

# CLSEG: CONTRASTIVE LEARNING OF STORY ENDING GENERATION

Yuqiang Xie   Yue Hu\*   Luxi Xing   Yunpeng Li   Wei Peng   Ping Guo

Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China  
School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

## ABSTRACT

Story Ending Generation (SEG) is a challenging task in natural language generation. Recently, methods based on Pre-trained Language Models (PLM) have achieved great prosperity, which can produce fluent and coherent story endings. However, the pre-training objective of PLM-based methods is unable to model the consistency between story context and ending. The goal of this paper is to adopt contrastive learning to generate endings more consistent with story context, while there are two main challenges in contrastive learning of SEG. First is the negative sampling of wrong endings inconsistent with story contexts. The second challenge is the adaptation of contrastive learning for SEG. To address these two issues, we propose a novel Contrastive Learning framework for Story Ending Generation (CLSEG)<sup>†</sup>, which has two steps: multi-aspect sampling and story-specific contrastive learning. Particularly, for the first issue, we utilize novel multi-aspect sampling mechanisms to obtain wrong endings considering the consistency of order, causality, and sentiment. To solve the second issue, we well-design a story-specific contrastive training strategy that is adapted for SEG. Experiments show that CLSEG outperforms baselines and can produce story endings with stronger consistency and rationality.

**Index Terms**— Contrastive Learning, Story Generation, Pre-trained Language Model, Natural Language Generation

## 1. INTRODUCTION

Story Ending Generation (SEG) is a challenging task in natural language generation and artificial intelligence (AI), which aims to complete the plot and conclude a story given a story context [1, 2, 3]. SEG systems require to understand the story context, and then generate coherent, reasonable, and diversified endings according to the temporal and causal relationships. Previous state-of-the-art Seq2Seq based methods, like GPT-2 [4] and BART [5], mainly generate in a left-to-right manner and train with language model objective, which undergo an issue of generating inconsistent and safe endings.

To solve this issue, researches consider integrating semantic [6, 7], commonsense [8, 9, 10], sentiment/emotion [11, 12]

or even multi-modal [13] knowledge into backbone models. Particularly, [14] introduces a self-supervised task to distinguish true stories from auto-constructed fake stories to incorporate commonsense knowledge into GPT-2, so that the content of generated endings appears more coherent. In terms of consistency and rationality, however, there is still a big gap between machines and humans. One reason is that the proposed self-supervised classification task can learn the difference between true story and fake story, but it can not directly model the consistency between story context and endings.

We argue that the comparison of story context and various endings is vital to generating more consistent story endings, therefore it is necessary to seek an approach to better grasp the consistency. Based on the above analysis, we adopt the paradigm of contrastive learning [15] to introduce an approach for story ending generation which achieves the goal of modeling the consistency between story context and candidate endings. **There are two crucial problems in contrastive learning of story ending generation:**

**Q1:** *How to generate high-quality wrong endings inconsistent with story contexts?*

**Q2:** *How to design a contrastive training strategy adapted for story ending generation?*

To address the aforementioned problems, we propose a novel Contrastive Learning framework for Story Ending Generation (CLSEG), which has two steps: multi-aspect sampling and story-specific contrastive learning. Targeted at **Problem Q1**, multi-aspect sampling mechanisms are utilized to sample high-quality wrong endings by considering the consistency of order, causality, and sentiment. To solve **Problem Q2**, a story-specific contrastive training strategy is designed, which is adapted for SEG. Experimental results demonstrate that CLSEG outperforms baselines and can produce more consistent and reasonable story endings.

Our contributions are summarized as below: (1) We propose a novel CLSEG framework for story ending generation that can generate endings more consistent to the story context. (2) To achieve contrastive learning of SEG, we introduce novel multi-aspect sampling mechanisms and a story-specific contrastive training strategy. (3) Automatic and manual evaluations demonstrate that CLSEG outperforms baselines on recall-oriented metrics and can generate story endings with stronger consistency and rationality.

