Generating Contrastive Narratives Using the Brownian Bridge Process for Narrative Coherence Learning

Feiteng Mu, Wenjie Li
The Department of Computing, The Hong Kong Polytechnic University, Hong Kong
{csfmu,cswjli}@comp.polyu.edu.hk

Abstract

A major challenge for narrative reasoning is to learn narrative coherence. Existing works mainly follow the contrastive learning paradigm. However, the negative samples in their methods can be easily distinguished, which makes their methods unsatisfactory. In this work, we devise two strategies for mining hard negatives, including (1) crisscrossing a narrative and its contrastive variants; and (2) event-level replacement. To obtain contrastive variants, we utilize the Brownian Bridge process to guarantee the quality of generated contrastive narratives. We evaluate our model on several tasks. The result proves the effectiveness of our method, and shows that our method is applicable to many applications.

1 Introduction

Narrative reasoning (Charniak, 1972; Winograd, 1972) is an account of the development of events, along with explanations of how and why these events happened (Hutto, 2015), which has provoked a variety of applcations, including commonsense causal reasoning (Roemmele et al., 2011; Gordon et al., 2012; Luo et al., 2016), abductive reasoning (Bhagavatula et al., 2019), and so on.

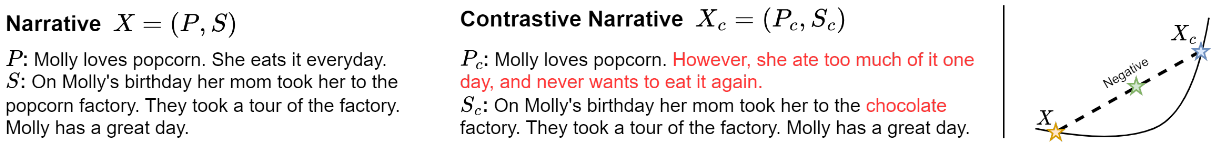
A major challenge for narrative reasoning is to evaluate narrative coherence (Mostafazadeh et al., 2016). Existing methods mainly focus on devising self-supervised tasks, in which positive samples are from large-scale real narratives (Mostafazadeh et al., 2016; Yao and Huang, 2018), and negative samples are created by sampling-based strategies. For example, Xie et al. (2020); Lin et al. (2020b); Uehara et al. (2020) create negative samples by shuffling or masking real narratives. Krishna et al. (2022) incorporates randomly sampled sequences and model-completed (Radford et al., 2019; Brown et al., 2020) sequences as negative samples. However, these strategies are generally coarse-grained and superficial. The resulting negatives still face problems of low quality, such as being irrelevant or repetitive (Krishna et al., 2022), making them less representative, and easily distinguishable.

Hard negatives are critical in the contrastive learning framework (Wu et al., 2017; Mishchuk et al., 2017; Xuan et al., 2020). The ideal of hard negative samples should be that are similar to a real narrative but actually less coherent. To mine such negatives, a possible approach is to introduce contrastive narratives. Contrastive narratives are examples that are similar in content, but convey different semantics (Margatina et al., 2021; Wang et al., 2021). Due to this property, we can crisscross[1] a narrative and its contrastive variants to obtain negative samples, as shown in Figure 1. The resulting negatives should be similar to the real narratives but less coherent, making them good candidates for hard negatives. However, existing works for collecting contrastive narratives rely heavily on manual annotation, which is costly and not scalable. To solve this problem, exploiting automated methods has great value, but is difficult since it requires preserving subtle differences while providing a clear delineation between the observed narrative and the generated ones.

Actually, the generation of contrastive narratives involves exploring the latent space surrounding a given narrative, enabling the creation of similar narratives with distinct characteristics. Assuming that the evolution tendency of an observed narrative can be represented as a continuous trajectory in latent space, which can be modeled by Brownian motion (Revuz and Yor, 2013; Wang et al., 2022). Consequently, we can sample the latent trajectories which exhibit proximity to the observed trajectory, and then decode the sampled trajectories into explicit narratives. But the problem is that the

[1] For example, according to $X = (P, S)$ and $X_c = (P_c, S_c)$, we can exchange their prefixes and suffixes to obtain the negatives (P, S_c) and (P_c, S). We define this strategy as “crisscrossing", and use this definition in the rest of our paper.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6538–6555
August 11-16, 2024 ©2024 Association for Computational Linguistics



Narrative $X = (P, S)$

P: Molly loves popcorn. She eats it everyday.
S: On Molly's birthday her mom took her to the popcorn factory. They took a tour of the factory. Molly has a great day.

Contrastive Narrative $X_c = (P_c, S_c)$

P_c: Molly loves popcorn. However, she ate too much of it one day, and never wants to eat it again.
S_c: On Molly's birthday her mom took her to the chocolate factory. They took a tour of the factory. Molly has a great day.

Figure 1: We define that an example consists of a prefix (P) and a suffix (S). **Left**: An ideal contrastive variant X_c, which is similar to X but conveys different semantics. Text with red color denotes the difference. **Right**: The solid line denotes the data manifold. The dashed line represents the methods for creating negatives, such as Mixup (Zhang et al., 2017) or crisscrossing. As X_c approaches X, the created negative example should be more "hard".

decoded narratives may differ significantly in content from the observed narrative, which may not meet the requirements for contrastive narratives. To simplify the problem, we further suggest that contrastive narratives keep the same endpoint as the observed narrative, which directly models the fact that a narrative event can evolve to the same end through different paths (Qin et al., 2019). Based on this constraint, we are able to sample different trajectories from the Brownian Bridge (Majumdar and Orland, 2015; Wang et al., 2022) region that is centered around the observed narrative. The sampled trajectories are decoded as narratives with the same start and end as the observed narrative, while also having similar but different intermediate event chains. Then we crisscross the observed narrative and the generated ones to synthesize negative samples. In fact, in our crisscrossing strategy, the start and end points of resulting negatives remain the same as the positive ones. That is, the start and end of positive narratives will never be perturbed. This further motivates us to design an event-level perturbation to obtain negatives, as more diverse negatives definitely benefit contrastive learning.

In this paper, we devise two strategies to create hard negatives for narrative coherence learning. The first strategy crisscrosses a narrative with its contrastive variants, and the second strategy performs an event-level replacement. To obtain contrastive narratives, we first sample different latent trajectories from the Brownian Bridge region, then fix the start and end points of the narrative, and generate diverse contrastive narratives.

Our contributions can be summarized as follows. (1) Based on the Brownian Bridge process, we generate high-quality contrastive narratives, which are used to synthesize hard negatives. (2) We propose a new *coh*erence *eval*uator (*CohEval*), which is enhanced by diverse and high-quality hard negatives. Our model is trained entirely through self-supervised contrastive learning, and can be applied to a wide range of downstream tasks.

We evaluate our model on multi-choice tasks and one narrative generation task. We also conduct an in-depth analysis of our negative sample synthesis strategies. The experimental results demonstrate the effectiveness of our method. Code is released at github.com/mufeiteng/ContrastiveNarratives.

2 Related Work

Counterfactual Story Generation Counterfactual story generation (Qin et al., 2019; Hao et al., 2021; Chen et al., 2021) requires predicting how alternative events, contrary to what actually happened, might have resulted in different story endings. Existing works for counterfactual story generation mainly include manual annotation (Qin et al., 2019) or supervised fine-tuning (Hao et al., 2021) methods. In our work, contrastive narratives can be seen as a special case of counterfactual narratives, where we confine that an observed narrative and its contrastive variants have the same start and end. We generate contrastive narratives in a self-supervised manner, which is based on the Brownian Bridge process (Wang et al., 2022).

Language Modeling via Stochastic Process Generating long, coherent text is conceptually difficult for autoregressive models because they lack the ability to model text structure and dynamics (Lin et al., 2020a). Wang et al. (2022) explicitly models latent structure with Brownian Bridge dynamics, which can capture how sentence embeddings evolve over a document. (Wang et al., 2023) uses the stochastic process to model the temporal dynamics of dialogue paths. Motivated by Wang et al. (2022), we use Brownian Bridge for generating contrastive stories because it allows for the smooth modeling of gradual changes between two narrative states. Based on the simple constraint, we are able to generate coherent contrastive narratives, which are used to synthetic hard negatives.

