	20.27	59.62	58.11	0.33	1.09
GPT2-M + zero-shot	1.51	19.41	60.17	58.59	0.34	1.12
Training: Unsupervised + Generative				*Input:* $s_1 s_2'$		
GPT + FT	4.20	24.55	64.38	62.60	0.56	1.48
GPT2-S + FT	3.78	24.18	64.25	62.60	0.54	1.40
GPT2-M + FT	4.09	24.08	62.23	62.49	0.53	1.42
GPT + FT + CF	3.82	24.21	64.48	62.66	0.57	1.45
GPT2-S + FT + CF	3.96	24.06	64.50	62.71	0.53	1.44
GPT2-M + FT + CF	4.00	24.38	64.31	62.59	0.48	1.33
Training: Unsupervised + Discriminative				*Input:* $s_1 s_2 y [S] s_1 [MASK]$		
GPT2-S + Recon + CF	47.08	51.19	**63.82**	62.36	5.53	8.08
GPT2-M + Recon + CF	**76.57**	**71.35**	64.15	62.49	**18.29**	**20.87**
Training: Supervised + Discriminative				*Input:* $s_1 s_2 y [S] s_1 s_2'$		
GPT + Sup	80.09	75.03	64.15	62.36	20.93	23.37
GPT2-S + Sup	79.03	73.31	64.14	62.40	20.57	22.97
GPT2-M + Sup	76.63	74.42	64.06	**62.33**	19.62	22.01
Human	65.12	68.58	63.58	61.82	16.95	19.16

Table 7: Results on automatic metrics for the cross-product of the models and loss functions proposed in Section 4. **Bolded** results are closest to the human score.

event descriptions, we anticipate that automatic metrics would have difficulty assessing rewritten endings. To test the correlation between available evaluation metrics for long-form generation and human opinions of quality of counterfactual generations, we compute the Pearson Correlation between automatic scores and human scores for 800 validation set data points, 300 taken from the gold annotations and 100 generated from each of the 5 GPT2-M variants.[2] For each example, we use the same questions and Likert scale evaluation as in §5 and report the results in Table 6.

As expected, the automatic metrics are decently correlated with human scores for adherence to the premise sentence and plot. However, these same metrics correlate negatively with question (3) – adherence to the counterfactual sentence – indicating poor measurement of counterfactual understanding if they were to be reported in their typical manner (i.e., higher score indicating superior performance). Only the BERTScore metrics appear to positively correlate with human scores for counterfactual understanding, making them usable for evaluating generations across properties related to all three questions. However, the correlation is weak, and the results in Table 7 indicate that the BERTScore metrics are difficult to distinguish between models.

---

[2]We include both human annotations and model-generated outputs in this computation to encourage diversity of source.

7 Conclusion

We introduced a new task of *Counterfactual Story Rewriting* that challenges current language understanding and generation systems with counterfactual reasoning. Our new dataset, TIMETRAVEL, provides nearly 30k counterfactual revisions to simple commonsense stories together with 100k counterfactual sentences. We establish baseline performances of state-of-the-art neural language models with over 14 model variants with zero-shot, unsupervised and supervised settings. The empirical results demonstrate that while neural language models show promises, they generally have difficulties in rewriting the consequences of the counterfactual condition with full consistency, suggesting more focused research on integrating true reasoning capabilities to neural language models.

Acknowledgements

References

Leon Bottou. 2019. Learning representations using causal invariance. In *ICLR*.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *JMLR*.

Ruth MJ Byrne. 2002. Mental models and counterfactual thoughts about what might have been. *Trends in cognitive sciences*, 6(10):426–431.

Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *EMNLP*, pages 1797–1807.

Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. 2019. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kai Epstude and Neal J Roese. 2008. The functional theory of counterfactual thinking. *Personality and Social Psychology Review*, 12(2):168–192.

Nelson Goodman. 1947. The problem of counterfactual conditionals. *The Journal of Philosophy*, 44(5):113–128.

Jerry R Hobbs. 2005. Toward a useful concept of causality for lexical semantics. *Journal of Semantics*, 22(2):181–209.

Zhiting Hu, Haoran Shi, Bowen Tan, Wentao Wang, Zichao Yang, Tiancheng Zhao, Junxian He, Lianhui Qin, Di Wang, et al. 2019. Texar: A modularized, versatile, and extensible toolkit for text generation. In *ACL, System Demonstrations*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *ICML*.

Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. 2017. Re-evaluating automatic metrics for image captioning. In *EACL*.

Rik Koncel-Kedziorski, Ioannis Konstas, Luke S. Zettlemoyer, and Hannaneh Hajishirzi. 2016. A theme-rewriting approach for generating algebra word problems. In *EMNLP*.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In *ICML*.

Guillaume Lample, Sandeep Subramanian, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *ICLR*.

Carolin Lawrence and Stefan Riezler. 2018. Improving a neural semantic parser by counterfactual learning from human bandit feedback. In *ACL*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *EMNLP*.

Ryan Lowe, Michael Noseworthy, Iulian Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In *ICLR*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and evaluation framework for deeper understanding of commonsense stories. *arXiv preprint arXiv:1604.01696*.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Judea Pearl. 2000. *Causality: models, reasoning and inference*, volume 29. Springer.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Karl Pichotta and Raymond Mooney. 2014. Statistical script learning with multi-argument events. In *EACL*, pages 220–229.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Semeval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *SemEval@NAACL-HLT*.

Maarten Sap, Ronan LeBras, 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 *AAAI*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *NeurIPS*.

Youngseo Son, Anneke Buffone, Joe Raso, Allegra Larche, Anthony Janocko, Kevin Zembroski, H. Andrew Schwartz, and Lyle H. Ungar. 2017. Recognizing counterfactual thinking in social media texts. In *ACL*.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in data-to-document generation. In *EMNLP*.

Jim Woodward. 2002. What is a mechanism? a counterfactual account. *Philosophy of Science*, 69(S3):S366–S377.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *EMNLP*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *ACL*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan R. Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *ICCV*, pages 19–27.