\*Corresponding author. E-mail: huyue@iie.ac.cn

<sup>†</sup>Code and Data: <https://github.com/IndexFziQ/CLSEG>

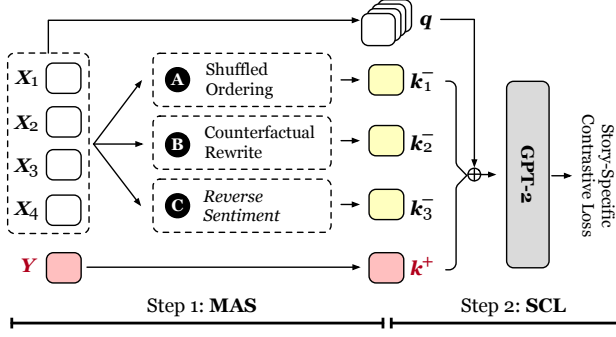


Fig. 1: Overview of CLSEG.

## 2. CLSEG

### 2.1. Overview

SEG task can be formulated as follows: given a story context consisting of a sequence of sentences  $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_m\}$  where  $m = 4$  in this paper.  $\mathcal{X}_i = \{x_i^1, x_i^2, \dots, x_i^n\}$  represents the  $i$ -th sentence that consists of  $n$  words, and  $1 \leq i \leq m$ . The goal of SEG is to generate a story ending  $\mathcal{Y}$  according to the given story context  $\mathcal{X}$ .

CLSEG is based on a LM architecture GPT-2 [4]. Firstly, aiming at generating candidate wrong endings, we design a series of methods considering the consistency of story order, causality, and sentiment in the multi-aspect sampling. Specifically, we introduce three novel negative sampling mechanisms: shuffled ordering, counterfactual rewriting and reverse sentiment. Then, we design a story-specific contrastive training strategy to further pre-train GPT-2 to learn the consistency between story context and endings, which compares right ending and wrong endings, thus helping GPT-2 to generate a more consistent ending.

### 2.2. Multi-Aspect Sampling (MAS)

In order to address **Problem Q1**, we generate high-quality wrong endings considering three aspects: order, causality, and sentiment. As exhibited in Fig. 1, Multi-Aspect Sampling (MAS) consists of three sampling mechanisms to generate wrong endings  $\{\mathcal{Y}_{SO}^-, \mathcal{Y}_{CR}^-, \mathcal{Y}_{RS}^-\}$  based on the original story context  $\mathcal{X}$ . Specifically, Shuffled Ordering (SO) is used for generating temporal inconsistent endings. Counterfactual Rewriting (CR) is designed for creating causal inconsistent endings. Reverse Sentiment (RS) is utilized for writing sentiment inconsistent endings.

**Shuffled Ordering (SO)** For the purpose of modeling the temporal consistency between story context and ending, we design a SO mechanism to generate wrong endings, which is similar to [14, 16]. In the beginning, we fine-tune a story ending generation model  $\text{SEG}_{\text{BASE}}$  based on GPT-2 [4] on ROCStories [1]. As shown in Fig. 2 (A), we randomly shuffle the order of story context. The wrong ending  $\mathcal{Y}_{SO}^-$  generated

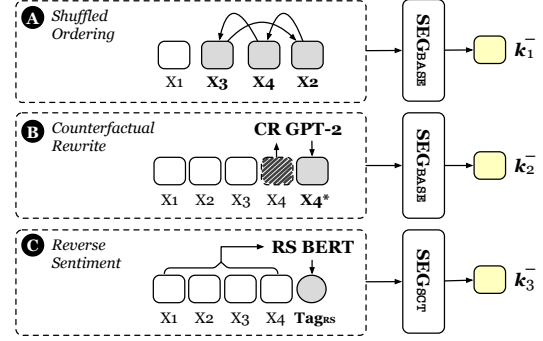


Fig. 2: Details of Multi-Aspect Sampling.

by SO mechanism is defined as:

$$\mathcal{Y}_{SO}^- = \text{SEG}_{\text{BASE}}(\text{shuffled}(\{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_m\})) \quad (1)$$

The generated wrong ending will be considered as the wrong ending for the next contrastive learning step.

**Counterfactual Rewriting (CR)** Aiming at taking the causal consistency between story context and ending into account, we introduce a Counterfactual Rewriting (CR) mechanism to create wrong endings. As demonstrated in Fig. 2 (B), we replace the last event  $\mathcal{X}_m$  of story context with a counterfactual rewriting event  $\mathcal{X}_m^*$ . Thus, the wrong ending  $\mathcal{Y}_{CR}^-$  generated by CR mechanism as below:

$$\mathcal{Y}_{CR}^- = \text{SEG}_{\text{BASE}}(\{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_m^*\}) \quad (2)$$

where  $\mathcal{X}_m^*$  is generated by **CR GPT-2** that trained on TIME-TRAVEL [17]. TIME-TRAVEL is a large-scale human-annotated counterfactual story generation data set which consists of almost 100K Counterfactual Rewriting sentence pairs (original event, counterfactual event).  $\mathcal{Y}_{CR}^-$  will be given to the next contrastive learning step.