Hard Negatives Mining Earlier researchers devise a series of corrupting strategies, such as shuffling, masking, or lexical conversion, to perturb real narratives (Cai et al., 2020; Xie et al., 2020; Lin et al., 2020b; Uehara et al., 2020; Zhou et al., 2022a,b). Recent methods focus on mining hard negatives. For example, Jwalapuram et al. (2021) retrieves hard negatives from the corpus with a momentum encoder. Krishna et al. (2022) incorporates random sequences and model-generated sequences as hard negatives. Kalantidis et al. (2020) mixes different negatives in latent space to create hard negatives. Zhang et al. (2022) mixes multiple positive samples to produce hard negatives. Instead, we propose to use a narrative with its contrastive variants to synthesize hard negatives. Since the contrastive narratives are similar to the original ones, we can obtain qualified negatives.

3 Methods

3.1 Data Preparation

Following the previous method (Cai et al., 2020), we use RocStories (Mostafazadeh et al., 2016) as our data corpus, since it contains abundant event commonsense knowledge, making it a good resource for narrative reasoning. Due to the limitation of computational resources, we randomly select about 20k samples from RocStories, and denote them as the positive sample set $\mathcal{D}^+$. Each sample in $\mathcal{D}^+$ is a narrative $X = \{e_1, \cdots, e_5\}$, in which each e_i $(i = 1, \cdots, 5)$ is an event. Following previous works, we lay narrative coherence learning in the contrastive learning framework, in which the negative samples are needed for training.

We devise two strategies for mining hard negatives: (1) crisscrossing a narrative and its contrastive variants; (2) event-level replacement. Next, we introduce how to obtain contrastive narratives.

3.2 Generating Contrastive Narratives via the Brownian Bridge Process

Given a narrative, the contrastive variants should be similar to it and express distinctive characteristics. We regard this problem as exploring the latent space surrounding the given narrative, and propose to model this problem by the Brownian Bridge process (Wang et al., 2022). The density distribution of a Brownian Bridge process from a start point z_0 at $t = 0$ to an endpoint z_T at $t = T$ is:

$$p(z_t|z_0, z_T) \sim \mathcal{N}((1 - \frac{t}{T})z_0 + \frac{t}{T}z_T, \frac{t(T-t)}{T}). \quad (1)$$

The Brownian Bridge density acts like a noisy linear interpolation between the start and end point of the trajectory, where the intermediate point z_t should be more like z_0 at the start and more like z_T at the end of the trajectory. Uncertainty is highest in the middle region, and low near the start and end points (rf. Figure 2, the green region). The characteristic of the Brownian Bridge is to maintain a smooth transition between the sampling midpoint and the starting and ending points.

Following (Wang et al., 2022), we pre-train an event encoder with the Brownian Bridge contrastive loss[2]. Given the starting and ending events, the encoder is responsible for encoding an event e into latent code z and ensuring that the latent codes of any intermediate events conform to the Brownian bridge distribution (Equation 1). Once the event encoder is trained, it will be frozen in all subsequent processes and will never get updated.

To train the contrastive narratives generator, we automatically construct training examples. Given the start event e_1 and the end event e_5, we first use the event encoder to encode them into latent vectors z_1 and z_5. Once z_1 and z_5 are obtained, we know the corresponding Brownian Bridge density (Equation 1), and we can sample middle points to obtain diverse latent trajectories, i.e., $\mathbf{Z} = \{z_1, z_2, z_3, z_4, z_5\}$. To generate contrastive narratives, we encode (e_1, e_5) with BART (Lewis et al., 2019) to obtain the context embeddings:

$$\mathbf{H}_c = \text{BARTEncoder}([e_1, e_5]), \quad (2)$$

where $[;]$ denotes the concatenation, $\mathbf{H}_c \in \mathcal{R}^{l \times d}$, l is the length of $[e_1; e_5]$. Next, given $\mathbf{H}_c$ and latent codes $\mathbf{Z}$, we generate middle events $y = (e_2, e_3, e_4)$. Specifically, let y_t denotes the t-th tokens in y. At the timestep t, the generator must predict y_t using $\mathbf{H}_c$, all tokens in the past $y_{<t}$, as well as the event latent codes $\mathbf{Z}$:

$$\begin{aligned} \mathbf{h}_{y_t} &= \text{BARTDecoder}(y_{<t}, \mathbf{H}_c, \mathbf{W}_z^T \mathbf{Z}) \\ P(y_t|Y_{<t}) &= \text{softmax}_V(\mathbf{W}_v \mathbf{h}_{y_t} + b). \end{aligned} \quad (3)$$

where V denotes the standard vocabulary, $\mathbf{W}_z$ denotes a linear layer that maps the dimension of z to be identical to $\mathbf{H}_c$. This can be seen as decoding a latent trajectory $\{z_1, z_2, z_3, z_4, z_5\}$ into narrative events given the start event e_1 and end event e_5.

However, in our preliminary trials, we found that the generated narratives are coherent but less similar to the original one, which brings difficulties to the construction of hard negatives. The possible reason is that the encoding process, i.e., encoding

[2] See Appendix A and (Wang et al., 2022) for details.

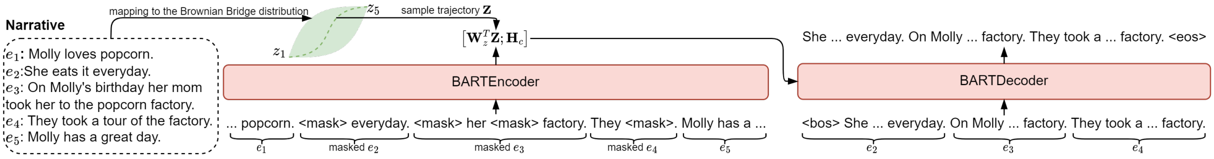

Narrative
e_1: Molly loves popcorn.
e_2:She eats it everyday.
e_3: On Molly's birthday her mom took her to the popcorn factory.
e_4: They took a tour of the factory.
e_5: Molly has a great day.
mapping to the Brownian Bridge distribution
z_1
z_5
sample trajectory $\mathbf{Z}$
$[\mathbf{W}_z^T\mathbf{Z}; \mathbf{H}_c]$
BARTEncoder
... popcorn. <mask> everyday. <mask> her <mask> factory. They <mask>. Molly has a ...
e_1
masked e_2
masked e_3
masked e_4
e_5
She ... everyday. On Molly ... factory. They took a ... factory. <eos>
BARTDecoder
<bos> She ... everyday. On Molly ... factory. They took a ... factory.
e_2
e_3
e_4

Figure 2: The training phrase of contrastive narratives generation. Given z_1 and z_5, $\mathbf{Z}$ is sampled according to Equation 1. The masked e_2, e_3, e_4 are used as the prompt for decoding.

e to z, lost too much information, making it difficult for the model to reconstruct y. To solve this problem, we randomly mask the y with the ratio of ρ (0.85 by default), and use the masked sequence as the prompt for the decoding phrase, which encourages the generator to generate more similar events to y. Actually, these can be seen as two types of constraints, where $\mathbf{Z}$ requires that y and the generated text show similar trajectories in latent space, and the masked prompt requires that y and the generated text are similar in vocabulary. The whole training process is shown in Figure 2.

When training, we use RocStories excluding $\mathcal{D}^+$ as training data. We have also tried other pretrained models, such as GPT2 (Radford et al., 2019) and T5 (Raffel et al., 2020), and BART empirically performs best, as shown in Appendix B. Therefore, we choose BART as the backbone. After training, for each $X \in \mathcal{D}^+$, we fix its start and end events, then sample different intermediate events. For each X, we first generate 200 candidates, then use several criteria[3] to filter low-quality candidates. We finally retain N (60 by default) most-qualified contrastive examples.

3.3 Synthesizing Negative Examples

We devise two strategies to create negative examples. The first strategy crisscrosses a narrative with its contrastive variants, and the second strategy performs an event-level replacement.

3.3.1 Crisscrossing a Narrative and its Contrastive Variants

Note that each X contains five events. For simplicity, we define the first three events as the prefix (P), and the last two events as the suffix (S), so that we denote $X = (P, S)$ and the contrastive variant $X_c = (P_c, S_c)$. Then we are able to synthesize the negative example $X^- = (P_c, S)$. The basic intuition is: P_c is coherent with S_c, so it should be less coherent with S. This is because X and X_c are different paths with the same start and end points. More specifically, by comparing $X = (P, S)$ and $X^- = (P_c, S)$, we can find that the difference between X and X^- lies in the second and third events. In other words, we have replaced two events in the original story with different events. Therefore, the resulting samples should be less coherent than X. Meanwhile, $X^- = (P_c, S)$ is similar to $X = (P, S)$, making it qualified as a hard negative[4]. With loss of generality, we denote the obtained negative samples as $\mathcal{C}_X = \{X_i^-\}_{i=1}^{2N}$.

