# Long Text Generation by Modeling Sentence-Level and Discourse-Level Coherence

**Jian Guan**[1], **Xiaoxi Mao**[2], **Changjie Fan**[2], **Zitao Liu**[3], **Wenbiao Ding**[3]
**and Minlie Huang**[1*]

[1]The CoAI group, DCST; [1]Institute for Artificial Intelligence; [1]State Key Lab of Intelligent Technology and Systems;
[1]Beijing National Research Center for Information Science and Technology;
[1]Tsinghua University, Beijing 100084, China. [2]Netease Fuxi AI Lab. [3]TAL Education Group.

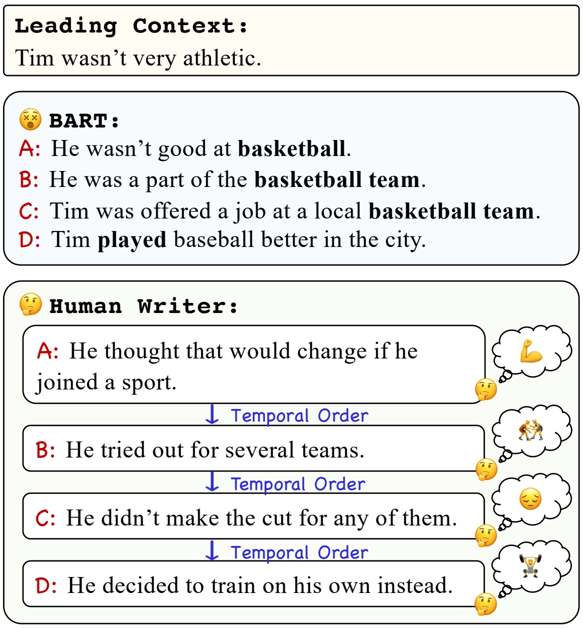
j-guan19@mails.tsinghua.edu.cn, {maoxiaoxi,fanchangjie}@corp.netease.com,
zitao.jerry.liu@gmail.com, dingwenbiao@100tal.com, aihuang@tsinghua.edu.cn

## Abstract

Generating long and coherent text is an important but challenging task, particularly for open-ended language generation tasks such as story generation. Despite the success in modeling intra-sentence coherence, existing generation models (e.g., BART) still struggle to maintain a coherent event sequence throughout the generated text. We conjecture that this is because of the difficulty for the decoder to capture the high-level semantics and discourse structures in the context beyond token-level co-occurrence. In this paper, we propose a long text generation model, which can represent the prefix sentences at sentence level and discourse level in the decoding process. To this end, we propose two pretraining objectives to learn the representations by predicting inter-sentence semantic similarity and distinguishing between normal and shuffled sentence orders. Extensive experiments show that our model can generate more coherent texts than state-of-the-art baselines.

## 1 Introduction

The ability to generate coherent long texts plays an important role in many natural language generation (NLG) applications, particularly for open-ended language generation tasks such as story generation, namely generating a reasonable story from a prompt or a leading context. While existing generation models (Fan et al., 2018; Radford et al., 2019) can generate texts with good intra-sentence coherence, it is still difficult to plan a coherent plot throughout the text, even when using the powerful pretrained models, as illustrated in Figure 1.



**Leading Context:**
Tim wasn't very athletic.

**BART:**
A: He wasn't good at **basketball**.
B: He was a part of the **basketball team**.
C: Tim was offered a job at a local **basketball team**.
D: Tim **played** baseball better in the city.

**Human Writer:**
A: He thought that would change if he joined a sport.
B: He tried out for several teams.
C: He didn't make the cut for any of them.
D: He decided to train on his own instead.

Figure 1: Story examples written by the fine-tuned BART model (Lewis et al., 2020) and a human writer given the same leading context from ROCStories (Mostafazadeh et al., 2016). The generated story by BART suffers from severe incoherence issue in spite of some related concepts (in **bold**). In comparison, the human writer can write a coherent story because they fully consider the context semantics and discourse relations (e.g., the temporal order) among the sentences.

Pretrained generation models have shown state-of-the-art performance on various NLG tasks such as summarization and translation (Radford et al., 2019; Lewis et al., 2020). However, such tasks provide sufficient source information in the input for generating desired texts, while open-ended generation tasks require expanding reasonable plots from very limited input information (Guan et al., 2020). As exemplified in Figure 1, we observe severe issues of incoherence when applying BART for story generation. Although BART performs reasonably well at generating some concepts related to the context (e.g., *"basketball", "player"*), they are used incoherently in the generated texts, which is manifested in repetitive plots (e.g., the sentences *B* and *C*), unrelated events (e.g., *"played baseball*

[*] Corresponding author

*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*
*and the 11th International Joint Conference on Natural Language Processing*, pages 6379–6393
August 1–6, 2021. ©2021 Association for Computational Linguistics

*better"*) and conflicting logic (e.g., *"not good at basketball"* but *"in the basketball team"*). These issues are also commonly observed in other NLG models (Holtzman et al., 2020; Guan and Huang, 2020). We argue that existing models are rarely trained beyond the token-level co-occurrence, and therefore they can easily generate related concepts but do not arrange them reasonably. In contrast, human writers always first fully understand the semantics (e.g., some key events such as *"try out", "not make the cut"*) and the discourse relations (e.g., temporal orders) among the already written sentences before deciding the following content. In this way, the writers can write coherent stories even with few related concepts, as shown in Figure 1. Therefore, it is important for subsequent generation to capture high-level features in the context.

In this paper, we propose HINT, *a generation model equipped with **HI**gh-level representations for loNg **T**ext generation*. Typical generative models usually train a left-to-right decoder by next word prediction based on the attention to all the prefix words. In order to encourage the model to capture high-level features, we extend the decoder to represent the prefix information at sentence level and discourse level, respectively, with special tokens which are inserted at the end of each sentence. To effectively learn the representations, we propose two pretraining objectives including: (a) *semantic similarity prediction*, which requires predicting the inter-sentence similarity using the sentence-level representation, with the powerful sentence understanding model SentenceBERT (Reimers and Gurevych, 2019) as the teacher model; and (b) *sentence order discrimination*, which requires distinguishing between the normal and shuffled sentence orders using the discourse-level representation. The objectives are designed to help the decoder capture the semantics and discourse structure of the prefix, which can benefit modeling the long-range coherence when generating long texts. We summarize our contributions in two folds:

**I.** We propose a generation model named HINT for long text generation. HINT derives high-level representations for each decoded sentence to model the long-range coherence. We adopt two pretraining objectives called similarity prediction and order discrimination to learn the representations at sentence level and discourse level, respectively.

**II.** We conduct extensive experiments on commonsense story and fiction generation tasks. Results show that HINT can learn meaningful high-level representations and generate more coherent long texts than baselines.[1]

## 2 Related Works

**Long Text Generation** Recent studies tackle the incoherence problem in long text generation from the following perspectives. Li et al. (2015) adopted a hierarchical RNN-based decoder to learn the sentence representation but without any external supervision. Shao et al. (2017) proposed a self-attention mechanism to attend on the prefix by appending it to the RNN-based encoder, which is a similar idea with the vanilla Transformer (Vaswani et al., 2017). However, the token-level self-attention mechanism still struggles to model high-level dependency in the context. Recent works proposed several multi-step generation models (Fan et al., 2018; Yao et al., 2019; Shao et al., 2019; Tan et al., 2020; Goldfarb-Tarrant et al., 2020), which first plan high-level sketches and then generate texts from the sketches. However, the lack of exposure to degenerate sketches may impair the generation performance since the models are only trained on sketches constructed from golden truth texts (Tan et al., 2020). Another line is to incorporate external knowledge into generation especially for commonsense story generation (Guan et al., 2020; Xu et al., 2020). However, the methods may not be always effective for other types of generation tasks. Guan et al. (2020) also required the decoder to distinguish true texts from negative samples to alleviate potential issues such as repetition. But the classification objective does not provide explicit guidance for generation at each step. Therefore, the coherence of language generation is still an open problem.

