Stylized Story Generation with Style-Guided Planning

Xiangzhe Kong*, Jialiang Huang*, Ziquan Tung, Jian Guan and Minlie Huang†
The CoAI group, DCST, Institute for Artificial Intelligence,
State Key Lab of Intelligent Technology and Systems,
Beijing National Research Center for Information Science and Technology,
Tsinghua University, Beijing 100084, China
{kxz18,huang-jl17,tongzq18,j-guan19}@mails.tsinghua.edu.cn,
aihuang@tsinghua.edu.cn

Abstract

Current storytelling systems focus more on generating stories with coherent plots regardless of the narration style, which is important for controllable text generation. Therefore, we propose a new task, stylized story generation, namely generating stories with specified style given a leading context. To tackle the problem, we propose a novel generation model that first plans the stylized keywords and then generates the whole story with the guidance of the keywords. Besides, we propose two automatic metrics to evaluate the consistency between the generated story and the specified style. Experiments demonstrates that our model can controllably generate *emotion*-driven or *event*-driven stories based on the ROCStories dataset (Mostafazadeh et al., 2016). Our study presents insights for stylized story generation in further research.

1 Introduction

Story generation is a challenging task in natural language generation (NLG), namely generating a reasonable story given a leading context. Recent work focuses on enhancing the coherence of generated stories (Fan et al., 2018; Yao et al., 2019) or introducing commonsense knowledge (Guan et al., 2020; Xu et al., 2020). However, it has not yet been investigated to generate stories with controllable styles, which is important since different styles serve different writing purposes. As exemplified in Figure 1, *emotion-driven* stories use emotional words (e.g., *"excited"*, *"enjoyed"*) to reveal the inner states of the characters and bring the readers closer to the characters. In comparison, *event-driven* stories usually contain a sequence of events with a clear temporal order (e.g., *"tearing"*→*"tried"*→*"found"*→*"hooked"*), which aims to narrate the story objectively.

Leading Context: Alice bought a new television.

Style	Story
Emotion-driven	She got **excited** to have it. It was **pretty** for the extra large screen. After using, she was very **satisfied** with the sound and design. She didn't **regret** buying this TV.
Event-driven	The picture on screen was **tearing**. Then she **tried** different adjustments. Eventually, she **found** out what was wrong. She got the wrong cable **hooked** up.

Figure 1: Example of stylized story generation given the same leading context. The stylized keywords are in bold.

In this paper, we formalize the task of stylized story generation, which requires generating a coherent story with a specified style given the first sentence as the leading context. *Style* has multiple interpretations, which can be seen as a unique voice of the author expressed through the use of certain stylistic devices (e.g. choices of words)(Mou and Vechtomova, 2020). In this work we focus on the choices of words and define the story styles based on the pattern of wording. Specifically, we focus on two story styles, including *emotion-driven* and *event-driven* stories. *Emotion-driven* stories contain abundant words with emotional inclination. We identify the emotional words using the off-the-shelf toolkit NRCLex (Mohammad, 2020), which supports retrieving the emotional effects of a word from a predefined lexicon. And *event-driven* stories tend to use serial actions as an event sequence. We use NLTK (Bird et al., 2009) to extract verbs in a story as the actions. Since no public datasets are available for learning to generate stylized stories, we regard the extracted words as stylistic keywords and then annotate the story styles for existing story datasets automatically based on the keyword distribution. Note that the story styles can be extended easily by defining new stylistic keywords.

In this work, we propose a generation model

*Equal contribution.
†Corresponding author.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2430–2436
August 1–6, 2021.

for stylized story generation. Our model first predicts the distribution of stylistic keywords and then generates a story with the guidance of the distribution. Furthermore, we propose two new automatic metrics to evaluate the consistency between the generated stories and the specified styles: lexical style consistency (LSC) and semantic style consistency (SSC), which focus on the number of stylistic keywords and the overall semantics, respectively. Extensive experiments demonstrate that the stories generated by our model not only achieve better fluency and coherence than strong baselines but also have better consistency with the specified styles.[1]