For each training epoch, we randomly sample K (15 by default) negatives samples $\{X_k^-\}_{k=1}^{K}$ from $\mathcal{C}_X$ for each X, and feed them as well as X into a pre-trained language model (PLM) (Devlin et al., 2018a; Liu et al., 2019), e.g. RoBERTa, to obtain sequence-level representations:

$$\mathbf{h}^+ = \text{RoBERTa}(X), \mathbf{h}_k^- = \text{RoBERTa}(X_k^-), \quad (4)$$

where $k = \{1, \cdots, K\}$, $\mathbf{h}^+$ and $\mathbf{h}_k^- \in \mathcal{R}^d$, d is the hidden size of RoBERTa. We have also tried BERT (Devlin et al., 2018b) as the backbone, as shown in Appendix E. Next, the sequence-level representations are passed into a linear layer $\mathbf{W}_c \in \mathcal{R}^d$ to derive coherence scores of all samples:

$$s^+ = \mathbf{W}_c^T \mathbf{h}^+, s_k^- = \mathbf{W}_c^T \mathbf{h}_k^-. \quad (5)$$

Lastly, we use the contrastive classifying objective to distinguish the positive examples from the corresponding negative examples:

$$\mathcal{L}_1 = -\frac{1}{|\mathcal{D}^+|} \sum_{\mathcal{D}^+} \log \frac{\exp(s^+)}{\exp(s^+) + \sum_{k=1}^{K} \exp(s_k^-)}. \quad (6)$$

It should be noted that the difference between $X^- = (P, S_c)$ and $X = (P, S)$ lies in the third and fourth events, i.e., e_3 and e_4. Due to the masked prompt, some tokens in (e_3, e_4) of X^- are similar to those of X, making X^- qualified. However, in the crisscrossing strategy, e_1 and e_5 will never be perturbed. This further motivates us to perform a simple event-level perturbation to X to create more diverse negative samples.

[3] See Appendix C for details.

[4] Similarly, we can obtain the negative example $X^- = (P, S_c)$ by defining the first two events as the prefix.

3.3.2 Event-level Replacement

Due to the fact that events are the basic semantic unit of neural language, for a narrative, if we replace a component event with another similar but different event, the resulting example should be less coherent and similar to the original narrative.

Specifically, based on $\mathcal{D}^+$, we build an event pool, which consists of about 100k different events. We pre-compute the cosine similarity among all event pairs using SimCSE (Gao et al., 2021), and cache the top 20 most similar events Q^e for each query event e. Then, given a positive example X, we randomly select a position i and replace i-th event e_i with a randomly sampled event $\bar{e}$ from Q^e to create a negative example $\bar{X} = \{\cdots, e_{i-1}, \bar{e}, e_{i+1}, \cdots\}$. Likewise, for each training epoch, we create K negatives samples $\{\bar{X}_k\}_{k=1}^K$. After obtaining hidden states of negatives: $\bar{\mathbf{h}}_k = \text{RoBERTa}(\bar{X}_k)$, we derive coherence scores of all samples and use the contrastive loss to rank the positive sample above the negatives:

$$\begin{aligned} & s^+ = \mathbf{W}_c^T \mathbf{h}^+, \bar{s}_k = \mathbf{W}_c^T \bar{\mathbf{h}}_k, \\ & \mathcal{L}_2 = -\frac{1}{|\mathcal{D}^+|} \sum_{\mathcal{D}^+} \log \frac{\exp(s^+)}{\exp(s^+) + \sum_{k=1}^K \exp(\bar{s}_k)}. \end{aligned} \quad (7)$$

3.4 Training and Knowledge Transferring

When training, the final loss is

$$\mathcal{L} = \gamma \mathcal{L}_1 + (1-\gamma)\mathcal{L}_2, \quad (8)$$

where γ is set to 0.5. It should be noted that another way is to merge two types of negatives and directly perform contrastive learning. However, this requires more GPU memory, which exceeds our condition. Therefore, we calculate the two losses separately and then average them.

Our *CohEval* can be easily transferred to many downstream applications. For example, for the multi-choice task with a input C and option candidates $O = \{o_1, \cdots, o_n\}$, we can use *CohEval* to select most reasonable o by:

$$o \leftarrow \arg\max_i \textit{CohEval}([C, o_i]). \quad (9)$$

Motivated by existing plug-and-play text generation methods (Miao et al., 2018; Chen et al., 2021), we also evaluate our *CohEval* in narrative text generation, with *CohEval* as coherence guidance. Details can be seen in the experiment.

4 Experiment

4.1 Datasets and Experimental Details

The evaluation datasets include COPA (Roemmele et al., 2011), e-Care (Du et al., 2022a), αNLI (Bhagavatula et al., 2020), Cloze (Mostafazadeh et al., 2016), Swag (Zellers et al., 2018), HellaSwag (Zellers et al., 2019), and TimeTravel (Qin et al., 2019). TimeTravel is a text-generation dataset, while others are multi-choice datasets. We evaluate our model on these datasets in the zero-shot setting. Note that the test sets of e-Care and HellaSwag are not released. So we evaluate our model on the validation set of the three datasets. The statistics of the datasets, as well as the experimental details are shown in Appendix D.

Methods	COPA	e-Care	αNLI	Cloze	Swag	HS.
LLMs-based Prompting						
Alpaca-lora (7B)	57.4	54.5	52.6	66.1	36.0	30.2
ChatGLM2 (6B)	78.1	66.9	58.1	84.3	48.7	41.2
ChatGPT	96.2	81.8	75.5	94.7	70.7	76.4
Contrastive Training Based Methods						
RankGen(base)	63.8	70.3	52.2	50.7	46.3	33.9
RankGen(large)	70.2	72.1	54.8	54.4	49.2	40.5
EventBERT	N/A	N/A	59.5	75.6	N/A	N/A
CohEval (ours)	**77.8**	71.9	**67.6**	**77.6**	**67.4**	**44.9**
Ablation Study						
M_{ER}	73.4	**75.4**	65.3	77.1	61.8	38.9
M_{CC}	75.8	68.2	67.2	69.4	66.9	44.7

Table 1: The accurary (%) on multi-choice datasets. HS. denotes HellaSwag. Scores with **bold** denote the best results among contrastive training based methods.

4.2 Baselines and Metrics

For multi-choice tasks, the metric is Accuracy. We compare our method with EventBERT (Zhou et al., 2022a), RankGen (Krishna et al., 2022), and several large language models (LLMs), including Alpaca-lora (7B)[5], ChatGLM2 (6B) (Du et al., 2022b; Zeng et al., 2022) and ChatGPT (OpenAI. , 2023). For LLMs, we use one-shot prompting for experiments, the used prompts are in Appendix F. For TimeTravel, we follow Chen et al. (2021) and formulate this task in the MCMC-based sampling paradigm. The details are in Appendix G. We compare our method with DELOREAN (Qin et al., 2020), ClarET (Zhou et al., 2022b), CGMH (Miao et al., 2018), EDUCAT (Chen et al., 2021). Automatic evaluation metrics

[5] The checkpoint is at https://github.com/tloen/alpaca-lora.

include BLEU4 (Papineni et al., 2002), BertScore (Zhang et al., 2019), ENTScore (Chen et al., 2021), and HMean= $\frac{2\cdot\text{BLEU4}\cdot\text{ENTScore}}{\text{BLEU4}\cdot\text{ENTScore}}$ (Chen et al., 2021). Manual evaluation metrics include Fluency, Min-Edits (Chen et al., 2021), and Coherence.

4.3 Overall Results

Automatic Evaluation The automatic evaluation result can be seen in Table 1 and 2, respectively. We have the following observations.

- In Table 1, our model surpasses all contrastive training-based methods. This indicates that the negative samples we create are more qualified, which verifies the effectiveness of our method.
- Although there is still a significant gap compared to ChatGPT, our method surpasses smaller LLMs, e.g., ChatGLM2, on most datasets.
- In Table 2, our method outperforms EDUCAT. Since EDUCAT uses the off-the-shelf PLMs for evaluating coherence, the performance improvement proves that our CohEval is better at evaluating narrative coherence.
- Compared with our method, ChatGLM2 and ChatGPT achieve high ENTScore, but low BLEU4. This indicates that auto-regressive methods tend to generate coherence counterfactual ending with massive edits. These behaviors conflict with the requirements of the task.