**High-Level Language Representation** Significant advances have been witnessed in many NLP tasks with pretrained contextualized representation (Peters et al., 2018; Devlin et al., 2019). However, most models were limited on token-level representation learning, which is not enough for capturing the hierarchical structure of natural language texts (Ribeiro et al., 2020). Several works have tried to learn high-level representation. Skip-Thought vectors (Kiros et al., 2015) learned to encode a sentence by reconstructing its neighboring sentences. HLSTM (Yang et al., 2016) considered a

[1] The codes are available at `https://github.com/thu-coai/HINT`

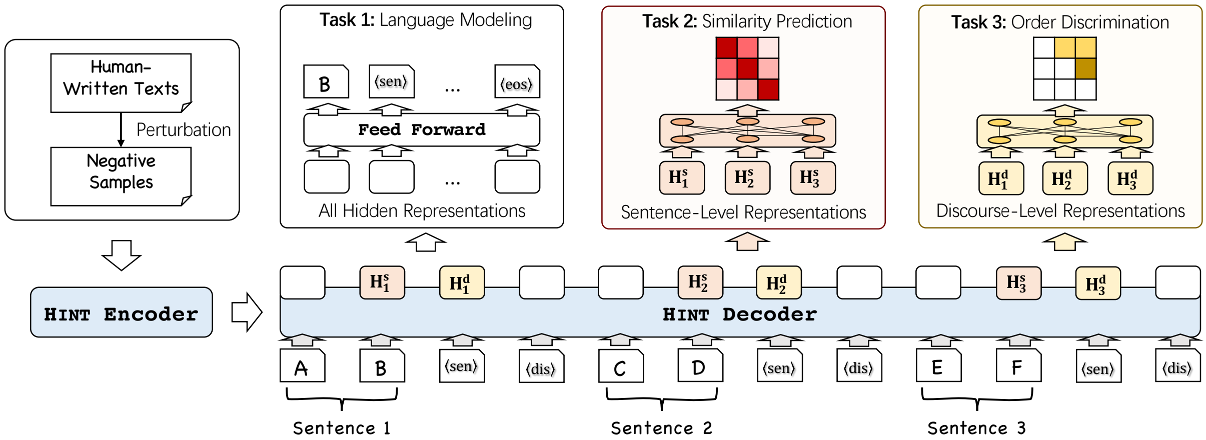


Figure 2: Model overview of HINT, which is pretrained to predict the next token (Task 1), predict inter-sentence semantic similarity with the sentence-level representations (Task 2), and distinguish between normal and shuffled sentence orders with the discourse-level representations (Task 3) based on the human-written texts and auto-constructed negative samples.

hierarchical LSTM-based encoder to learn the contextualized sentence representation by downstream classification. HIBERT (Zhang et al., 2019) incorporated the hierarchical architecture to BERT (Devlin et al., 2019) and learned sentence representation by recovering masked sentences. Sentence-BERT (Reimers and Gurevych, 2019) derived sentence representation by fine-tuning BERT for natural language inference. CONPONO (Iter et al., 2020) and SLM (Lee et al., 2020) further trained BERT to understand relations among sentences at discourse level by distance prediction and sentence unshuffling, respectively. However, all these models focused on enhancing the representation of encoders for language understanding, while improving decoders by high-level representation for long text generation is yet to be well investigated.

## 3 Methodology

### 3.1 Task Definition and Model Overview

Our task can be defined as follows: given an input $X = (x_1, x_2, \cdots, x_m)$ (e.g., a beginning or a prompt), the model should generate a multi-sentence text $Y = (y_1, y_2, \cdots, y_n)$ with a coherent plot (each $x_i$ or $y_i$ is a token). To tackle the problem, the conventional generation models such as BART commonly employ a bidirectional encoder and a left-to-right decoder to minimize the negative log-likelihood $\mathcal{L}_{LM}$ of human-written texts:

$$\begin{aligned} \mathcal{L}_{LM} &= -\sum_{t=1}^{n} \log P(y_t|y_{<t}, X), && (1)\\ P(y_t|y_{<t}, X) &= \text{softmax}(\mathbf{H}_t \boldsymbol{W} + \boldsymbol{b}), && (2)\\ \mathbf{H}_t &= \texttt{Decoder}(y_{<t}, \{\mathbf{S}_i\}_{i=1}^{m}), && (3)\\ \{\mathbf{S}_i\}_{i=1}^{m} &= \texttt{Encoder}(X), && (4) \end{aligned}$$

where $\mathbf{H}_t$ is the decoder's hidden state at the $t$-th position computed from the context (i.e., the prefix $y_{<t}$ and the input $X$), and $\mathbf{S}_i$ is the contextualized representation of $x_i$ acquired from the encoder, $\boldsymbol{W}$ and $\boldsymbol{b}$ are trainable parameters.

However, as aforementioned, the models often generate incoherent texts due to the decoder's inability to capture high-level features of the prefix sentences. Therefore, we extend the decoder with high-level representations to gather the prefix information. Specifically, we split the human-written texts into sequential sentences and add special tokens at the end of each sentence, which will be used to aggregate their respective semantics and their discourse relations with one another during decoding. To this end, we devise two pretraining tasks besides the standard language modeling objective, including similarity prediction and order discrimination to learn the sentence-level and discourse-level representations, respectively, as Figure 2 shows. Although we only consider sentence as segments in this work, our method can be easily extended to other syntactic levels such as phrases or paragraphs.

### 3.2 Sentence-Level Representation

Assume that the target text $Y$ consists of $K$ sentences, denoted from $Y_1$ to $Y_K$ (e.g., `AB` and `CD` in Figure 2). We insert a special sentence token, $\langle$`sen`$\rangle$, at the end of every sentence in $Y$, which is designed to aggregate the semantics of each sentence. Let $\mathbf{H}_k^{\mathrm{s}}$ $(1 \leqslant k \leqslant K)$ denote the decoder's hidden state at the position where the $k$-th sentence token is the golden truth for next token prediction. We expect $\mathbf{H}_k^{\mathrm{s}}$ to be a meaningful sentence representation for $Y_k$, which means semantically similar sentences have close representations in the vector space. Since sentence representation has been well studied for language understanding with many powerful models such as SentenceBERT (Reimers and Gurevych, 2019), we propose to directly transfer their semantic knowledge for our sentence representation learning. Specifically, we require the HINT decoder to predict the similarity of any two sentences $Y_i$ and $Y_j$ only using the corresponding sentence representations $\mathbf{H}_i^{\mathrm{s}}$ and $\mathbf{H}_j^{\mathrm{s}}$, with the SentenceBERT similarity as the golden truth[2]. We do not directly learn the SentenceBERT representation for each sentence but the similarity score to avoid the discrepancy between different model bias. Furthermore, to alleviate the innate bias of SentenceBERT, we do not enforce HINT to exactly fit the golden similarity. Instead, it would be enough that the difference between the predicted score and the golden similarity is less than a margin $\Delta \in [0, 1]$. Formally, the loss function $\mathcal{L}_{Sen}$ for the similarity prediction task can be derived as follows:

$$\mathcal{L}_{Sen} = \frac{1}{K^2} \sum_{i=1}^{K} \sum_{j=1}^{K} \max(|p_{ij} - t_{ij}|, \Delta), \quad (5)$$

$$p_{ij} = \mathrm{sigmoid}(s_{ij} + s_{ji}), \quad (6)$$

$$s_{ij} = (\mathbf{H}_i^{\mathrm{s}})^{\mathrm{T}} \boldsymbol{W}^{\mathrm{s}} \mathbf{H}_j^{\mathrm{s}}, \quad (7)$$

where $t_{ij}$ is the golden similarity, $p_{ij}$ is the predicted similarity score, $s_{ij}$ is an intermediate variable to guarantee $p_{ij}$ is symmetric with respect to $i$ and $j$, $\boldsymbol{W}^{\mathrm{s}}$ is a trainable parameter to transform the representation space of HINT to that of SentenceBERT. The task explicitly exerts external supervision to learn the sentence-level representation, enhancing the ability of the HINT decoder to fully understand the semantics of prefix sentences.