2 Related Work

Story Generation Recently there have been significant advances for story generation with the encoder-decoder paradigm (Sutskever et al., 2014), the transformer-based architecture (Vaswani et al., 2017) and the large-scale pre-trained models (Radford et al., 2019; Lewis et al., 2020). Prior studies usually decomposed the generation into separate steps by first planning a sketch and then generating the whole story from the sketch. The sketch is usually a series of keywords (Yao et al., 2019), a learnable skeleton (Xu et al., 2018) or an action sequence (Fan et al., 2019; Goldfarb-Tarrant et al., 2020). Another line is to incorporate external knowledge into story generation (Guan et al., 2020; Xu et al., 2020). However, generating stories with controllable styles has hardly been investigated.

Stylized Generation Stylized generation aims to generate texts with controllable attributes. For example, recent studies in dialogue systems focused on controlling persona (Zhang et al., 2018; Boyd et al., 2020), sentence functions (Ke et al., 2018), politeness (Niu and Bansal, 2018), and topics (Tang et al., 2019). In story generation, Huang et al. (2019) and Xu et al. (2020) controlled the story topics and planned keywords, respectively. Besides, for general text generation, the authorship (Tikhonov and Yamshchikov, 2018), sentiment (Hu et al., 2017), and topics (Li et al., 2020) can also be controlled for different purposes. We introduce a new controllable attribute in story generation, i.e., the story style, which has been paid little attention to in prior studies.

[1]Link to the code: `https://github.com/thu-coai/Stylized-Story-Generation-with-Style-Guided-Planning.git`

3 Proposed Method

In this section, we first show the task formulation for stylized story generation (§3.1). Then we present the details of our two-step model: **style-guided keywords planning** (§3.2) and **generation with planned keywords** (§3.3).

3.1 Task Formulation

Input: The first sentence $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ of a story with length n, where x_i is the i-th word. A special token l to indicate the expected style of the generated story. $l \in \{\langle\texttt{emo}\rangle, \langle\texttt{eve}\rangle\}$, which refers to the *emotion-driven* and *event-driven* styles, respectively. Besides, in the training phase, we set $l = \langle\texttt{other}\rangle$ if the training example is neither emotion-driven nor event-driven to improve the data efficiency.

Output: A story $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ of length m with the style l, where y_i is i-th word.

3.2 Planning

We insert l at the beginning of $\boldsymbol{x}$ and encode them as follows:

$$[\boldsymbol{h}_0, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_n] = \texttt{Enc}(l, x_1, x_2, \ldots, x_n), \quad (1)$$

where $\boldsymbol{h}_i$ $(1 \leqslant i \leqslant n)$ is the hidden state corresponding to x_i, $\boldsymbol{h}_0$ is the hidden state at the position of l, and $\texttt{Enc}$ is a bidirectional or unidirectional encoder. Then, we regard the stylistic keywords as bag-of-words (Kang and Hovy, 2020) and predict the keyword distribution $P_k(w|\boldsymbol{x}, l)$ over the whole vocabulary $\mathbb{V}$ as follows:

$$P_k(w|\boldsymbol{x}, l) = \text{softmax}(\mathbf{W}_k \boldsymbol{h}_c + \mathbf{b}_k), \quad (2)$$

where $\mathbf{W}_k$ and $\mathbf{b}_k$ are trainable parameters, and $\boldsymbol{h}_c$ is the context embedding to summarize the input information. We directly set $\boldsymbol{h}_c = \boldsymbol{h}_0$. The training objective in this stage is to minimize the cross-entropy loss $\mathcal{L}_k$ between the predicted keyword distribution $P_k(w|l, \boldsymbol{x})$ and the ground truth $\hat{P}_k(w|l, \boldsymbol{x})$ as follows:

$$\mathcal{L}_k = -\sum_{i=1}^{|\mathbb{V}|} \hat{P}_k(w_i|l, \boldsymbol{x}) \log P_k(w_i|l, \boldsymbol{x}), \quad (3)$$

where w_i denotes the i-th word in $\mathbb{V}$ and $\hat{P}_k(w|l, \boldsymbol{x})$ is an one-hot vector over $\mathbb{V}$. We do not decode a keyword sequence explicitly (Yao et al., 2019) but generate stories directly based on the keyword distribution $P_k(w|l, \boldsymbol{x})$ to avoid introducing extra exposure bias (He et al., 2019).