Methods	BLEU4	BertS.	ENTS.	HMean
LLMs-based Prompting				
ChatGLM2 (6B)	16.47	60.03	66.15	26.37
ChatGPT	36.41	69.81	82.62	50.55
Off-the-shelf small PLMs				
DELOREAN	23.89	59.88	**51.40**	32.62
ClarET	23.75	63.93	N/A	N/A
CGMH[†]	41.09	73.90	28.06	33.34
EDUCAT	44.05	74.06	32.28	37.26
EDUCAT[†]	43.57	74.00	33.41	37.82
CohEval (ours)	42.46	73.36	37.39	**39.77**
Ablation Study				
M_{ER}	**44.18**	**74.34**	34.63	38.82
M_{CC}	42.99	73.64	35.78	39.05

Table 2: The automatic result on TimeTravel. [†] denotes our implementation. BertS. denotes BertScore. ENTS. denotes ENTScore. Scores with **bold** denote the best results among off-the-shelf small PLMs.

Ablation Study To investigate the influence of the two kinds of negatives, we devise two ablated variants: (1) M_{ER} which means we create negatives via **e**vent-level **r**eplacement; (2) M_{CC} which means we create negatives via the **c**riss**c**rossing strategy. The ablation study result is shown in Table 1, 2. We have the following observations.

- Compared to CohEval, M_{ER} and M_{CC} achieve lower ENTScore, indicating their weaker coherence evaluation abilities. But both variants obtain higher BLEU4 and BertScore. In TimeTravel, there is a trade-off phenomenon between BLEU and EntScore. This is because the gold y' is obtained through editing the original y with minimal-edits. This leads to a high word overlap between y' and y. Due to the weaker coherence evaluation abilities of the two variants, the probability of accepting transitions is lower when adopting MCMC for rewriting. In other words, when using M_{ER} and M_{CC}, the number of rewritings is relatively low, resulting in higher BLEU4 and BertScore but lower ENTScore.
- The best ENTScore is achieved by combining two kinds of hard negatives. This indicates the two kinds of negatives complement each other. The reason is that more diverse negative examples contribute to contrastive learning.
- M_{CC} generally performs better than M$_{ER}$. The possible reason is that, compared to the crisscrossing strategy, the event-level perturbation is more coarse-grained. Nevertheless, event-level replacement is an effective supplement to the crisscrossing strategies.

Manual Evaluation on TimeTravel We perform an A/B test to compare our method with several baselines. Following (Chen et al., 2021), the human evaluation mainly focuses on three primary criteria: i) Fluency, whether a model produces fluent text; ii) Coherence, the logical consistency between the counterfactual context (z, x') and the generated endings y; and iii) Min-Edits, the extent of minimal revision between two endings. We carry out a pairwise comparison with CGMH, EDUCAT, and two ablated models: M_{exp} and M_{imp}. We randomly sample 100 cases for each pair of models. Three annotators are recruited to make a preference among win, tie, and lose given the counterfactual context and two outputs by our model and a baseline respectively. The annotators are research students from the field of text generation to make sure they have a fair judgment of used metrics. We calculate Fleiss's kappa reliability as the inter-annotator agreement. As is shown in Table 3, LLMs are able to generate fluent and coherent counterfactual ending, but tend to massively edit the original ending, which coincides with the finding in automatic evaluation. Compared to EDUCAT and two ablated variants, CohEval achieves better fluency and coherence results. In addition, four models achieve similar Min-Edits results, this is because they run for the same editing steps. The Fleiss's kappa reliability of Fluency, Min-Edits, and Coherence is 0.488, 0.507, and 0.428, respectively.

Methods	Fluency		Min-Edits		Coherence	
	W(%)	L(%)	W(%)	L(%)	W(%)	L(%)
vs. EDUCAT†	27.0	13.7	23.0	24.7	33.7	4.7
vs. M_{ER}	25.7	16.7	22.3	23.3	28.0	6.7
vs. M_{CC}	20.0	12.0	23.7	22.3	23.0	7.0
vs. ChatGLM2	13.3	45.3	84.7	7.7	19.0	37.0
vs. ChatGPT	14.7	41.3	60.3	25.0	13.7	40.0

Table 3: Manual evaluation result on TimeTravel. Scores indicate the percentage of Win(W) and Lose(L).

Human Correlation with our CohEval Same as (Chen et al., 2021), we analyze the correlation between our CohEval and human ratings in terms of coherence evaluation. We calculate Pearson's r and Kendall's τ coefficients. The result is shown in Appendix H. All results show positive correlations. The result of our CohEval is close to that of ENTScore. Notice that ENTScore is trained with human-labeled counterfactual data, while our CohEval is trained in a self-supervised manner. This demonstrates the applicability of our CohEval.

Overall, the result demonstrates that our CohEval is a generic narrative coherence evaluator, and can be applied to a wide range of downstream tasks.

4.4 Deeper Analysis about Contrastive Narratives Generation

Indirect Evaluation through Multi-choice Tasks We conduct an ablation experiment to explore the impact of different sub-modules in contrastive narratives generation. We compare our Brownian-Bridge based method (denoted as "BB") with the following variants. (1) "w/o prompt", in which we ablate the masked prompt when training. (2) "w/o trajectory", in which we ablate the latent trajectories sampled from the Brownian bridge. (3) "Infilling", in which we ablate the masked prompt and the sampled latent trajectory when training. In this case, the ablated variant degenerates into a text-infilling model. We use the counterparts generated by different variants for crisscrossing to obtain negative examples, which are then used for contrastive learning. The result is shown in Table 4. We find: (1) Compared to "BB", "w/o prompt" and "w/o trajectory" get result drops, respectively; (2) "Infilling" gets a further performance drop.

The possible reasons lie in the following aspects. (1) If contrastive narratives are incoherent, then the synthesized negatives are not "hard". The sampled latent trajectories help to maintain the coherence of generated contrastive narratives, which benefits the quality of synthesized negatives. (2) The masked prompt helps to reduce the difficulty of the generation process, as a result, the obtained contrastive counterparts are similar to the original ones, making the resulting negatives more qualified.

Methods	COPA	e-Care	αNLI	Cloze	Swag	HS.	∇
BB (our M_{CC})	75.8	68.2	67.2	69.4	66.9	44.7	—
w/o prompt	79.0	65.4	68.5	75.6	59.2	39.9	-4.6
w/o trajectory	71.0	71.2	65.9	69.9	67.1	42.0	-5.1
Infilling	72.2	71.9	64.8	77.7	58.1	40.4	-7.1

Table 4: The result (%) of different kinds of counterparts for synthesizing negative examples.

Methods	Coherence		Similarity		SubtleDiff.	
	W(%)	L(%)	W(%)	L(%)	W(%)	L(%)
vs. w/o prompt	43.0	19.0	46.0	6.3	27.3	7.0
vs. w/o trajectory	53.7	15.3	26.7	7.7	28.0	12.7
vs. Infilling	60.3	10.3	56.3	5.7	49.0	6.7
vs. ChatGLM2	40.0	20.0	39.0	20.3	24.3	28.3
vs. ChatGPT	21.0	26.0	30.7	17.0	18.0	23.0

Table 5: The manual evaluation on contrastive narratives generation. We compare "BB" with "w/o prompt", "w/o trajectory", "Infilling", ChatGLM2, and ChatGPT.

Direct Evaluation through Manual Judgement We further conduct a manual evaluation to directly evaluate the quality of generated contrastive narratives. Since we want the generated narrative to be *similar* to the original one and reflect *subtle differences* (such as changes in opinions or entities) to make itself a different story, we use Coherence, as well as Similarity and SubtleDifference (SubtleDiff.) as metrics. Coherence reflect the logical consistency between the given (start,end) events and the generated middle events. Similarity reflects the similarity between the generated middle events and those of the original story. SubtleDiff. measures whether the generated example is a qualified contrastive narrative, which reflects subtle differences from the original story but is actually a different story. We randomly select 100 stories that have no overlap with train data for the experiment.

For each story, we use different models to generate its contrastive variant. We also perform a pairwise comparison with “w/o prompt", “Infilling", and two LLMs: ChatGLM2 and ChatGPT. The same three annotators are asked to make a preference among win, tie, and lose for each pair of generations. In Table 5, ChatGPT generally exhibits the best result, which reflects its powerful reasoning ability. Our “BB" is slightly inferior to ChatGLM2 on *SubtleDiff.* , but wins on the other two metrics. This indicates that our method is comparable to small LLMs. In addition, “BB" significantly surpasses the ablated variants. Specifically, we find that the masked prompt helps to improve Similarity, while latent trajectory helps to improve Coherence. This coincides with human intuition. The Fleiss’s kappa reliability of Coherence, Similarity, and SubtleDiff. is 0.369, 0.371, 0.244, respectively.