**Reverse Sentiment (RS)** In order to consider the sentiment consistency between story context and ending, we propose a Reverse Sentiment (RS) mechanism to writing wrong endings. As displayed in Fig. 2 (C), the wrong ending  $\mathcal{Y}_{RS}^-$  generated by RS mechanism is defined as:

$$\mathcal{Y}_{RS}^- = \text{SEG}_{\text{SCT}}(\{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_m, \text{Tag}_{\text{RS}}\}) \quad (3)$$

where a reverse sentiment tag  $\text{Tag}_{\text{RS}} \in \{0, 1\}$  is predicted by an **RS BERT** which is a BERT<sub>LARGE</sub> [18] fine-tuned on the SST-2 [19] dataset. We utilize the **RS BERT** to label Story Cloze Test (SCT) [1] and extract the examples that the sentiment orientation of wrong ending is opposite to the sentiment orientation of story context. In statistics, 86.5% of examples fit our setting. A story ending generation model  $\text{SEG}_{\text{SCT}}$  is a GPT-2 [4] fine-tuned on labeled SCT [1].  $\mathcal{Y}_{RS}^-$  will be fed to the next contrastive learning step.

### 2.3. Story-Specific Contrastive Learning (SCL)

To solve **Problem Q2**, Story-Specific Contrastive Learning (SCL) is designed, which is adapted for SEG. As demon-

Models	BLEU $\uparrow$	R-1-P $\uparrow$	R-1-R $\uparrow$	R-1-F1 $\uparrow$	R-2-P $\uparrow$	R-2-R $\uparrow$	R-2-F1 $\uparrow$	R-L-P $\uparrow$	R-L-R $\uparrow$	R-L-F1 $\uparrow$	Meteor $\uparrow$
<b>GPT-2(PT)</b>	1.14	13.99	<b>15.99</b>	13.56	1.34	<b>2.51</b>	1.68	13.17	<b>15.89</b>	12.65	10.36
<b>GPT-2(FT)</b>	<b>2.57</b>	15.15	12.55	13.30	<b>2.12</b>	1.82	1.87	13.87	11.45	12.14	10.48
<b>GPT-2(GCL)</b>	1.71	14.81	14.66	14.21	1.82	2.13	1.89	13.67	12.56	13.12	10.54
<b>CLSEG</b>	1.97	<b>15.68</b>	14.73	<b>14.63</b>	1.87	2.15	<b>1.91</b>	<b>14.44</b>	13.61	<b>13.48</b>	<b>10.73</b>

**Table 1:** The results of automatic evaluation considering common-used metrics.  $\uparrow$  indicates the higher, the better.

strated in Fig. 1, we design a story-specific contrastive training strategy for **CLSEG**. Different from contrastive learning for natural language understanding tasks (pre-training encoder), we focus on pre-training decoder for SEG.

In Section 2.2, we explicitly construct negative examples for each right endings in ROCStories [1] for **SCL**. We encourage story context  $\mathcal{X}$  (the query, denoted as  $q$ ) to be as consistent as possible to the right ending  $\mathcal{Y}$  (the positive sample, denoted as  $k^+$ ) but inconsistent to all wrong endings  $\{\mathcal{Y}_{SO}^-, \mathcal{Y}_{CR}^-, \mathcal{Y}_{RS}^-\}$  based on the original story context  $\mathcal{X}$  (i.e., the negative sample with inconsistent order, causal, and sentiment denoted as  $\{k_i^-\}_{i=1}^3 = \{k_1^-, k_2^-, k_3^-\}$  individually) in a training batch. Inspired by [20], the story-specific contrastive loss for  $\mathcal{X}$  is defined as:

$$\mathcal{L}_{SCL}^t(p_\theta(\mathcal{Y}|\mathcal{X}), \{\mathcal{Y}_i^-\}_{i=1}^3) = -\log p_\theta(k_t^+ | k_{<t}^+, q) - \alpha \cdot \frac{1}{N} \sum_i \log(1 - p_\theta(k_{i,t}^- | k_{i,<t}^-, q)) \quad (4)$$

where  $p_\theta$  represents backbone LM GPT-2,  $t$  is the time step, and  $\alpha$  is a hyper-parameter. In this paper, we set  $N = 3$ .