### 3.3 Discourse-Level Representation

In analogy to the sentence-level representation learning, we also insert a special discourse token, $\langle$`dis`$\rangle$, after every sentence and the corresponding sentence token to gather the discourse information between different sentences. Let $\mathbf{H}_k^{\mathrm{d}}$ $(1 \leqslant k \leqslant K)$ denote the decoder's hidden state at the position where the $k$-th discourse token is the golden truth to be predicted. $\mathbf{H}_k^{\mathrm{d}}$ should be a meaningful representation which can be used to derive discourse relations with others (e.g., the $k$-th sentence precedes another one in terms of the temporal order). Previous work has shown that reconstructing the correct order from shuffled sentences helps understand the discourse relations (Lee et al., 2020). However, the unshuffling task is not directly applicable for NLG since the decoder should learn to dynamically model the discourse structure in the decoding process rather than wait until finishing decoding the whole text. Therefore, we propose to learn the discourse-level representation in a pair-wise manner by discriminating whether the order of two sentences is correct. Formally, we minimize the cross-entropy loss $\mathcal{L}_{Dis}$ as follows:

$$\mathcal{L}_{Dis} = \frac{2}{K(K-1)} \sum_{i=1}^{K} \sum_{j>i}^{K} l_{ij}, \quad (8)$$

$$l_{ij} = -o_{ij}\mathrm{log}q_{ij} - (1 - o_{ij})\mathrm{log}(1 - q_{ij}) \quad (9)$$

$$q_{ij} = \mathrm{sigmoid}\left((\mathbf{H}_i^{\mathrm{d}})^{\mathrm{T}} \boldsymbol{W}^{\mathrm{d}} \mathbf{H}_j^{\mathrm{d}}\right), \quad (10)$$

where $o_{ij}$ is the golden label (1 if $Y_i$ should precede $Y_j$, 0 otherwise), $q_{ij}$ is the predicted discrimination score, and $\boldsymbol{W}^{\mathrm{d}}$ is a trainable parameter. Compared with the sentence-level representation $\mathbf{H}_k^{\mathrm{s}}$ which aggregates the semantics of a single sentence, the discourse-level representation $\mathbf{H}_k^{\mathrm{d}}$ focuses more on the relationship with other sentences, thereby improving HINT's ability to capture the high-level features in both content and order.

### 3.4 Pretraining and Fine-tuning

To learn the high-level representations more effectively, we propose to augment the training corpus by automatically constructing negative samples from the human-written texts for pretraining. Specifically, for the order discrimination task, we randomly shuffle the sentences in human-written texts as negative samples. And for the similarity prediction task, besides the negative samples with shuffled sentences, we also randomly repeat a sentence, or substitute a sentence with another from

[2] The SentenceBERT similarity is computed as the cosine distance of two sentence embeddings which are derived by applying mean-pooling on the output vectors of SentenceBERT. And we normalize the results to $[0, 1]$ range by linear scaling.

other texts as negative samples. We expect the negative samples to help enhance the generalization ability of HINT during fine-tuning or inference. In summary, the overall loss function $\mathcal{L}_{Pre}$ for pretraining is computed as follows:

$$\mathcal{L}_{Pre} = \mathcal{L}_{LM} + \lambda_1 \mathcal{L}_{Dis} + \lambda_2 \mathcal{L}_{Sen}, \qquad (11)$$

where we optimize the language modeling objective $\mathcal{L}_{LM}$ only on the human-written texts, $\mathcal{L}_{Dis}$ on the human-written texts and the negative samples with shuffled sentences, and $\mathcal{L}_{Sen}$ on all the human-written texts and the negative samples. $\lambda_1$ and $\lambda_2$ are adjustable scale factors. By pretraining with the proposed two objectives, the decoder can better capture the semantics and discourse structures in the context. And during fine-tuning, we train HINT only with the language modeling objective.

## 4 Experiments

### 4.1 Implementation and Pretraining Dataset

Since our approach can adapt to all the generation models with auto-regressive decoders (e.g., GPT-2 (Radford et al., 2019), UniLM (Dong et al., 2019), etc.), we use BART as the base framework of HINT, which has been shown to have strong performance for long text generation (Goldfarb-Tarrant et al., 2020). And we also provide the performance of GPT-2 widely used in the literature. Due to the limited computational resources, we follow $\text{BART}_{\text{BASE}}$'s hyper-parameters and utilize the public pretrained checkpoint to initialize HINT. The batch size is set to 10 and the maximum sequence length is set to 512 for both the encoder and the decoder. The margin $\Delta$ in Equation 5 is set to 0.1 and we present the results with other settings of $\Delta$ in the appendix. Both the scale factors $\lambda_1$ and $\lambda_2$ in Equation 11 are set to 0.1.

We adopt BookCorpus (Zhu et al., 2015) as our pretraining dataset and split each text to sentences using NLTK (Bird and Loper, 2004). We create the training texts by taking a sentence as the input and the following ten sentences as the target output. Besides, we construct the same number of negative samples with the human-written texts. And it is evenly possible for a negative sample to be repeated, substituted or shuffled. We pretrain HINT on BookCorpus for 0.1M steps.

### 4.2 Fine-tuning Setting

We evaluate HINT on ROCStories (**ROC** for short) (Mostafazadeh et al., 2016) and WritingPrompts (**WP** for short) (Fan et al., 2018). ROC contains 98,162 five-sentence commonsense stories. We follow Guan et al. (2020) to delexicalize stories in ROC by masking all the names with special placeholders to achieve better generalization. WP originally contains 303,358 stories paired with writing prompts, which are usually unconstrained on writing topics. Considering that using too many examples for fine-tuning may weaken the influence of post-training, we randomly selected stories from the original validation set and test set of WP for the subsequent experiments. We regard the first sentence and the prompt as the input to generate a text for ROC and WP, respectively. And we only retain the first ten sentences (split using NLTK) of the texts in WP for fine-tuning. We present more details in Table 1. The batch size is set to 10/4 for ROC/WP, respectively. And other hyper-parameters are the same as the pretraining phase.

| Dataset | Input | Output | Train | Val | Test |
|---|---|---|---|---|---|
| **ROC** | 14.47 | 56.29 | 88,344 | 4,908 | 4,909 |
| **WP** | 30.02 | 185.65 | 26,758 | 2,000 | 2,000 |

Table 1: The average number of tokens in the **input** and **output** in the whole dataset, and the numbers of stories for **train**ing/**val**idation/**test**.

### 4.3 Baselines

We compared HINT with the following baselines:
**Seq2Seq:** It generates a text conditioned upon the input. For better performance, We implement the baseline by training BART from scratch on the downstream datasets without pretraining.
**Plan&Write:** It first plans a keyword sequence conditioned upon the input; and then generates a text based on the keywords (Yao et al., 2019). We implement the model based on the codes provided by the original paper.
**GPT-2** and **BART:** They are fine-tuned on the downstream datasets with the language modeling objective.
**BART-Post:** It is first post-trained on the pretraining dataset with the original pretraining objectives of BART (text infilling and sentence permutation) for the same number of steps with HINT; and then fine-tuned on the downstream datasets with the language modeling objective.
**BART-MTL:** The model is trained by fine-tuning BART on the downstream datasets with multi-task learning (MTL), including the language model-

ing objective and an auxiliary multi-label classification objective (Guan et al., 2020), which requires distinguishing human-written texts from auto-constructed negative samples.