Generally, by utilizing the Brownian bridge process, we harvest qualified contrastive narratives, which contributes to contrastive learning.

4.5 Further Discussion

Strategies		COPA	e-Care	αNLI	Cloze	Swag	HS.
Mixup	Random	60.2	49.7	52.1	59.1	32.7	28.8
	w/o prompt	61.8	55.5	57.0	64.3	35.4	32.1
	BB	63.6	60.0	64.4	66.5	41.9	29.3
CrissC.	Random	72.6	71.8	58.8	70.0	53.6	37.4
	w/o prompt	79.0	65.4	68.5	75.6	59.2	39.9
	BB (our M_{CC})	75.8	68.2	67.2	69.4	66.9	44.7

Table 6: The result of different strategies for creating negatives. CrissC. denotes the crisscrossing strategy.

Influence of Different Strategies for Creating Negatives In our method, we crisscross a positive narrative with its contrastive counterparts to create negatives. Here, we further investigate the result when using Mixup (Zhang et al., 2017) to create negatives. The experimental setting is shown in Appendix I. We additionally explore three ways of obtaining the counterparts: (1) “BB" denotes our Brownian-Bridge based contrastive narratives; (2) “w/o prompt" denotes we ablate the prompt when generating contrastive narratives; (3) Random denotes we randomly select different positive narratives as counterparts. The result is shown in Table 6. We observe that:

- The crisscrossing strategy is superior then Mixup by a large margin. We speculate that in the era of self-attention (Vaswani et al., 2017), using the transformer to directly learn the representation of negative samples is better than manipulating representations of samples in the hidden space.

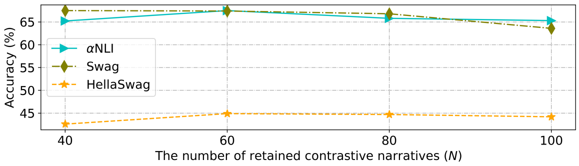


Figure 3: Results under the different number of retrained contrastive narratives.

- Whether adopting “CrissC." or Mixup, our BB-based contrastive narratives far surpass “random", which proves the strength of our method.

Results under Different Number of Retained Contrastive Narratives We explore the influence of the number of retained contrastive narratives. The result is shown in Figure 3. Our method generally achieves the best result when $N = 60$, and the result even decreases when N further increases. We speculate that as N increases, incoherent contrastive examples increase, which has a negative impact on the quality of synthesized negative examples. So, we set $N = 60$ by default.

Impact of the Mask Ratio ρ We investigate the impact of the different mask ratios ρ when generating contrastive narratives. In Table 7, the result is best when $\rho = 0.85$. As ρ decreases, the result gets worse. To investigate the reason, we manually examine the generated examples, and find the model tends to paraphrase the original story and generate duplicate examples when ρ decreases. This is because more information about the original story will be exposed when using a lower mask rate, making it easier to reconstruct the original story. We additionally calculate the diversity of the contrastive narratives generated at different ρ. We use Distinct-n (Li et al., 2015) as the metric. As shown in Table 7, as ρ decreases, the corresponding Distinct scores also decrease. This indicates that a lower mask rate ρ may lead to duplicate samples when the generation phase, which harms the diversity of synthesized negative samples. Therefore, we proactively filter out duplicate items.

The Reliability of Created Negative Examples We further analyze whether the created negative samples are indeed “negative". On the training set, we first use ENTScore to directly evaluate the coherence of positive samples and two types of negatives. As shown in Table 8, the real positive ex-

ρ	Accuracy(%)			Dist-2	Dist-3
	αNLI	Swag	HS.		
$\rho = 0.90$	65.2	**67.5**	42.6	26.4	41.0
$\rho = 0.85$	**67.5**	67.4	**44.9**	**27.1**	42.6
$\rho = 0.80$	66.2	66.5	43.3	26.8	**42.9**
$\rho = 0.70$	64.0	63.4	42.9	25.0	40.7

Table 7: The result under the different ρ. Dist-n denotes Distinct-n. Scores with **bold** denote the best result.

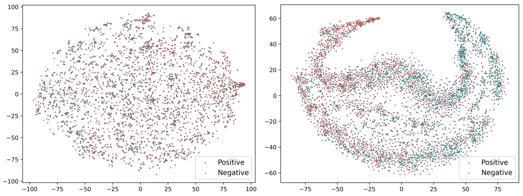
Types	ENTScore	FN Rate
Positive examples	94.6	N/A
Negatives via replacement	54.5	3.0%
Negatives via crisscrossing	65.9	4.3%

Table 8: The reliability evaluation of created negatives. FN denotes *false negative*.

amples receive an especially high ENTScore. However, the synthesized two types of negatives receive lower ENTScore, proving that they are obviously less coherent than positive examples. Next, we sample 100 cases and ask the annotators to make a judgment about whether the created ‘negatives‘ are actually more coherent than positives, making them false negatives. As shown in Table 8, both types of negatives show a low FN rate. We show the **error cases** in the Appendix J.

Visualize the Representations of Examples using t-SNE It is interesting to qualitatively visualize our model's ability to distinguish hard negatives. Based on the test set of TimeTravel, we are able to obtain positive examples and corresponding *hard negatives*. We leave the details in Appendix K. We use our CohEval and the ablated variant M_{ER}, respectively, to obtain the representations of the examples, then we use t-SNE (Van der Maaten and Hinton, 2008) to visualize the representations. As shown in Figure 4 (a), the representations of positive and negative examples obtained by M_{ER} entangle together, this shows that M_{ER}, a model that significantly outperforms baselines, still suffers from distinguishing the created positive and negative examples. But in Figure 4 (b), positive samples are concentrated on the right, while negative samples are concentrated on the left. This proves our CohEval's ability to distinguish positive examples from hard negatives, and confirms the effectiveness of the generated contrastive narratives.

Case Study Appendix L, Table 15 presents a case study for the task of TimeTravel. The counterfactual endings generated by ChatGLM2 and ChatGPT are coherent but very different from the original ending. This conflicts with the minimal-editing requirement of the task. On the contrary, based on the MCMC sampling, our method can produce similar and coherent counterfactual endings. Appendix L, Table 16 presents a case study for contrastive narratives generation. Due to the sampled different trajectories, in the case #1, our method shifts the topic of accent to personality, and produces a coherent story. And in the case #2, our method exchanges the opinions of two participants. On the contrary, the middle events generated by ChatGLM2 and ChatGPT show a significant difference from that of the original story.



(a) M_{ER} (b) CohEval

Figure 4: Visualization of the representations of examples obtained from different models.

5 Conclusion

In this paper, we propose to use the Brownian Bridge process to generate contrastive narratives, then we crisscross a positive story and its contrastive variants to create negative examples for contrastive learning. In addition, we devise the event-level replacement, which is an effective supplement to the crisscrossing strategy. The experiment verifies that (1) the generated contrastive narratives are qualified, and (2) our CohEval is effective and is a general coherence evaluator that is applicable to many downstream tasks.

Acknowledgements

We thank the anonymous reviewers for their encouraging feedback. This work is supported by Research Grants Council of Hong Kong(PolyU/15207920, PolyU/15207821, PolyU/5213323) and National Natural Science Foundation of China (62076212).

Limitations

To automatically generate contrastive narratives, we made the following assumption: the observed story and its contrastive variants have the same start and end events. However, this assumption may not be consistent with reality. In addition, under limited computing resources, we are unable to explore our method on larger data scales and larger pre-trained models. The experiment shows that our method is not able to surpass ChatGPT. But this does not mean that our work has no value in the era of large language models.

Our method is essentially a discriminative model, while LLMs are generative models. They have different advantages. For example, LLM is better at generating coherent text, and our CohEval is better at multi-choice tasks. In fact, on TimeTravel, we use MCMC to make our CohEval applicable to generating tasks. Therefore, the gap between our method and LLM has been magnified. On discriminative tasks, although our model is not as good as ChatGPT, it outperforms the smaller ChatGLM on most multi-choice tasks. On the other hand, it is inherently unfair to directly compare small models with LLMs, as large models are obtained with massive resources, e.g., data, hardware, funding, etc. Due to resource limitations, our method is not as good as ChatGPT, but it is superior to ChatGLM, which also indicates that our method is valuable in low-resource scenarios. With sufficient computational resources, we can use a larger backbone and more data for training, which is expected to yield better results. We leave this in future works.