### 3. EXPERIMENTS

#### 3.1. Experimental Setup

**Dataset** We evaluate our model on the ROCStories [1] corpus. The corpus contains 98,162 five-sentence stories for evaluating story understanding and generation. Following the 8:1:1 splitting ratio, we obtain 78,530/9,816/9,816 five-sentence stories as train/dev/test sets for the SEG task.

**Implement Details** The stories are tokenized using byte pair encoding (BPE). We set the parameters following the medium version of [4]’s design. The batch size is 32 during fine-tuning on the ROCStories corpus using Adam optimizer with an initial learning rate of  $5e-5$ . We set  $\alpha = 1$ . We generate stories using a greedy sampling strategy. We use the HuggingFace<sup>‡</sup> [21] PyTorch [22] implementation on Tesla V100 GPU.

#### 3.2. Evaluation Metrics

**Automatic** We use the following metrics for automatic evaluation<sup>¶</sup> [23]: **(1) BLEU** [24] is used for evaluating the overall quality of the generated story. We use geometric mean

<sup>‡</sup><https://github.com/huggingface/transformers>

<sup>¶</sup>Evaluate with <https://github.com/thu-coai/OpenMEVA>

Model	Quality $\uparrow$		Order	Consistent $\uparrow$	
	Fluency	Coherence		Causal	Sentiment
<b>GPT-2(PT)</b>	1.87	1.58	1.22	1.07	1.19
<b>GPT-2(FT)</b>	<b>2.24</b>	2.13	1.47	1.67	1.68
<b>GPT-2(GCL)</b>	2.08	2.07	1.62	1.73	1.63
<b>CLSEG</b>	2.20	<b>2.43</b>	<b>2.08</b>	<b>2.19</b>	<b>2.02</b>

**Table 2:** Manual Evaluation in terms of quality and rationality about the generated story endings.

of 1-gram to 4-gram. **(2) ROUGE (R-n-P/R/F1)** [25] (with  $n=1, 2, L$ ) is used to measure the recall-oriented similarity between automatically generated and reference results, including Precision, Recall and F1. **(3) Meteor** [26] is based on the harmonic mean of unigram precision and recall, with recall weighted higher than precision. This metric can produce good correlation with human judgement.

**Manual** We also conduct a manual evaluation of generated story endings. Crowd-workers are required to evaluate actions on a 0-3 scale (3 being very good) from two different perspectives: **(1) Content Quality** to indicate whether the generated story ending is fluent and coherent. **(2) Content Rationality** to assess whether the story endings are reasonable and consistent to the story context.

#### 3.3. Results

**Baselines** We use the following baselines: **GPT-2(PT)**: This model directly use the public checkpoint of pre-trained parameters for story ending generation which follows the setting of paper [4]. **GPT-2(FT)**: This model is fine-tuned on the ROCStories corpus from the public checkpoint of pre-trained parameters. **GPT-2(GCL)**: This model is pre-trained similar to our model, but the negative sampling is in general noisy ways: *random shuffle/drop/replace tokens*.

**Automatic Evaluation** The results of automatic evaluation are shown in Table 1. GPT-2(PT) shows highest recall but the lowest precision and BLEU scores. One reason is that GPT-2(PT) tends to generate as many high-frequency co-occurring tokens as possible. However, GPT-2(PT)’s generated endings is of low quality. **CLSEG** outperforms GPT-2(FT) and GPT-2(GCL) in terms of recall-oriented Meteor, and the recall of Rouge scores than all the baselines, indicating better consistency with the reference golden endings. Meanwhile, compared with GPT-2(PT), the content quality (BLEU and Precision of Rouge) of **CLSEG**’s generated endings is guar-

Model	BLEU $\uparrow$	R-1-R $\uparrow$	R-2-R $\uparrow$	R-L-R $\uparrow$	Meteor $\uparrow$
CLSEG	1.97	<b>14.73</b>	<b>2.15</b>	<b>13.61</b>	<b>10.73</b>
only SO	1.94	13.77	1.96	12.78	10.16
only CR	2.13	13.71	1.96	12.66	10.20
only RS	<b>2.33</b>	14.24	2.03	13.02	10.64

**Table 3:** Ablation study of CLSEG.

<b>Context</b>	Yesterday, I needed to buy a belt so I went to the mall. After choosing which store to shop from, I quickly got out of my car. Upon going inside there was an associate to help me. We talked about what i wanted and she led me to the section.
<b>Golden</b>	I picked out a great belt and left feeling excellent!
<b>GPT-2(PT)</b>	I was so out of my car I was about to buy a belt.
<b>GPT-2(FT)</b>	I was so quickly and quickly, i was so out of my car.
<b>GPT-2(GCL)</b>	I was so so so ...
<b>CLSEG</b>	After I got my belt, I went to the store.