Furthermore, we conduct ablation tests by removing the proposed components respectively to investigate the influence of each component. Besides, we also demonstrate the adaption of our approach to general language generation models by directly fine-tuning BART and HINT on downstream datasets with the proposed two objectives as auxiliary tasks. For fair comparison, we set all the pretrained models to the base version. And we also insert the sentence token and discourse token into each training text for all the baselines.

We generate texts using nucleus sampling (Holtzman et al., 2020) with p=0.9 and a softmax temperature of 0.7 (Goodfellow et al., 2016) to balance the trade-off between diversity and fluency. And we set the probability of generating $\langle\texttt{dis}\rangle$ to 1 if the last token is $\langle\texttt{sen}\rangle$ to ensure that HINT can obtain the high-level representations for each sentence. And during evaluation, we remove the special tokens in the generated texts. We apply these settings to all the baselines.

### 4.4 Automatic Evaluation

**Evaluation Metrics** We adopt the following automatic metrics to evaluate the performance on the test sets: **(1) Perplexity (PPL)**: Smaller perplexity scores indicate better fluency in general. We do not count the probability values at the positions where the sentence or discourse token is the golden truth. **(2) BLEU (B-n)**: We use $n = 1, 2$ to evaluate $n$-gram overlap between generated texts and human-written texts (Papineni et al., 2002). **(3) Lexical Repetition (LR-n)**: The metric computes the percentage of those texts which repeat a 4-gram at least $n$ times in all the generated texts (Shao et al., 2019). We set $n = 2$ for ROC and $n = 5$ for WP. **(4) Semantic Repetition (SR-n)**: The metric first computes the average top-$n$ SentenceBERT similarity between any two sentences in each generated text, and then averages the results as the final score. We set $n = 1$ for ROC and $n = 10$ for WP. **(5) Distinct-4 (D-4)** (Li et al., 2016): We adopt distinct-4, the ratio of distinct 4-grams to all the generated 4-grams, to measure the generation diversity. **(6) Context Relatedness:** It is a learnable automatic metric (Guan and Huang, 2020). First, we train a classifier with RoBERTa$_{\text{BASE}}$ (Liu et al., 2019) to distinguish human-written texts and negative samples constructed by substituting words, phrases and sentences of human-written texts randomly. Then, we use the average classifier score of all the generated texts to measure the context relatedness. **(7) Sentence Orders:** In analogy to relatedness measurement, we train another classifier to distinguish human-written texts and negative samples where sentences are randomly shuffled. We use the average classifier score to measure sentence orders. We train the last two metrics based on the training sets of the downstream datasets.

**Results on ROC** We show the results on ROC in Table 2. We do not provide the perplexity scores of Plan&Write and GPT-2 since they do not tokenize texts with the same vocabulary as used in BART. HINT outperforms all the baselines in terms of perplexity, indicating the better ability to model the texts in the test set. And HINT can generate more word overlaps with reference texts as shown by better BLEU scores. It is accordant with the previous observation (Xu et al., 2020) that Plan&Write has less lexical repetition than pretraining models possibly because small models are better at learning short term statistics (e.g., $n$-gram) but not long term dependencies. However, HINT improves the situation compared with GPT-2 and BART, and has less semantic repetition than all the baselines, indicating the better ability of HINT to capture semantic features. Besides, our approach does no harm to the generation diversity. HINT also outperforms baseline models in generating related events and arranging a proper order, as shown by the higher relatedness and order scores. Furthermore, fine-tuning with the proposed objectives as auxiliary tasks can further reduce the lexical and semantic repetition, and improve the relatedness and order scores for both BART and HINT, suggesting the general benefit of modeling the long-range coherence at sentence level and discourse level.

Besides, the ablation test shows that the sentence-level and discourse-level representations are relatively more important to enhance the ability to generate texts with related events and reasonable orders, respectively. And both of them contribute to reducing semantic redundancy. When post-training only with the language modeling objective, almost all the metrics drops substantially, indicating the importance to model high-level coherence.

Furthermore, we also notice that some models achieve even higher relatedness score than the

| Models | PPL↓ | B-1↑ | B-2↑ | LR-2↓ | SR-1↓ | D-4↑ | Relatedness↑ | Order↑ |
|---|---|---|---|---|---|---|---|---|
| **Seq2Seq** | 18.14 | 0.302 | 0.130 | 0.280 | 0.626 | 0.663 | 0.841 | 0.685 |
| **Plan&Write** | N/A | 0.297 | 0.130 | **0.201** | 0.628 | 0.677 | 0.915 | 0.801 |
| **GPT-2** | N/A | 0.305 | 0.131 | 0.331 | 0.636 | 0.684 | 0.919 | 0.813 |
| **BART** | 9.83 | 0.307 | 0.133 | 0.307 | 0.635 | 0.699 | 0.916 | 0.816 |
| **BART-MTL** | 9.68 | 0.312 | 0.137 | 0.271 | 0.629 | 0.683 | 0.945 | 0.820 |
| **BART-Post** | 9.49 | 0.326 | 0.147 | 0.279 | 0.632 | 0.698 | 0.947 | 0.842 |
| **HINT** | **9.20** | 0.334 | **0.154** | 0.253 | 0.619 | 0.693 | 0.987 | 0.882 |
| **w/o Sen** | 9.25 | 0.332 | 0.152 | 0.264 | 0.622 | 0.702 | 0.970 | 0.873 |
| **w/o Dis** | 9.24 | 0.329 | 0.150 | 0.248 | 0.621 | 0.694 | 0.978 | 0.864 |
| **w/o Sen&Dis** | 9.45 | 0.324 | 0.146 | 0.277 | 0.634 | 0.686 | 0.937 | 0.847 |
| **BART w/ aux** | 9.50 | 0.323 | 0.145 | 0.243 | **0.614** | **0.710** | 0.968 | 0.837 |
| **HINT w/ aux** | 9.22 | **0.335** | 0.153 | 0.232 | 0.615 | 0.700 | **0.989** | **0.892** |
| ***Golden Text*** | *N/A* | *N/A* | *N/A* | *0.058* | *0.531* | *0.891* | *0.970* | *0.903* |

Table 2: Automatic evaluation results on ROC. ↓ / ↑ means the lower/higher the better. The best performance is highlighted in **bold**. **w/o Sen** and **w/o Dis** means ablating the sentence-level and discourse-level representation learning, respectively. Namely, **w/o Sen&Dis** means post-training only with the language modeling objective. **BART w/ aux** and **HINT w/ aux** means fine-tuning BART and HINT on the downstream dataset with the proposed objectives as *aux*iliary tasks, respectively.

golden texts. We summarize the possible reasons as follows: (a) It is still difficult for the learned classifier to judge implicit relatedness in some golden texts, which may require a strong reasoning ability. (b) There exist some noisy texts with poor relatedness in the golden texts. And (c) the systems tend to generate a limited set of texts (as demonstrated by much lower distinct-4 than golden texts) with generic plots (Guan et al., 2020), which may get high relatedness scores easily. However, we believe the learnable metric is still meaningful to compare different models with similar diversity regarding the context relatedness.

**Results on WP** We present the results on WP in Table 3. We use a larger $n$ to compute the lexical/semantic repetition since we find that all the models tend to repeat similar texts easily when generating texts with hundreds of words. And we do not provide the relatedness and order scores because it is difficult to train satisfactory classifiers to distinguish human-written texts from negative samples well. Table 3 shows that HINT outperforms baselines except for lexical repetition, which is accordant with the results on ROC. Therefore, the high-level representations are effective for generating long texts with different lengths and domains.