Ethical Considerations

This work does not involve any sensitive data, but only crowd-sourced datasets released in previous works, including RocStories (Mostafazadeh et al., 2016), COPA (Roemmele et al., 2011), e-Care (Du et al., 2022a), αNLI (Bhagavatula et al., 2020), Cloze (Mostafazadeh et al., 2016), Swag (Zellers et al., 2018), HellaSwag (Zellers et al., 2019), and TimeTravel (Qin et al., 2019). We believe that our research work meets the ethics of ACL.

References

C. Bhagavatula, R. L. Bras, C. Malaviya, K. Sakaguchi, A. Holtzman, H. Rashkin, D. Downey, W. T. Yih, and Y. Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Deng Cai, Yizhe Zhang, Yichen Huang, Wai Lam, and Bill Dolan. 2020. Narrative incoherence detection. *arXiv preprint arXiv:2012.11157*.

Eugene Charniak. 1972. *Toward a model of children's story comprehension.* Ph.D. thesis, Massachusetts Institute of Technology.

J. Chen, C. Gan, S. Cheng, H. Zhou, Y. Xiao, and L. Li. 2021. Unsupervised editing for counterfactual stories.

J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. 2018a. Bert: Pre-training of deep bidirectional transformers for language understanding.

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

L. Du, X. Ding, K. Xiong, T. Liu, and B. Qin. 2022a. e-care: a new dataset for exploring explainable causal reasoning.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022b. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

T. Gao, X. Yao, and D. Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, Montréal, Canada. Association for Computational Linguistics.

Changying Hao, Liang Pang, Yanyan Lan, Yan Wang, Jiafeng Guo, and Xueqi Cheng. 2021. Sketch and customize: A counterfactual story generator. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12955–12962.

Daniel D Hutto. 2015. Narrative understanding. In *The Routledge companion to philosophy of literature*, pages 291–301. Routledge.

Prathyusha Jwalapuram, Shafiq Joty, and Xiang Lin. 2021. Rethinking self-supervision objectives for generalizable coherence modeling. *arXiv preprint arXiv:2110.07198*.

Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus. 2020. Hard negative mixing for contrastive learning.

Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. Rankgen: Improving text generation with large ranking models. *arXiv preprint arXiv:2205.09726*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *Computer Science*.

C. C. Lin, A. Jaech, X. Li, M. R. Gormley, and J. Eisner. 2020a. Limitations of autoregressive models and their alternatives.

Shih-Ting Lin, Nathanael Chambers, and Greg Durrett. 2020b. Conditional generation of temporally-ordered event sequences. *arXiv preprint arXiv:2012.15786*.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Zhiyi Luo, Yuchen Sha, Kenny Q Zhu, Seung-won Hwang, and Zhongyuan Wang. 2016. Commonsense causal reasoning between short texts. In *Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Satya N Majumdar and Henri Orland. 2015. Effective langevin equations for constrained stochastic processes. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(6):P06039.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764*.

N. Miao, H. Zhou, L. Mou, R. Yan, and L. Li. 2018. Cgmh: Constrained sentence generation by metropolis-hastings sampling.

A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. 2017. Working hard to know your neighbor's margins:local descriptor learning loss.

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*.

OpenAI. (2023). ChatGPT (gpt-3.5-turbo) [Large language model]. Https://chat.openai.com/chat.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation.

L. Qin, V. Shwartz, P. West, C. Bhagavatula, and Y. Choi. 2020. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. 2019. Counterfactual story reasoning and generation. *arXiv preprint arXiv:1909.04076*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Daniel Revuz and Marc Yor. 2013. *Continuous martingales and Brownian motion*, volume 293. Springer Science & Business Media.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Yui Uehara, Tatsuya Ishigaki, Kasumi Aoki, Hiroshi Noji, Keiichi Goshima, Ichiro Kobayashi, Hiroya Takamura, and Yusuke Miyao. 2020. Learning with contrastive examples for data-to-text generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2352–2362.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Chaoqi Wang, Adish Singla, and Yuxin Chen. 2021. Teaching an active learner with contrastive examples. *Advances in Neural Information Processing Systems*, 34:17968–17980.

Jian Wang, Dongding Lin, and Wenjie Li. 2023. Dialogue planning via brownian bridge stochastic process for goal-directed proactive dialogue. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 370–387, Toronto, Canada. Association for Computational Linguistics.

Rose E Wang, Esin Durmus, Noah Goodman, and Tatsunori Hashimoto. 2022. Language modeling via stochastic processes. *arXiv preprint arXiv:2203.11370*.

Terry Winograd. 1972. Understanding natural language. *Cognitive psychology*, 3(1):1–191.

C. Y. Wu, R. Manmatha, A. J. Smola, and Philipp Krhenbühl. 2017. Sampling matters in deep embedding learning. *IEEE*.

Yuqiang Xie, Yue Hu, Luxi Xing, Chunhui Wang, Yong Hu, Xiangpeng Wei, and Yajing Sun. 2020. Enhancing pre-trained language models by self-supervised learning for story cloze test. In *Knowledge Science, Engineering and Management: 13th International Conference, KSEM 2020, Hangzhou, China, August 28–30, 2020, Proceedings, Part I 13*, pages 271–279. Springer.

Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. 2020. Hard negative examples are hard, but useful. page 126–142, Berlin, Heidelberg. Springer-Verlag.

Wenlin Yao and Ruihong Huang. 2018. Temporal event knowledge acquisition via identifying narratives. *arXiv preprint arXiv:1805.10956*.

R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference.

R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. 2019. Hellaswag: Can a machine really finish your sentence?

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Shaofeng Zhang, Meng Liu, Junchi Yan, Hengrui Zhang, Lingxiao Huang, Xiaokang Yang, and Pinyan Lu. 2022. M-mix: Generating hard negatives via multi-sample mixing for contrastive learning. KDD '22, page 2461–2470, New York, NY, USA. Association for Computing Machinery.

T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. 2019. Bertscore: Evaluating text generation with bert.

Yucheng Zhou, Xiubo Geng, Tao Shen, Guodong Long, and Daxin Jiang. 2022a. Eventbert: A pre-trained model for event correlation reasoning. In *Proceedings of the ACM Web Conference 2022*, pages 850–859.

Yucheng Zhou, Tao Shen, Xiubo Geng, Guodong Long, and Daxin Jiang. 2022b. Claret: Pre-training a correlation-aware context-to-event transformer for event-centric generation and classification. *arXiv preprint arXiv:2203.02225*.

PLMs	Fluency (↓)	ENTScore (↑)
GPT2	2.8	58.2
T5	3.3	52.2
BART	3.4	66.7

Table 9: Impact of different backbones for contrastive narratives generation.

A Training Details about the Event Encoder

The event encoder is a nonlinear mapping from raw input space to latent space, $f_\theta : \mathcal{X} \rightarrow \mathcal{Z}$. Consider a set of triplet observations, (x_1, x_2, x_3), the goal is to ensure that $f_\theta(x_1), f_\theta(x_2), f_\theta(x_3)$ follow the Brownian bridge transition density in Equation 1. Following (Wang et al., 2022), we ensure this using a contrastive objective. Formally, given a narrative event sequences, $S = \{e_0, \cdots, e_4\}$, we draw batches consisting of randomly sampled positive triplets e_0, e_t, e_T where $0 < t < T : \mathcal{B} = \{(e_0, e_t, e_T)\}$. Note that we use indices $0, t, T$ to denote the start, middle, and end points of a Brownian bridge, but these do not correspond to strictly sampling the first, middle, and last events of a narrative story. The encoder is optimized by,

$$\mathcal{L}_f = -\log \frac{\exp(d(e_0, e_t, e_T; f_\theta))}{\sum\limits_{(e_0, e_{t'}, e_T) \in \mathcal{B}} \exp(d(e_0, e_{t'}, e_T; f_\theta))} \tag{10}$$
$$d(e_0, e_t, e_T; f_\theta) = -\frac{1}{2\sigma^2} ||f_\theta(e_t) - \mu||_2^2,$$

where μ and σ^2 are the mean and variance in Equation 1. As suggested by (Wang et al., 2022), we freeze the BART and add a non-linear layer that converts the BART output to a latent vector. The size of the latent space is set to 64 by default.