3.3 Generation

We employ a left-to-right decoder to generate a story conditioned upon the input and the predicted keyword distribution. The training objective in this stage is to minimize the negative log-likelihood $\mathcal{L}_{st}$ of the ground truth stories:

$$\mathcal{L}_{st} = -\sum_{t=1}^{m} \log P(y_t|l, \boldsymbol{x}, y_{<t}). \quad (4)$$

We derive $P(y_t|l, \boldsymbol{x}, y_{<t})$ by explicitly combining the stylistic keyword distribution into the decoding process as follows:

$$P(y_t|l, \boldsymbol{x}, y_{<t}) = P_l(y_t|l, \boldsymbol{x}, y_{<t}) \cdot (1 - \boldsymbol{g}_t) + P_k(y_t|l, \boldsymbol{x}) \cdot \boldsymbol{g}_t, \quad (5)$$

$$P_l(y_t|l, \boldsymbol{x}, y_{<t}) = \text{softmax}(\mathbf{W}_s \boldsymbol{s}_t + \mathbf{b}_s), \quad (6)$$

$$\boldsymbol{s}_t = \texttt{Dec}(y_{<t}, \{\boldsymbol{h}_i\}_{i=0}^{n}), \quad (7)$$

where $\mathbf{W}_s$ and $\mathbf{b}_s$ are trainable parameters, P_l is a distribution over $\mathbb{V}$ without conditioning on the predicted keywords, and $\boldsymbol{g}_t \in \mathbb{R}^{|\mathbb{V}|}$ is a gate vector indicating the weight of the keyword distribution P_k. We compute $\boldsymbol{g}_t$ as follows:

$$\boldsymbol{g}_t = \text{sigmoid}(\mathbf{W}_g[\boldsymbol{r}_t; \boldsymbol{s}_t] + \boldsymbol{b}_g), \quad (8)$$

$$\boldsymbol{r}_t = \boldsymbol{W}_r P_k(y_t|l, \boldsymbol{x}) + \boldsymbol{b}_r, \quad (9)$$

where $\boldsymbol{W}_g$, $\boldsymbol{b}_g$, $\boldsymbol{W}_r$ and $\boldsymbol{b}_r$ are trainable parameters. In summary, the final training objective $\mathcal{L}$ of our model is derived as follows:

$$\mathcal{L} = \mathcal{L}_{st} + \alpha \cdot \mathcal{L}_k. \quad (10)$$

where α is an adjustable scale factor.

4 Experimental Setup

4.1 Dataset

We conduct the experiments on the ROCStories corpus (Mostafazadeh et al., 2016), which contains 98,159 five-sentence stories. We randomly split ROCStories by 8:1:1 for training/validation/test, respectively. The average number of words in the input (the first sentence) and the output (the last four sentences) are 9.1 and 40.8, respectively. Besides, we follow Guan et al. (2020) to delexicalize stories in the dataset by masking all the male /female/neutral names with $\langle$MALE$\rangle$/ $\langle$FEMALE$\rangle$/ $\langle$NEUTRAL$\rangle$ to achieve better generalization.

4.2 Style Annotation

We extract stylistic keywords from stories in the dataset and assign a style label for each story according to the distribution of stylistic keywords.

Stylistic Keywords We use NRCLex and NLTK to extract stylistic keywords. NRCLex maps each word in a story to its underlying emotion labels according to a word-emotion lexicon (e.g., *"favorite"* → *"joy"*). We select the words with following emotion labels: *"fear"*, *"anger"*, *"surprise"*, *"sadness"*, *"disgust"* and *"joy"*, as the keywords for the emotion-driven style. Besides, we use NLTK to extract verbs as keywords for the event-driven style. We filter out the stop words and common verbs with bottom ten IDF[2] (e.g., *"is"*, *"have"*) from the extracted verbs. Intuitively, the more stylistic keywords of some style a story has, the more consistent it is with that style. Therefore, we propose to compare the numbers of keywords for different styles for style annotation.