## 4.5 Manual Evaluation

For manual evaluation, we conduct pair-wise comparisons with two strong baseline models (BART and BART-Post), and three ablated models of HINT. We randomly sample 200 texts from the test set of ROC[3] and obtain 1,200 texts from the six models.

| Models | PPL↓ | B-1↑ | B-2↑ | LR-5↓ | SR-10↓ | D-4↑ |
|---|---|---|---|---|---|---|
| **Seq2Seq** | 129.51 | 0.165 | 0.070 | 0.623 | 0.819 | 0.283 |
| **Plan&Write** | N/A | 0.199 | 0.070 | **0.524** | 0.851 | 0.272 |
| **GPT-2** | N/A | 0.200 | 0.073 | 0.655 | 0.883 | 0.287 |
| **BART** | 34.42 | 0.205 | 0.075 | 0.620 | 0.854 | 0.291 |
| **BART-MTL** | 35.71 | 0.198 | 0.076 | 0.654 | 0.846 | 0.305 |
| **BART-Post** | 35.11 | 0.205 | 0.076 | 0.671 | 0.862 | 0.271 |
| **HINT** | **32.73** | **0.224** | **0.084** | 0.567 | **0.805** | **0.313** |
| **w/o Sen** | 33.08 | 0.216 | 0.080 | 0.598 | 0.823 | 0.303 |
| **w/o Dis** | 33.18 | 0.223 | 0.083 | 0.588 | 0.818 | 0.307 |
| **w/o Sen&Dis** | 33.71 | 0.207 | 0.076 | 0.610 | 0.845 | 0.280 |
| ***Golden Text*** | *N/A* | *N/A* | *N/A* | *0.007* | *0.448* | *0.928* |

Table 3: Automatic evaluation results on WP.

| Models | Fluency |  |  |  |
|---|---|---|---|---|
|  | Win | Lose | Tie | $\kappa$ |
| **HINT vs. BART** | 37.5* | 24.0 | 38.5 | 0.58 |
| **HINT vs. BART-Post** | 35.5** | 21.0 | 43.5 | 0.63 |
| **HINT vs. HINT w/o Sen** | 37.0 | 31.0 | 32.0 | 0.68 |
| **HINT vs. HINT w/o Dis** | 33.0 | 25.5 | 41.5 | 0.62 |
| **HINT vs. HINT w/o Sen&Dis** | 35.5 | 28.5 | 36.0 | 0.60 |
| **Models** | **Coherence** |  |  |  |
|  | **Win** | **Lose** | **Tie** | $\kappa$ |
| **HINT vs. BART** | 54.5** | 11.0 | 34.5 | 0.59 |
| **HINT vs. BART-Post** | 47.5** | 21.5 | 31.0 | 0.62 |
| **HINT vs. HINT w/o Sen** | 47.5** | 23.0 | 29.5 | 0.67 |
| **HINT vs. HINT w/o Dis** | 42.0* | 28.0 | 30.0 | 0.63 |
| **HINT vs. HINT w/o Sen&Dis** | 55.5** | 24.0 | 20.5 | 0.58 |

Table 4: Manual evaluation results on ROC. The scores indicate the percentages (%) of **Win**, **Lose** or **Tie** when comparing HINT with a baseline. $\kappa$ denotes Fleiss' kappa (Fleiss and Joseph, 1971) to measure the inter-annotator agreement (all are *moderate* or *substantial*). The scores marked with * and ** mean HINT outperforms the baseline significantly with p-value<0.05 and p-value<0.01 (sign test), respectively.

[3] We do not conduct manual evaluation on WP since it would be hard to obtain acceptable annotation agreement for

| Aspects | Coherent Examples ↓ | | | | Incoherent Examples ↑ | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Rel | Neg | Caus | Temp | Rept | Rel | Neg | Caus | Temp |
| **Number** | 563 | 455 | 476 | 2,376 | 3,235 | 3,324 | 3,664 | 394 | 1,795 |
| **BART** | 11.91 | 9.15 | 10.56 | 10.29 | 14.11 | 15.60 | 13.69 | 13.47 | 13.04 |
| **BART-Post** | 11.46 | 8.86 | 10.21 | 9.94 | 14.06 | 15.45 | 13.35 | 13.31 | 12.72 |
| **HINT** | **10.90**$^{**}$ | **8.50**$^{*}$ | **9.68**$^{*}$ | **9.50**$^{**}$ | **14.74**$^{**}$ | **16.32**$^{**}$ | **13.96**$^{*}$ | **13.68** | **13.15** |
| **w/o Sen** | 11.00$^{*}$ | 8.55$^{*}$ | 9.75$^{*}$ | 9.53$^{**}$ | 14.04 | 15.43 | 13.29 | 13.59 | 13.04 |
| **w/o Dis** | 10.97$^{*}$ | 8.52$^{*}$ | 9.87 | 9.61$^{**}$ | 14.64$^{**}$ | 16.18$^{*}$ | 13.83$^{*}$ | 13.14 | 12.57 |
| **w/o Sen&Dis** | 11.41 | 8.84 | 10.16 | 9.89 | 13.80 | 15.14 | 13.17 | 13.04 | 12.51 |

Table 5: Perplexity scores on the coherent or incoherent examples within different aspects including *Semantic Repetition* (Rept), *Relatedness* (Rel), *Negation* (Neg), *Causal Relationship* (Caus) and *Temporal Relationship* (Temp). **Number** means the number of the corresponding test examples. ↓ / ↑ means the lower/higher perplexity the better. The best performance is highlighted in **bold**. $^{*}$ and $^{**}$ indicate that the corresponding model significantly outperforms BART with p-value<0.05 and p-value<0.01 (t-test), respectively.

For each pair of texts (one by our model and the other by a baseline, along with the input), three annotators are hired to give a preference (win, lose, or tie) in terms of fluency and coherence, respectively. We adopt majority voting to make final decisions among the three annotators. We resort to Amazon Mechanical Turk (AMT) for annotation. We follow Xu et al. (2020) to define *fluency* as a measure of intra-sentence linguistic quality and grammatical correctness, and *coherence* as inter-sentence relatedness, causal and temporal dependencies. Note that the two aspects are independently evaluated. Besides, we control the annotation quality by filtering out those annotations where the annotator can not make reasonable judgments when comparing a human-written text with a negative sample. Furthermore, we also ask workers to annotate the specific errors in the generated texts. We show the annotation instruction and the error analysis of different models in the appendix.

Table 4 shows the manual evaluation results. All the results show moderate inter-annotator agreement (0.4$\leqslant \kappa \leqslant$0.6) or substantial agreement (0.6 $\leqslant \kappa \leqslant$0.8). And we can see that HINT performs significantly better than baselines in coherence by capturing the high-level features, and has comparable fluency with baselines.

### 4.6 Language Modeling

It is still necessary to further investigate whether the learned representations help HINT capture the high-level coherence better. Therefore, we propose to evaluate the models using individual language modeling tests in different aspects (Ribeiro et al., 2020). To this end, we construct coherent and incoherent examples based on the test set of ROC, and compute perplexity on the examples of different aspects. Specifically, we focus on the following aspects: semantic repetition, relatedness, negation, causal and temporal relationship. We select human-written texts as coherent examples and construct incoherent examples by perturbing human-written texts. For example, we select those texts with time-related words (e.g., *"then"*) as coherent examples for testing in the temporal relationship. And we exchange two sequential events connected by *"then"* of a human-written text or substitute *"before"* with *"after"* as incoherent examples of the aspect. We show more details in the appendix.