B Impact of Different Backbones for Generating Contrastive Narratives

We conduct a preliminary study on the influence of different backbones, including GPT2 (Radford et al., 2019) and T5 (Raffel et al., 2020), and BART (Lewis et al., 2019), for generating contrastive narratives. We use Fluency and ENTScore as metrics. Fluency evaluates whether the generated text is a fluent text sequence. We use off-the-shelf GPT2 to calculate Fluency. ENTScore evaluates the coherence of the generated stories. We randomly sample 2000 examples that do not exist in training for evaluation. We calculate the average result. As shown in Table 9, GPT2 is good at generating more fluent text, and BART generates more coherent text. A possible reason is that the contrastive narrative generation is more compatible with BART's pretraining task, e.g., masked auto-encoding. Finally, we choose BART as the backbone.

C Criteria for Filtering Low-Quality Candidates

For each positive narrative, we generate 200 candidates. In practice, we observe that the generator may produce incoherent or duplicate candidates. Therefore, we set several rules to filter low-quality items. We first use our event-level replacement strategy to train the base evaluator M_{ER}. We use M_{ER} to filter items whose coherence scores are smaller than a threshold (empirically set to 0). Next, for each candidate, we calculate its text similarity with the remaining candidates. We gradually discard the candidates with the highest similarity until there are 100 remaining. When training Coheval, we select N top-ranked candidates according to their coherence scores for synthesizing negative samples.

D Statistics and Experimental Details

Statistics Table 12 shows the statistics of the used datasets.

Experimental Details For training the contrastive narratives generator, we use BART-base as the backbone. Batch-size is set to 16. We use the AdamW optimizer. lr is set to 5e-5. Weight-decay is set to 1e-4. We train the generator with 10 epochs and linearly decrease the lr to zero with no warmup. When the generation phase, we kept the $N = 60$ most qualified contrastive narratives for creating negative examples. For training our *CohEval*, we adopt RoBERTa-large as the backbone. We train our model for 5 epochs, and then evaluate it on downstream tasks. We set batch-size to 1 and gradient-accumulation-steps to 16. For each positive example, we sample 15 negative examples for contrastive training. lr is set to 5e-5. Weight-decay is set to 1e-4. We use the AdamW optimizer and linearly decrease the lr to zero with a 10% warmup ratio. The random seed is set to 42 for all experiments. All experiments are performed on a Ubuntu server with 4×RTX2080Ti GPUs.

Tasks	Prompt
HellaSwag	Multi-choice Task: Given a context event, select the most reasonable subsequent event from the following four choices. Here is one example: ### Context event: The man examines the instrument in his hand. Please select the most reasonable subsequent event from the following four choices. Choice1: The person studies a picture of the man playing the violin. Choice2: The person holds up the violin to his chin and gets ready. Choice3: The person stops to speak to the camera again. Choice4: The person puts his arm around the man and backs away. Between Choice1, Choice2, Choice3 and Choice4, the correct one is: Choice2 ### Now, given the following example, please select the correct answer. No further explanation is required. Context event: {context} Please select the most reasonable subsequent event from the following four choices. Choice1: {op1} Choice2: {op2} Choice3: {op3} Choice4: {op4} Between Choice1, Choice2, Choice3 and Choice4, the correct one is:
TimeTravel	Each story contains 5 sentences, where the first two sentences are the story premise, and the last 3 sentences are the story ending. I will apply subtle a perturbation to the second sentence, making the first two sentences a counterfactual story premise. Due to the slight perturbation, the counterfactual premise is very similar to the original premise, with only some words being different. According to the original story and the counterfactual story premise, you are required to predict the counterfactual story ending. Note that the counterfactual story ending should be similar to the original story ending, as well as be coherent with the counterfactual story premise. Here is one example: ### <Original 5-sentences story> 1. Bella wanted to cook some spaghetti and meatballs. 2. She discovered she had no pasta noodles. 3. She found a recipe online that used spaghetti squash instead. 4. Bella luckily had a spaghetti squash on hand. 5. She was surprised to find the spaghetti and meatballs delicious! <Counterfactual story premise> 1. Bella wanted to cook some spaghetti and meatballs. 2. She realized she didn't have the time to make it properly so she changed made an omelette instead. <Counterfactual story ending> 3. She found a recipe online that used egg whites instead. 4. Bell luckily had many eggs on hand. 5. She was surprised to find the egg white omelette delicious! <END> ### Now, given the following example, please write the counterfactual story ending. There should be only three sentences at the counterfactual story ending. Ending with <END>. <Original 5-sentences story> {original_story} <Counterfactual story premise> {counterfactual_premise} <Counterfactual story ending>
Contrastive Narratives Generation	Contrastive story generation: You will see a five-sentence story. Now let's fix the first and last sentences, and you need to generate another middle three sentences to make the resulted five sentences form a different story. Ensure that your generation is similar to the original story and conveys different semantics. Here is one example: ### <Original Story> 1. Sam and John went out to play some ultimate Frisbee one day. 2. Upon arrival at the field, there was a pickup game of football going. 3. Sam approached them and asked them to let him and John play as well. 4. After a few minutes talk, they agreed and everyone played for a bit. 5. Then they all went home. <The fixed first and last sentences> 1. Sam and John went out to play some ultimate Frisbee one day. 5. Then they all went home. <Generated middle 3 sentences> 2. Upon arrival at the field they found it deserted. 3. Sam and John played on the field by themselves. 4. After a few minutes, they agreed they were bored. ### Now, given the following input, generate the middle three sentences. <Original Story> {original_story} <The fixed first and last sentences> {first_last_events} <Generated middle 3 sentences>

Table 10: The prompts used for different tasks.

Premise (z)	Alec's daughter wanted more blocks to play with.
Initial (x)	Alec figured that blocks would develop her scientific mind.
Original Ending (y)	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 (x')	Alec couldn't afford to buy new blocks for his daughter.
Edited Ending (y')	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.

Table 11: An examples from TimeTravel.

	COPA	e-Care	αNLI	Cloze	Swag	HS.	TimeT.
#numAns	2	2	2	2	4	4	N/A
#numVal	500	2132	1532	1871	20006	10041	1871
#numTest	500	N/A	3059	1871	N/A	N/A	1871

Table 12: The statistics of the used datasets. #numVal and #numTest denotes the number of samples in the val and test set. #numAns denotes the size of the answer set of multi-choice datasets. HS. and TimeT. denotes HellaSwag and TimeTravel, respectively.

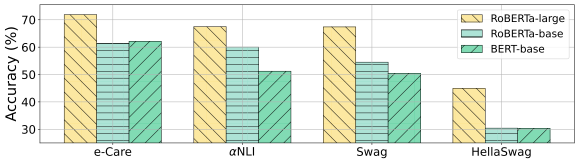


Figure 5: Results under different backbones for narrative coherence learning.

E Different Backbones for Narrative Coherence Learning

We additionally build our method on the BERT-base (Devlin et al., 2018b) and RoBERT-base backbones, as shown in Figure 5. RoBERTa-base has a better performance than BERT-base, and the RoBERTa-large tends to have a better result than RoBERTa-base. However, due to the limitation of computing resources, we are not able to evaluate our method under larger pre-trained models.

F The Prompts for Different Tasks

The multi-choice datasets have a similar format except for the number of choices. For simplicity, we take the HellaSwag dataset as an example. An example and its corresponding instruction are shown in Table 10. For other multi-choice datasets, we use a similar format for evaluation.

G The Non-autoregressive Generation Process on TimeTravel

TimeTravel is a counterfactual story generation dataset. A story is defined as a five-sentence text $\{z, x, y\}$, where the first sentence z is the premise, the second sentence x is the original condition, and the last three sentences constitute the original ending, abbreviated as y. After given a counterfactual condition denoted as x', the task requires revising the original ending y into a counterfactual ending y' which minimally modifies the original one and regains narrative coherency to the counterfactual condition. An example is shown in Table 11.

Existing EDUCAT (Miao et al., 2018; Chen et al., 2021) adopts the Markov chain Monte Carlo (MCMC) sampling process to this task. EDUCAT directly samples from the sentence space with three local operations: token replacement, deletion and insertion. During sampling, after an edit position is found, the operation is randomly chosen with equal probability. Finally, the proposed new sentence will either be accepted or rejected according to the acceptance rate computed by desired properties $\pi(y)$. The above process is repeated till convergence.

The stationary distribution $\pi(y)$ in EDUCAT is defined as the product of the fluency score and the coherence score as follows:

$$\pi(y) = \mathcal{X}_{LM}(y) \cdot \mathcal{X}_{Coh}(y), \tag{11}$$

where the fluency score $\mathcal{X}_{LM}(y)$ is the probability of the generated ending based on GPT2. The coherence score $\mathcal{X}_{Coh}(y)$ is defined by:

$$\mathcal{X}_{Coh}(y') = \frac{P_{Coh}(Y = y'|z, x')}{P_{Coh}(Y = y'|z, x)}, \tag{12}$$

where $P_{Coh}(\cdot)$ is the conditional probability calculated by GPT2. This definition encourages the generated y' to be more coherent to x' instead of x.