**Table 4:** Generated story endings by different models.

anteed. Besides, we have conducted p-value [27] evaluation for significance of the differences. The results show that our method outperforms all baseline models significantly with p-value  $< 0.012$  (Wilcoxon signed-rank test).

**Manual Evaluation** We perform a manual evaluation between our model and baselines. We randomly generate 200 stories from the test set. For each story, we hire three annotators to give a score in terms of content quality (fluency, coherence) and content rationality (order, causal, sentiment). For each aspect, we use average of the three annotations. We adopt majority voting to make the final decisions among the annotators. As exhibited in Table 2, all the results show that our model outperforms baselines significantly in content coherence, and all aspects of consistency.

#### 4. DISCUSSION AND ANALYSIS

**Ablation Study** An ablation study is conducted on the ROC-Stories dataset to examine the impact of each negative sampling mechanism separately. We train the model each time by using one of our negative sampling mechanisms. As shown in Table 3, the results (ROUGE recall accuracy and Meteor) illustrate the harms that the elimination of each of the proposed negative sampling mechanisms (SO, CR or RS) from CLSEG could cause. In summary, all negative sampling mechanisms contribute for CLSEG and RS shows best performance.

**Case Study** In this part, we present some generated examples in Table 4. The first two lines are the story context and the golden right ending. From the next 4 lines, our model can generate more consistent and reasonable story ending to the story context than baselines. In specific, **GPT-2(PT)** generate a repetitive event to story context. Besides, the generated ending of **GPT-2(FT)** are inconsistent and far from the golden

Error Type	Story Context + Ending
<b>Repetition</b>	Morgan enjoyed long walks on <i>the beach</i> . She and her boyfriend decided to go ... Morgan decided to <i>propose</i> to her boyfriend. <i>The walk to the beach and the propose.</i>
<b>Conflicting</b>	Frank had been drinking beer. He got a call from his girlfriend ... <i>Since Frank was already a bit drunk, he could not drive. Frank had to drive to his date.</i>
<b>Ambiguous</b>	Sunny enjoyed going to the beach. As she stepped out of her car, ... Sunny got back into her car and heading towards the mall. <i>She was going to the park.</i>

**Table 5:** Typical errors generated by CLSEG. *Italic* words denote the error generated story endings.

endings. And, **GPT-2(GCL)** fails to generate a event. The examples show that CLSEG can generate more consistent endings by contrastive learning high-quality wrong endings.

**Error Analysis** In order to explore the limitation of CLSEG, we also perform error analysis by studying many inconsistent endings lost to baselines in manual evaluation. Hence, we manually examine all the bad endings in the pairwise comparison between our model and two strong baselines **GPT-2(FT/GCL)** to analyze the types of errors. Error types indicate the causes that influence the performance. The numbers of endings lost to our model are 43/58 of 100/100 in total for **GPT-2(FT/GCL)**, respectively. And there are 35 endings of 200 generated by CLSEG losing to these two baselines.

We conclude three main types of error from the lost endings: **Repetition** (repeating the same words or event to the story context), **Conflicting** (inconsistent to the story context), and **Ambiguous** (difficult to understand the ending). As displayed in Table 5, CLSEG generates the repeated words corresponding to story context. The last two examples show conflicting and ambiguous endings. All in all, the analysis result illustrates that generating a high-quality consistent ending to the story context is still a challenging task.

#### 5. CONCLUSION AND FUTURE WORK

In this work, we propose a novel CLSEG framework for story ending generation which has two steps: multi-aspect sampling (MAS) and story-specific contrastive learning (SCL) to generate endings more consistent to story context. Experimental results demonstrate that CLSEG outperforms baselines and creates more consistent and reasonable story endings. In the future, we will consider constructing harder yet high-quality wrong endings, designing more appropriate CL training strategies and applying CLSEG to more NLP tasks.