We present the results in Table 5. HINT can model the context coherence better in the above aspects than baseline models (lower perplexity on the coherent examples), and recognize the incoherent errors more effectively (higher perplexity on the incoherent examples). By contrast, **both BART-Post and HINT (w/o Sen&Dis) achieve an overall drop of perplexity compared with BART even on the negative examples, indicating that they may still focus on capturing the token-level features**. As for the ablation study, we can see that the sentence-level representation enhances the ability of HINT to capture the relatedness, negation and semantic repetition, while the discourse-level representation works mainly for causal and temporal relationship. However, we also notice the insignificant improvement of HINT compared with BART in recognizing the unreasonable causal and temporal relationship, which may require injecting explicit inferential knowledge besides learning sentence orders.

too long texts.

### 4.7 Case Study

We present several cases in the appendix to demonstrate that HINT can derive meaningful sentence-level and discourse-level representations, and generate texts with better coherence than baselines with the help of the representations.

## 5 Conclusion

We present HINT, a generation model for ation, which can represent the prefix information at sentence level and discourse level in the decoding process. We propose two pretraining objectives including inter-sentence similarity prediction and sentence order discrimination to learn the sentence-level and discourse-level representations, respectively. Extensive experiments demonstrate that HINT can generate more coherent texts with related context and proper sentence orders than strong baselines. Further analysis shows that HINT has better ability of language modeling thanks to ability of modeling high-level coherence.

## Acknowledgments

This work was supported by National Key R&D Program of China, under Grant No. 2020AAA0104500. This work was jointly supported by the NSFC projects (Key project with No. 61936010 and regular project with No. 61876096), and the Guoqiang Institute of Tsinghua University, with Grant No. 2019GQG1 and 2020GQG0005. We would also like to thank the anonymous reviewers for their invaluable suggestions and feedback.

## Ethics Statement

We conduct the experiments based on two existing public datasets ROCStories and WritingPrompts, which are widely used for commonsense story generation and fiction generation tasks, respectively. Automatic and manual evaluation show that our model outperforms existing state-of-the-art models on both datasets, suggesting the generalization of our model to different domains. Besides, our approach can be easily extended to different syntactic levels (e.g., phrase-level, paragraph-level), different model architectures (e.g., GPT, UniLM) and different generation tasks (e.g., dialog generation, essay generation).

We resorted to Amazon Mechanical Turk (AMT) for manual evaluation. We did not ask about personal privacy or collect personal information of annotators in the annotation process. We hired three annotators and payed each annotator $0.05 for comparing each pair of stories. The payment is reasonable considering that it would cost average 30 seconds for an annotator to finish a comparison.

## References

Steven Bird and Edward Loper. 2004. NLTK: the natural language toolkit. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, July 21-26, 2004 - Poster and Demonstration*. ACL.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

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

Fleiss and L. Joseph. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. 2020. Content planning for neural story generation with aristotelian rescoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4319–4338.

Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph M. Weischedel, and Nanyun Peng. 2020. Content planning for neural story generation with aristotelian rescoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4319–4338. Association for Computational Linguistics.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

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

Jian Guan and Minlie Huang. 2020. UNION: an unreferenced metric for evaluating open-ended story generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9157–9166. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Dan Iter, Kelvin Guu, Larry Lansing, and Dan Jurafsky. 2020. Pretraining with contrastive sentence objectives improves discourse performance of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4859–4870. Association for Computational Linguistics.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. *Advances in neural information processing systems*, 28:3294–3302.

Haejun Lee, Drew A Hudson, Kangwook Lee, and Christopher D Manning. 2020. Slm: Learning a discourse language representation with sentence unshuffling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1551–1562.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119. The Association for Computational Linguistics.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1106–1115.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *EMNLP: System Demonstrations*, pages 119–126.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of NAACL-HLT*, pages 839–849.

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

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3973–3983.

Marco Túlio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4902–4912. Association for Computational Linguistics.

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2210–2219.

Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. 2019. Long and diverse text generation with planning-based hierarchical variational model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3257–3268, Hong Kong, China. Association for Computational Linguistics.

Bowen Tan, Zichao Yang, Maruan Al-Shedivat, Eric P. Xing, and Zhiting Hu. 2020. Progressive generation of long text. *CoRR*, abs/2006.15720.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33.

Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro. 2020. MEGATRON-CNTRL: controllable story generation with external knowledge using large-scale language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2831–2845. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. In *EMNLP-IJCNLP*, pages 563–578.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society.

# A Implementation Details

We implement our model based on $\text{BART}_{\text{BASE}}$ and use the public checkpoint and code of HuggingFace's Transformers[4]. Both the encoder and the decoder contain 6 hidden layers with 12 attention heads. The vocabulary consists of 50,625 tokens with Byte-Pair Encoding (Radford et al., 2019). And we regard ⟨mask⟩ and ⟨s⟩ in the original vocabulary as the sentence token ⟨sen⟩ and the discourse token ⟨dis⟩, respectively. The learning rate for both post-training and fine-tuning is 3e-5 with Adam as the optimizer. The Adam epsilon is 1e-6.

It cost about 32 hours for HINT's post-training on BookCorpus, and 7 hours/8 hours for fine-tuning on ROC/WP, respectively. The results are based on 1 NVIDIA TITAN X GPU.

# B Results on the Validation Set

Besides the performance on the test set which has been reported in the main paper, we also provide the performance on the validation set of ROC in Table 6 for HINT and strong baselines.

| Models | PPL | B-1 | LR-2 | SR-1 | Rel | Ord |
|---|---|---|---|---|---|---|
| **BART** | 10.04 | 0.315 | 0.301 | 0.634 | 0.924 | 0.821 |
| **BART-Post** | 9.75 | 0.321 | 0.278 | 0.630 | 0.949 | 0.850 |
| **HINT** | **9.45** | **0.331** | **0.249** | **0.623** | **0.989** | **0.881** |

Table 6: Automatic Evaluation results of different models on the validation set of ROC. We do not show BLEU-2 results due to the space limitation. **Rel** and **Ord** are short for *Relatedness* and *Order*, respectively.

# C Δ for Sentence-Level Representation Learning

We tune Δ in Equation 5 to investigate the influence of the margin between the predicted similarity score of HINT and that of SentenceBert. We present some automatic evaluation results with different Δ in Table 7. Note that we use $\Delta = 0.1$ for the experiments in the main paper. We can see that a smaller Δ (e.g., 0.01) would lead to less lexical and semantic repetition but worse fluency (indicated by higher perplexity) and context relatedness, which may be caused by the over-fitting to the model bias of the teacher model. On the other hand, a larger Δ (e.g., 0.5) would result in worse performance in almost all the metrics even than $\Delta = 1.0$ (without the similarity prediction task). The result indicates that a large Δ makes the model not learn effectively from the teacher model, and impact on the representations of the model itself. By contrast, $\Delta = 0.1$ would bring better overall performance.

| Δ | PPL↓ | B-1↑ | B-2↑ | LR-2↓ | SR-1↓ | Relatedness↑ |
|---|---|---|---|---|---|---|
| 0.01 | 10.00 | 0.313 | 0.139 | **0.249** | **0.599** | 0.937 |
| 0.05 | 9.78 | 0.316 | 0.140 | 0.264 | 0.610 | 0.962 |
| 0.1 | **9.20** | **0.334** | **0.154** | 0.253 | 0.619 | **0.987** |
| 0.2 | 9.67 | 0.326 | 0.146 | 0.273 | 0.628 | 0.975 |
| 0.5 | 9.72 | 0.319 | 0.143 | 0.261 | 0.629 | 0.954 |
| *1.0* | *9.25* | *0.332* | *0.152* | *0.264* | *0.622* | *0.970* |

Table 7: Automatic Evaluation results for HINT with different Δ. $\Delta = 1.0$ means post-training ablating the sentence-level representation learning (HINT w/o Sen).