Normalized Numbers of Keywords Let N_s denote the number of keywords for style s in a story. We assume N_s is a random variable, and follows a Gaussian distribution $\mathcal{N}(\mu_s, \sigma_s^2)$, where μ_s and σ_s are the mean and standard deviation computed on the training set. Given a story which contains n_s keywords for style s, we normalize n_s to $n'_s = P(N_s \leqslant n_s) \in [0, 1]$ for fair comparison between keywords for different styles.

Styles	Training	Validation	Test
Emotion-driven	17.7%	18.0%	17.9%
Event-driven	17.6%	17.0%	17.5%
Others	64.7%	65.0%	64.6%

Table 1: Distribution of stories annotated with different style for the training/validation/test set.

Annotation We annotate the style label l for a given story by comparing its n'_{emo} and n'_{eve}, which refer to the normalized numbers of keywords for emotion-driven and event-driven styles, respectively. We annotate the story with $\langle$emo$\rangle$ if n'_{emo} is higher than n'_{eve}, and $\langle$eve$\rangle$ otherwise. However, if both n'_{emo} and n'_{eve} are lower than τ_1, or $|n'_{\text{emo}} - n'_{\text{eve}}| < \tau_2$, we annotate the story with $\langle$other$\rangle$ since there is no significant tendency to any styles. τ_1 and τ_2 are hyper-parameters, which are set to 0.7 and 0.3, respectively. For stories labeled with $\langle$other$\rangle$, we select five words as the stylistic keywords from those keywords for emotion-driven and event-driven styles. Table 1 shows the stylistic distribution of the dataset.

[2] Inverse Document Frequency (IDF) is statistically analyzed on the stems of all the extracted keywords by NLTK.

4.3 Baselines and Experiment Settings

We compare our model with GPT-2 (Radford et al., 2019) and BART (Lewis et al., 2020) as baselines. We fine-tune the baselines on ROCStories with the style tokens and the beginnings as input.

We build our model based on BART. Our approach can easily adapt to other pre-trained models such as BERT. We set the scale factor in Equation 10 to 0.2. For all models, We generate stories using top-k sampling (Fan et al., 2018) with $k = 50$ and a softmax temperature of 0.8.

4.4 Automatic Evaluation

Evaluation Metrics We use the following metrics for automatic evaluation: **(1) Perplexity (PPL)**. Since the automatically annotated style labels may contain innate bias, we do not calculate the perplexity conditioned on the annotated styles for the stories in the test set. Instead, we calculate the perplexity of a model for each sample conditioned on two styles (emotion-driven and event-driven), respectively, and then get the perplexity on the entire test set by averaging the smaller perplexity for each sample.**(2) BLEU (B-n)** (Papineni et al., 2002)**:** The metric evaluates n-gram overlap ($n = 1, 2$). For each beginning in the test set, we generate two stories conditioned on two styles, respectively. Then we calculate the BLEU score on the test set by averaging the higher BLEU with the reference story for each sample.**(3) Distinct (D-n)** (Li et al., 2016)**:** The metric measures the generation diversity with the percentage of unique n-grams ($n = 1, 2$). **(4) Numbers of Stylistic Keywords (Number):** We use the average n' (described in §4.2) to evaluate how many consistent stylistic keywords the generated stories have. **(5) Lexical Style Consistency (LSC):** We calculate the percentage of the stories annotated with the consistent style in all generated stories using the annotation strategy described in §4.2. **(6) Semantic Style Consistency (SSC):** It is a learnable automatic metric (Guan and Huang, 2020). We fine-tune $\mathrm{BERT}_{\mathrm{BASE}}$ on the training set as a classifier to distinguish whether a story is emotion-driven, event-driven, or others with the automatic labels as the golden truth. For each style, we regard the average classification score on the style to measure the style consistency. Table 2 shows the accuracy and F1-Scores of the BERT model on the test set.

Results We show the evaluation results of PPL and BLEU in Table 3. Note that we do not provide PPL for GPT-2 since it does not adopt the same vocabulary used in BART. We can see that our model has lower perplexity and higher word overlap with the human-written stories than baselines.