#### 6. ACKNOWLEDGMENTS

We thank all anonymous reviewers for their constructive comments. This work is supported by the National Natural Science Foundation of China (No.62006222 and No.U21B2009).

## 7. REFERENCES

- [1] N. Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen, “A corpus and cloze evaluation for deeper understanding of commonsense stories,” in *NAACL*, 2016.
- [2] Rishi Sharma, James Allen, Omid Bakhshandeh, and N. Mostafazadeh, “Tackling the story ending biases in the story cloze test,” in *ACL*, 2018.
- [3] Yan Zhao, Lu Liu, Chunhua Liu, Ruoyao Yang, and Dong Yu, “From plots to endings: A reinforced pointer generator for story ending generation,” *ArXiv*, vol. abs/1901.03459, 2018.
- [4] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever, “Language models are unsupervised multitask learners,” in *OpenAI Blog*, 2019.
- [5] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer, “BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” in *ACL*, 2020.
- [6] Qingbao Huang, Linzhang Mo, Pijian Li, Yi Cai, Qingguang Liu, Jielong Wei, Qing Li, and Ho fung Leung, “Story ending generation with multi-level graph convolutional networks over dependency trees,” in *AAAI*, 2021.
- [7] Peng Xu, Mostofa Ali Patwary, Mohammad Shoebybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro, “Controllable story generation with external knowledge using large-scale language models,” in *EMNLP*, 2020.
- [8] Zhongyang Li, Xiao Ding, and Ting Liu, “Generating reasonable and diversified story ending using sequence to sequence model with adversarial training,” in *COLING*, 2018.
- [9] Jian Guan, Yansen Wang, and Minlie Huang, “Story ending generation with incremental encoding and commonsense knowledge,” in *AAAI*, 2019.
- [10] Debjit Paul and A. Frank, “Coins: Dynamically generating contextualized inference rules for narrative story completion,” in *ACL/IJCNLP*, 2021.
- [11] Fuli Luo, Damai Dai, Pengcheng Yang, Tianyu Liu, Baobao Chang, Zhifang Sui, and Xu Sun, “Learning to control the fine-grained sentiment for story ending generation,” in *ACL*, 2019.
- [12] Faeze Brahman and Snigdha Chaturvedi, “Modeling protagonist emotions for emotion-aware storytelling,” in *EMNLP*, 2020.
- [13] Qingbao Huang, Chuan Huang, Linzhang Mo, Jielong Wei, Yi Cai, Ho fung Leung, and Qing Li, “Igseg: Image-guided story ending generation,” in *FINDINGS*, 2021.
- [14] Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang, “A knowledge-enhanced pretraining model for commonsense story generation,” *TACL*, 2020.
- [15] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi, “A theoretical analysis of contrastive unsupervised representation learning,” in *ICML*, 2019.
- [16] Yuqiang Xie, Yue Hu, Luxi Xing, Chunhui Wang, Yong Hu, Xiangpeng Wei, and Yajing Sun, “Enhancing pre-trained language models by self-supervised learning for story cloze test,” in *Knowledge Science, Engineering and Management*, 2020.
- [17] Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi, “Counterfactual story reasoning and generation,” in *EMNLP*, 2019.
- [18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *NAACL-HLT*, 2019.
- [19] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, A. Ng, and Christopher Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *EMNLP*, 2013.
- [20] Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston, “Neural text generation with unlikelihood training,” in *ICLR*, 2020.
- [21] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush, “Transformers: State-of-the-art natural language processing,” in *EMNLP- Demos*, 2020.
- [22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *NeurIPS*, 2019.
- [23] Jian Guan, Zhexin Zhang, Zhuoer Feng, Zitao Liu, Wenbiao Ding, Xiao-Xi Mao, Changjie Fan, and Minlie Huang, “Openmeva: A benchmark for evaluating open-ended story generation metrics,” in *ACL/IJCNLP*, 2021.
- [24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *ACL*, 2002.
- [25] Chin-Yew Lin, “Rouge: A package for automatic evaluation of summaries,” in *ACL*, 2004.
- [26] Satanjeev Banerjee and Alon Lavie, “Meteor: An automatic metric for mt evaluation with improved correlation with human judgments,” in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005.
- [27] Anders Søgaard, Anders Johannsen, Barbara Plank, Dirk Hovy, and Hector Martínez Alonso, “What’s in a p-value in NLP?,” in *CoNLL*, 2014.