# D Manual Evaluation

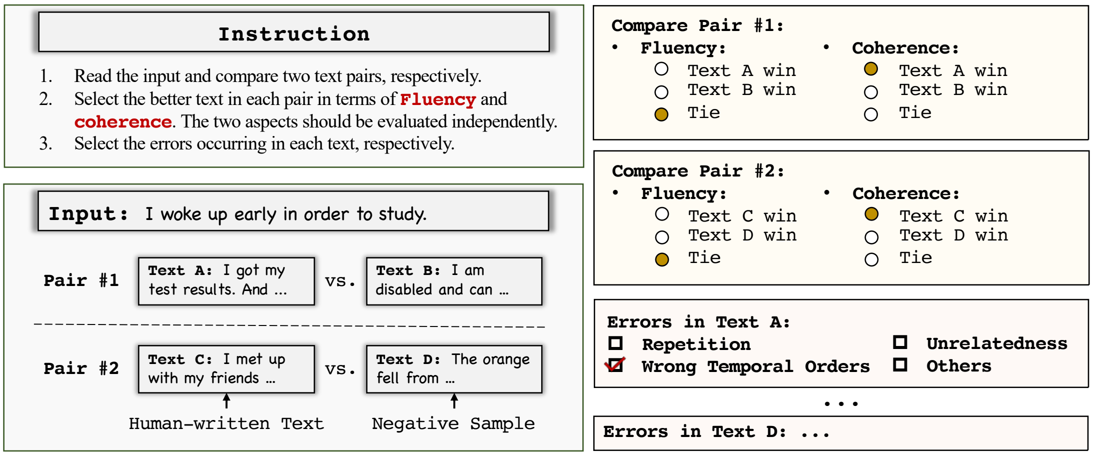
### Annotation Instruction

We show the manual annotation interface in Figure 3. In each HIT (human intelligence task) of AMT, we show workers an input along with two text pairs including (a) a pair of generated texts (one by HINT and the other by a baseline), and (b) a pair of the human-written text and a negative sample constructing by perturbing a text (e.g., repetition, substitution) randomly sampled from the data. Note that the two pairs are presented in random order. Then, we ask workers to select the better text in each pair in terms of the fluency and coherence, respectively. Besides, we also require workers to annotate the errors in each text, including *repetition* (repeating the same or similar words), *unrelatedness* (with unrelated entities or events to the input or within its own context), *wrong temporal orders*, and *others*. We reject an HIT where the worker does not think the human-written text has better coherence than the negative sample, or the worker does not annotate any errors for the negative sample. In this way, we reject 21.09% HITs in total. Finally, we ensure that there are three valid and independent comparison results for each pair of generated texts.

### Error Analysis

Based on the manual annotation of errors in the generated texts, we summarize the percentages of those texts with some error in all the annotated texts (200 for each model) in Table 8. We decide that a text contains some error when at least two of three annotators annotate the error for it. Note that each text of HINT is annotated five times (three annotators each time) since HINT is compared with other five models. Therefore, we take the average

[4] https://github.com/huggingface/transformers



**Instruction**

1. Read the input and compare two text pairs, respectively.
2. Select the better text in each pair in terms of **Fluency** and **coherence**. The two aspects should be evaluated independently.
3. Select the errors occurring in each text, respectively.

**Input:** I woke up early in order to study.

**Pair #1** **Text A:** I got my test results. And ... vs. **Text B:** I am disabled and can ...

**Pair #2** **Text C:** I met up with my friends ... vs. **Text D:** The orange fell from ...

Human-written Text Negative Sample

**Compare Pair #1:**
- **Fluency:** ○ Text A win ○ Text B win ● Tie
- **Coherence:** ● Text A win ○ Text B win ○ Tie

**Compare Pair #2:**
- **Fluency:** ○ Text C win ○ Text D win ● Tie
- **Coherence:** ● Text C win ○ Text D win ○ Tie

**Errors in Text A:**
□ **Repetition** ☑ **Wrong Temporal Orders** □ **Unrelatedness** □ **Others**

...

**Errors in Text D: ...**

Figure 3: A simplified version of the manual annotation interface.

of five annotation results. We can see that HINT has less repetition, better context relatedness and temporal orders than baselines. However, the results show that generating coherent long texts is still challenging.

| Models | Rept | Unrel | Temp | Others |
|---|---|---|---|---|
| **BART** | 32.5 | 48.0 | 43.5 | 6.5 |
| **BART-Post** | 30.5 | 38.5 | 46.0 | 19.5 |
| **HINT** | **12.0** | **13.5** | **18.8** | **9.8** |
| **w/o Sen** | 23.5 | 29.0 | 20.5 | 14.0 |
| **w/o Dis** | 16.0 | 15.5 | 42.0 | 18.0 |
| **w/o Sen&Dis** | 27.5 | 48.5 | 49.0 | 5.0 |

Table 8: Percentages (%) of the texts which are annotated with some error in all the annotated texts. The error types include repetition (Rept), unrelatedness (Unrel), wrong temporal orders (Temp) and others. The percentages in each row do not sum to 100% since each text may contain multiple errors. The best performance for each error type is highlighted in **bold**.

## E Constructing Coherent and Incoherent Examples

Table 9 presents the details for constructing examples to test the ability to model the context coherence in different aspects. However, the approach of automatic construction may inevitably introduce unexpected grammatical errors, which would also impact the text coherence. To alleviate the issue, we train a binary classifier on the CoLA corpus (Warstadt et al., 2019) to learn to judge the grammaticality, and then filter out those examples that are classified as ungrammatical (the classifier score less than 0.5). For simplicity, we directly use the public model from TextAttack (Morris et al., 2020) as the classifier, which achieves an accuracy of 82.90% on the test set of CoLA. Finally, we filter out about 15.51% of the test examples.

## F Case Study

### Sentence-Level Representation

Table 10 presents some cases from the test set of ROC to demonstrate the effectiveness of the learned sentence-level representation of HINT. We compute BLEU-1, BART similarity and HINT similarity for different sentence pairs, where BART/HINT similarity means the cosine distance between BART/HINT representations of two sentences. To obtain the BART representation of a sentence, we feed it into the BART decoder (along with its context) and apply mean-pooling on the hidden states at the last layer. HINT representation refers to the corresponding sentence-level representation after decoding the sentence. We normalize all the results into the standard Gaussian distribution[6]. We can see that HINT can derive meaningful sentence-level representations and gives high scores for semantically similar sentence pairs (the first two pairs) but low scores for dissimilar pairs (the last two pairs). By contrast, BART focuses more on token-level similarity and thus derives accordant similarity with BLEU.

### Discourse-Level Representation

We also present a case in Table 11 to indicate the effectiveness of the learned discourse-level representation of HINT. We consider a segment in the text of Table 11, which consists of two adjacent

[2] The paraphrases are generated based on the public checkpoint of the back translation augmentation system of UDA (Xie et al., 2020).