Accuracy	F1-Score		
	Emotion-Driven	Event-Driven	Other
89.7%	0.863	0.838	0.922

Table 2: Accuracy and F1-Scores for each class of the BERT used in SSC.

Models	PPL ↓	B-1 ↑	B-2 ↑
GPT-2	N/A	32.8	16.1
BART	11.72	33.2	16.6
Ours	**11.29**	**33.8**	**17.1**

Table 3: Automatic evaluation results on the entire test set. The Best results are highlighted in **bold**. ↓/↑ indicates the lower/higher, the better.

We present the results of diversity and style consistency on the generated stories with different specified styles in Table 4. Our model achieves comparable diversity with baselines, generates more keywords of the specified styles, and outperforms baselines in both lexical and semantic style consistency by a large margin.

Models	D-1 ↑	D-2 ↑	Number ↑	LSC↑	SSC ↑
Emotion-driven Style					
GPT-2	0.679	0.924	0.454	0.243	0.201
BART	**0.701**	**0.952**	0.538	0.366	0.298
Ours	0.697	**0.952**	**0.623**	**0.474**	**0.371**
Event-driven Style					
GPT-2	0.675	0.925	0.375	0.107	0.088
BART	0.697	0.954	0.460	0.162	0.129
Ours	**0.698**	**0.955**	**0.591**	**0.309**	**0.293**

Table 4: Automatic evaluation results. The best results are highlighted in **bold**.

4.5 Manual Evaluation

We conduct a pairwise comparison between our model and baselines. We randomly generate 100 stories from the test set for each style and model. For each pair of stories (one by ours, and the other by a baseline), we hire three annotators to give a preference (*win*, *lose* and *tie*) in terms of fluency, coherence, and style consistency. We adopt majority voting to make the final decisions among the annotators. We resort to Amazon Mechanical Turk for manual annotation. As shown in Table 5, all the results show moderate $(0.4 \leqslant \kappa \leqslant 0.6)$

Styles	Models	Fluency				Coherence				Style Consistency			
		Win(%)	Lose(%)	Tie(%)	κ	Win(%)	Lose(%)	Tie(%)	κ	Win(%)	Lose(%)	Tie(%)	κ
Emotion-driven	**Ours** vs. **GPT-2**	37.0**	19.0	44.0	0.751	52.0**	25.0	23.0	0.803	39.0*	22.0	39.0	0.773
	Ours vs. **BART**	51.0**	22.0	27.0	0.672	52.0**	16.0	32.0	0.591	50.0*	31.0	19.0	0.640
Event-driven	**Ours** vs. **GPT-2**	40.0*	25.0	35.0	0.754	50.0*	30.0	20.0	0.735	54.0**	28.0	18.0	0.726
	Ours vs. **BART**	43.0*	25.0	32.0	0.644	45.0**	22.0	33.0	0.635	48.0*	33.0	19.0	0.612

Table 5: Manual evaluation results. The scores indicates the percentage of *win*, *lose* or *tie* when comparing our model with a baseline. κ denotes Fleiss' Kappa (Fleiss, 1971) to measure the inter-annotator agreement. * and ** mean p-value<0.05 and p-value<0.01 (Wilcoxon signed-rank test), respectively.

Leading Context: Bob has a girl friend.	
Emotion-driven Style	
GPT-2	She wants to **take** a trip to Hawaii. She **goes** on vacation. She is at the beach. It is a *pretty* day.
BART	One day, Bob **saw** a *cute* necklace on the sidewalk. Bob decided to **buy** it. After buying it, Bob *loves* it. Bob *likes* the necklace.
Ours	He is really *nervous* about her feeling around her. His girlfriend is very *protective* . Bob gets along *great* with her. Bob has a *wonderful* time with his girlfriend.
Event-driven Style	
GPT-2	She *likes* her hair. She **takes** a few pictures of her friend's hair. He **takes** a picture of her hair and **posts** it. She *likes* it very much and she is *happy* .
BART	He knew he was always going to be *mean* to her. After a while Bob **realized** that he was being *annoying* . He had to **leave** his job and **walk** . Now he has a new girlfriend and a new job.
Ours	He has been **talking** to her all day. She stopped **listening** to him now. One day, she **says** his name and **walked** away. He decided to **break** up with her in another place.