Following EDUCAT, we define the stationary distribution $\pi(y)$ as Equation 11. The difference is that we replace $\mathcal{X}_{Coh}(y)$ with our CohEval:

$$\mathcal{X}_{Coh}(y) = \text{CohEval}([z; x; y']), \tag{13}$$

where [;] denotes the concatenation. Same as EDUCAT, we run our model and its variants for 100 steps for fairness.

H The Correlation between Automatic Metrics and Human Ratings

Metrics	Pearson's r	Kendall's τ
ENTScore	0.25	0.24
CohEval	0.20	0.18

Table 13: The correlation between automatic metrics, e.g., ENTScore and CohEval, and human ratings. All of these numbers are statistically significant at p < 0.01.

Table 13 shows the correlation between automatic metrics, including ENTScore, CohEval, and human ratings in coherence. All results show a positive correlation. The result of our CohEval is similar to that of ENTScore. Notice that ENTScore is trained with human-labeled counterfactual data, while our CohEval is trained in a self-supervised manner.

I Details and Results for Mixing-up in Latent Space

The mixup strategy creates negative examples via mixing-up a positive X and several counterparts $\{X_c^k\}_{k=1}^K$ in the latent space:

$$\begin{aligned} \mathbf{h}^+ &= \text{RoBERTa}(X) \\ \mathbf{h}_c^k &= \text{RoBERTa}(X_c^k), \\ \bar{\mathbf{h}}^k &= \alpha_k \mathbf{h}^+ + (1-\alpha_k)\mathbf{h}_c^k, \\ \alpha_k &\sim Uniform[0,1]. \end{aligned} \tag{14}$$

Then, the loss is:

$$\begin{aligned} s^+ &= \mathbf{W}_c^T \mathbf{h}^+, \\ \bar{s}^k &= \mathbf{W}_c^T \bar{\mathbf{h}}^k, \\ \mathcal{L}_M &= -\frac{1}{|\mathcal{D}^+|}\sum_{\mathcal{D}^+} \log \frac{\exp(s^+)}{\exp(s^+) + \sum_{k=1}^K \exp(\bar{s}^k)}. \end{aligned} \tag{15}$$

The experiment setting details are the same as those used in Appendix D.

J Error Cases when Creating Negative Examples

The most common error in event-level replacement is that the sampled event $\bar{e}$ from Q^e is especially similar to the original e, or is the paraphrase of the original e, as shown in Table 14, Case #1. The most common mistake in cross strategy is that the contrastive variant and the original story describe different actions for the same purpose, resulting in the false negative. An example is shown in Table 14, Case #2.

Overall, the proportion of errors is relatively low.

	Event-level replacement
#1	**Original story**: Rod was ironing his clothes. His iron is very old. The iron left a burn mark on his favorite shirt! Rod got very angry. He bought a new iron.
	Created negative: She even ironed her sheets! His iron is very old. The iron left a burn mark on his favorite shirt! Rod got very angry. He bought a new iron.
	Crisscrossing
#2	**Original story**: Katie needed to cross the street by herself. Being seven, she hadn't done it by herself before. She looked left then right. The coast was clear. She crossed the street successfully and was full of pride.
	Contrastive example: Katie needed to cross the street by herself. Being seven she knew it was possible. The nearest crosswalks were just behind her. She held her breath and did her best. She crossed the street successfully and was full of pride.
	Created negative: Katie needed to cross the street by herself. Being seven, she hadn't done it by herself before. The nearest crosswalks were just behind her. She held her breath and did her best. She crossed the street successfully and was full of pride.

Table 14: Error cases when creating negatives.

K Details of Obtaining Samples in t-NSE Visualization

In TimeTravel, each example consists of an original story (z, x, y) and a counterfactual story (z, x', y'), where y' is similar to y. Motivated by (Chen et al., 2021), we obtain positive and negative samples from the perspective of natural language inference, i.e., the original context (z, x) entails by y but contradicts with y', and the counterfactual context (z, x') entails by y' but contradicts with y. Because y is similar to y', (z, x, y') and (z, x', y) tend to be hard negatives. Based on the test set of TimeTravel, we obtain 3742 positive examples and 3742 negative examples. Then, we use t-SNE to visualize representations of the examples that are encoded by different models.

L Case Study

Table 15 presents a case study for the task of TimeTravel. The counterfactual endings generated by ChatGLM2 and ChatGPT are very different from the original ending, which conflicts with the minimal-edits requirement of the task. On the contrary, based on the MCMC-sampling, our method produces the counterfactual ending, which is similar to the original ending, as well as coherent to the counterfactual condition.

Table 16 presents a case study for the task of contrastive narratives generation. In the case #1, our method shifts the topic of accent to personality. In the case #2, our method exchanges the opinions

#1	Original context	Tim was mad. He saw roaches in his house.
	Original ending	He bought some cheap bug spray. However, it did not get rid of them. Tim called an exterminator.
	Counterfactual context	Tim was mad. He saw rats in his house.
	GhatGLM2	He found a rat trap and was able to catch them. Tim was relieved that the problem had been solved. He called an exterminator to make sure they were gone for good.
	ChatGPT	He bought some cheap rat traps. However, they did not catch any rats. Tim called a professional pest control service.
	CohEval(ours)	He bought rat poison. And it did not get rid of the rats. So he hired an exterminator.
#2	Original context	Jim worked at the local coffee shop. He was the best barista in town.
	Original ending	He always made sure his customers had the best drinks. Jim's boss saw him working one day and was impressed. Jim got a promotion for doing such a good job.
	Counterfactual context	Jim worked at the local coffee shop. He was the worst barista in town.
	GhatGLM2	He found a recipe online to make better drinks. Jim decided to try it out and impressed his boss. Jim got a raise for his excellent work.
	ChatGPT	He always made sure his customers had the worst drinks. Jim's boss saw him working one day and was disappointed. Jim got fired for doing such a bad job.
	CohEval(ours)	He always made sure his customers had the wrong drinks. Jim's boss saw him that night and was furious. He got a beating for doing a bad job.

Table 15: Some cases with the generated text by different models. The cases are from the test set of TimeTravel.

of two participants. Due to the limited changes, the generated story is very similar to the original story and meets the requirements for contrastive narratives. On the contrary, the middle events generated by ChatGLM2 and ChatGPT show a significant difference from that of the original story, as a result, the synthesized negative samples should be easily distinguished.

#1	Original story	Fanny is new at school. She is from England. Many kids do not understand her accent. This made Fanny sad. Fanny is having a difficult time making friends.
	ChatGLM2	Fanny is new at school. One of the kids at school does understand her accent, and they become her friends. Fanny meets a new student at her lunch table who helps her feel more comfortable. Fanny learns to embrace her accent and makes many new friends at school. Fanny is having a difficult time making friends.
	ChatGPT	Fanny is new at school. During lunchtime, Fanny sat alone at a table in the cafeteria. Some kids approached Fanny and asked her where she was from. Fanny tried her best to explain her accent, but they still had trouble understanding her. Fanny is having a difficult time making friends.
	CohEval(ours)	Fanny is new at school. She has just graduated from college. She is very shy. Fanny is not very outgoing. Fanny is having a difficult time making friends.
#2	Original story	Morgan and Timothy were having an argument. Morgan thought that all of The Beatles' music was good. Timothy thought their early stuff was not very good. Morgan made her case by simply playing a few songs. Timothy realized Morgan was probably right about the music.
	ChatGLM2	Morgan and Timothy were having an argument. Timothy tried to reason with Morgan, but she was too set in her opinion. Morgan refused to back down and continued to play her favorite songs. Timothy was left feeling frustrated, but eventually came to accept Morgan's love for her music. Timothy realized Morgan was probably right about the music.
	ChatGPT	Morgan and Timothy were having an argument. Morgan defended her stance by pointing out the innovative songwriting and harmonies in their later albums. Timothy reluctantly agreed to listen to a few songs from their later discography. As the music played, Timothy's opinion started to shift, realizing the depth and creativity in The Beatles' later work. Timothy realized Morgan was probably right about the music.
	CohEval(ours)	Morgan and Timothy were having an argument. Morgan said the Beatles' music was terrible. Timothy said their early music was good. Morgan disagreed. Timothy realized Morgan was probably right about the music.

Table 16: Case study for contrastive narratives generation.