[6] We compute the mean and standard deviation within 2,000 sentence pairs randomly sampled from the test set.

| **Aspects** | **Selecting Coherent Examples** | **Creating Incoherent Examples** |
|---|---|---|
| **Semantic Repetition** | N/A | Repeating a sentence with its paraphrase by back translation[5].<br>`Case:` They got themselves and him on a diet. {They put themselves on a diet with him}$_{\texttt{insert}}$ ... |
| **Relatedness** | Texts with weak token-level semantic similarity in the context (e.g., with maximum inter-sentence MoverScore (Zhao et al., 2019) less than 0.1).<br>`Case:` Lilly was afraid of heights and fast movement. She was convinced to ride a roller coaster. She hated every minute of it. She ran off and threw up immediately after ... *(Maximum inter-sentence MoverScore =0.03)* | Substituting 20% nouns and verbs or a sentence randomly.<br>`Case:` The orange fell from the tree. It hit a girl on the head. {The girl looked up at the tree.}$_{\texttt{delete}}$ $\rightsquigarrow$ {She was unable to put the top up on her convertible.}$_{\texttt{insert}}$ Another orange fell from the tree. That orange broke her nose. |
| **Negation** | Texts with negated words (e.g., "not", "unable").<br>`Case:` The man turned it on. It *did not* respond. The man unplugged it. He took it apart. He could *never* get ... | Inserting or Deleting negated words for 20% sentences.<br>`Case:` The man turned it on. It {did not respond}$_{\texttt{delete}}$ $\rightsquigarrow$ {responded}$_{\texttt{insert}}$. The man unplugged it. He took it apart. He could never get that thing to work. |
| **Causal Relationship** | Texts with causality-related words (e.g., "so", "because").<br>`Case:` Mike had a very stressful job. He needed a vacation. *So* he took one. He headed to the sunny beaches of Mexico. Mike had a great time on his vacation. | Reversing the cause and effect (two individual sentences or clauses connected by a causality-related conjunction such as "so"); Substituting the causality-related words with the antonyms (e.g., "reason" vs "result").<br>`Case:` Mike had a very stressful job. {He took one.}$_{\texttt{reverse}}$ $\leftrightsquigarrow$ So {he needed a vacation.}$_{\texttt{reverse}}$ He headed to the sunny beaches of Mexico ... |
| **Temporal Relationship** | Texts with time-related words (e.g., "then").<br>`Case:` Karen got stung by a bee. Her arm swelled up immediately. It turned out she was allergic to bees! She had to go to the hospital for medication. *Then* she felt much better better! | Reversing two sequential events (two individual sentences or two clauses) connected by a time-related conjunction; Substituting the time-related words with the antonyms (e.g., after vs. before)<br>`Case:` ... Her arm swelled up immediately. It turned out she was allergic to bees! {She felt much better better!}$_{\texttt{reverse}}$ $\leftrightsquigarrow$ Then {she had to go to the hospital for medication.}$_{\texttt{reverse}}$ |

Table 9: Instruction for selecting coherent examples from human-written texts and creating incoherent examples by perturbing human-written texts. We highlight the keywords in *italic* which are crucial for the corresponding aspects. We construct the incoherent examples by inserting, deleting or reversion.

| **Sentence 1** | **Sentence 2** | **B-1** | **BART** | **HINT** |
|---|---|---|---|---|
| He was really embarrassed by it. | He was very embarrassed of it. | 5.08 | 2.83 | 2.17 |
| He dreamed of making the world a better place. | He had a passion to change his country for better. | 0.30 | 0.63 | 2.17 |
| He wasn't having a good time. | He was having a good time. | 7.17 | 2.04 | 1.65 |
| I wanted to buy some fruit. | I wanted to go to a state college. | 1.40 | 1.46 | -0.50 |

Table 10: Sentence pairs sampled from the test set of ROC and the corresponding BLEU-1 (B-1), BART similarity and HINT similarity.

sentences (e.g., the segment $\underline{\text{③④}}$in ②$\underline{\text{③④}}$⑤). Then, we can derive the segment representation by concatenating the contextualized representations of the two sentences. Besides, if we reverse the two sentences (from $\underline{\text{③④}}$ to $\underline{\text{④③}}$, other sentences in the text unchanged), we can also derive the segment representation in the same way. Note that in this case we concatenate the two sentence representations still in the normal order (i.e., first the representation of ③ and then that of ④). We expect the segment representations before and after the reversion to be distant in the vector space if the sentence representation contains discourse-level information. Otherwise, the segment representations would be similar since the segments have the same tokens before and after the reversion. For BART, we derive the sentence representation by feeding the whole text into BART and mean pooling the hidden states at the positions of tokens in the sentence. And for HUGO, we regard the corresponding discourse-level representation of each sentence as the sentence representation. For reference, we also show the results using the hidden state of BART at the position of the discourse token as the sentence representation, i.e., B (D). Table 11 shows the similarity between the segment representations before and after the sentence reversion. All the results are normalized into the standard Gaussian distribu-

**Input:**
①Kate was at her garbage can on a dark night.

**Human-written Text:**
② And a raccoon was standing near the can.
③ It started to come towards her.
④ Kate turned and ran to the house hoping it wasn't behind her.
⑤ Once inside she was relieved to see it hadn't followed her.

| **Before** | **After** | **B (M)** | **B (D)** | **HINT** |
|---|---|---|---|---|
| $\underline{\text{②③}}$④⑤ | $\underline{\text{③②}}$④⑤ | 4.05 | 5.32 | -0.89 |
| ②$\underline{\text{③④}}$⑤ | ②$\underline{\text{④③}}$⑤ | 1.30 | 3.81 | -1.08 |
| ②③$\underline{\text{④⑤}}$ | ②③$\underline{\text{⑤④}}$ | 1.96 | 4.17 | -3.82 |

Table 11: A human-written text sampled from the test set of ROC with five sentences from ① to ⑤. We consider two adjacent sentences as a segment (underlined) and compute the similarity of the segment representations (derived by BART or HINT) **Before** and **After** reversing the two sentences. B (M) and B (D) mean using BART to derive the sentence representation by mean-pooling and taking the hidden state at the position corresponding to the discourse token, respectively.

| | |
|---|---|
| **Input:** | I woke up early in order to study. |
| **Golden Text:** | I **met up with my friends** at the library. We **reserved a studying room** on the first floor. A few hours have passed since we **stayed for a while**. By the time **we got out**, it was already midnight. |
| **Seq2Seq:** | I am disabled and *can not walk* into the library. I had to *look through it for about three days*. I found out I *had my book* in two days. I *walked* to the library to look for it. |
| **Plan&Write:** | I *got my test results*. And then I *began to study*. I *fell asleep* for the test and was very *sleepy*. I awoke to find my test results had been negative. |
| **BART:** | I woke up late, and *found a few other people* already on the road. I was able to *get to work*, but I also had to keep my eyes open. I *didn't want to miss* any of the work. So I was *too lazy to get to work*. |
| **BART-CLS:** | I had a very difficult time *finding my position*. I had to leave my house early in order to *got to a hotel*. I had to *walk around* the block in order to get to the hotel. I ended up *walking the rest* of the way to the hotel. |
| **BART-Post:** | I drove to the library. I *went to the library* and found out that the library *closed* today. I *got* to the library and *decided to go* to the library. I *sat in the library* and *went to the library* and finished my studies. |
| **HINT:** | I had to **go to the library** and **finish my homework**. I quickly **made it to the library** and **began to study**. After studying for an hour, I **fell asleep**. I ended up **not finishing** my homework and **went back home**. |

Table 12: Generated texts by different models. *Italic* words indicate the improper entities or events in terms of coherence in the context. And **bold** words denote the coherent event sequence.

tion[7]. The results show that BART derives similar representations for the segments before and after reversion whether using mean-pooling or the hidden state corresponding to the discourse token. In comparison, although the reversion does not change the sentence semantics, segment representations derived by HINT are very dissimilar, suggesting that HINT can derive meaningful discourse-level representations.

### Text Generation

We presented some generated cases in Table 12. HINT can generate more coherent stories than baselines. Specifically, the baselines can easily predict some words which are related to the input (e.g., *"sleepy", "library"*) or within its own context (e.g, *"test results", "hotel"*). However, these words are used incoherently. For example, the text generated by Plan&Write has a wrong temporal order among the sentences (first *"got test results"* and then *"fell asleep for the test"*). The texts generated by Seq2Seq, BART and BART-CLS are chaotic in semantics and discourse structures. The text generated by BART-Post suffers from repetitive plots (*"went to the library"*) and conflicting logic (*"the library closed"* but *"sat in the library"*). By contrast, the text generated by HINT has a coherent event sequence with related content and a proper temporal order. The results indicate the effectiveness of modeling high-level coherence for ation.

[7] We compute the mean and standard deviation within 2,000 segment pairs sampled from the test set of ROC.