Table 6: Generated stories by different models with different specified styles. *Emotion-related* keywords and **event-related** keywords are highlighted in italic and bold, respectively.

or substantial ($0.6 \leqslant \kappa \leqslant 0.8$) agreement, and our model outperforms baselines significantly in fluency, coherence, and style consistency.

5 Case Study

Table 6 shows several generated cases. We generate the stories using different models given the same leading context and specified style. For the *emotion-driven* style, our model can generate various emotional keywords (e.g., *"nervous"*, *"protective"*, *"great"*, and *"wonderful"*) and focus more on shaping the characters' personality. For the *event-driven* style, our model can generate fluent stories with a reasonable event sequence. In comparison, baselines tend to confuse the two styles. For example, the stories generated by the baselines for the event-driven style still contain many emotional keywords (e.g., *"likes"*, *"annoying"*). Besides, for the emotion-driven style, the baselines generate fewer and repetitive emotional keywords. Furthermore, the baselines may suffer from more severe repetition (e.g., "take a picture") than our model. And the baselines sometimes mix up or neglect some characters (e.g., GPT-2 and BART only cover one of *"Bob"* and *"his girlfriend"* but neglect the other one for the *emotion-driven* style). In summary, our model can generate more coherent stories with specified styles than the baselines.

6 Conclusion

We present a pilot study on a new task, stylized story generation. We define story *style* with respect to *emotion* and *event*, and propose a generation model which conditions on planned stylistic keywords. Comparative experiments with strong baselines show the promising results of the proposed model. Our work can inspire further research in this new direction.

Acknowledgements

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. We thank THUNUS NExT Joint-Lab for the support.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Alex Boyd, Raul Puri, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Large scale multi-actor generative dialog modeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 66–84.

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.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2650–2660. Association for Computational Linguistics.

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

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.

Jian Guan, Fei Huang, Minlie Huang, Zhihao Zhao, and Xiaoyan Zhu. 2020. A knowledge-enhanced pretraining model for commonsense story generation. *Trans. Assoc. Comput. 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.

Tianxing He, Jingzhao Zhang, Zhiming Zhou, and James R. Glass. 2019. Quantifying exposure bias for neural language generation. *CoRR*, abs/1905.10617.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596.

Qiuyuan Huang, Zhe Gan, Asli Celikyilmaz, Dapeng Wu, Jianfeng Wang, and Xiaodong He. 2019. Hierarchically structured reinforcement learning for topically coherent visual story generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8465–8472.

Dongyeop Kang and Eduard Hovy. 2020. Self-supervised text planning for paragraph completion task. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6533–6543.

Pei Ke, Jian Guan, Minlie Huang, and Xiaoyan Zhu. 2018. Generating informative responses with controlled sentence function. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1499–1508, Melbourne, Australia. Association for Computational Linguistics.

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*, pages 7871–7880.

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

Yuan Li, Chunyuan Li, Yizhe Zhang, Xiujun Li, Guoqing Zheng, Lawrence Carin, and Jianfeng Gao. 2020. Complementary auxiliary classifiers for label-conditional text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8303–8310.

Saif M. Mohammad. 2020. Practical and ethical considerations in the effective use of emotion and sentiment lexicons. *CoRR*, abs/2011.03492.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

Lili Mou and Olga Vechtomova. 2020. Stylized text generation: Approaches and applications. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 19–22, Online. Association for Computational Linguistics.

Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. *Transactions of the Association for Computational Linguistics*, 6:373–389.

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

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

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric Xing, and Zhiting Hu. 2019. Target-guided open-domain conversation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5624–5634, Florence, Italy. Association for Computational Linguistics.

Alexey Tikhonov and Ivan P Yamshchikov. 2018. Guess who? multilingual approach for the automated generation of author-stylized poetry. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 787–794. IEEE.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Jingjing Xu, Xuancheng Ren, Yi Zhang, Qi Zeng, Xiaoyan Cai, and Xu Sun. 2018. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4306–4315. Association for Computational Linguistics.

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

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.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213.