Rich Syntactic and Semantic Information Helps Unsupervised Text Style Transfer

Hongyu Gong[1], **Linfeng Song**[2], **Suma Bhat**[1]
[1]University of Illinois at Urbana-Champaign, IL, USA
[2]Tencent AI Lab, Bellevue, WA, USA
hgong6@illinois.edu
lfsong@tencent.com
spbhat2@illinois.edu

Abstract

Text style transfer aims to change an input sentence to an output sentence by changing its text style while preserving the content. Previous efforts on unsupervised text style transfer only use the surface features of words and sentences. As a result, the transferred sentences may either have inaccurate or missing information compared to the inputs. We address this issue by explicitly enriching the inputs via syntactic and semantic structures, from which richer features are then extracted to better capture the original information. Experiments on two text-style-transfer tasks show that our approach improves the content preservation of a strong unsupervised baseline model thereby demonstrating improved transfer performance.

1 Introduction

Text style transfer aims at rephrasing an input sentence as an output sentence in a target style (e.g. sentiment change from negative to positive), while preserving the original content. The utility of text style transfer has been shown in applications such as personalized response generation (Zhou et al., 2017; Niu and Bansal, 2018) and poetry generation (Yang et al., 2018a). In particular, unsupervised style transfer has been extensively explored due to a lack of parallel corpora (Hu et al., 2017; Shen et al., 2017; Yang et al., 2018b; John et al., 2019).

Most previous efforts on unsupervised text style transfer have relied on separating the content from the style of input texts. This was achieved via a transfer model with multiple decoders (Fu et al., 2018) or extra auxiliary losses (John et al., 2019) to learn the disentangled representation vectors for content and style respectively. The content vector was later combined with the vector of the desired style to produce the output.

While these prior studies have successfully demonstrated the capability to adapt input texts to the desired style, the proposed approaches suffer from a significant loss of semantic content. For instance, when a model takes as input "*The lounge is very outdated*" to generate "*The food is delicious*", where the key information of the input (*The lounge*) is missing in the output, the rendering of the transfer becomes irrelevant.

To alleviate this problem, some studies sought to explicitly replace the words related to the stylistic aspect (e.g., sentiment) and retain other content-related words (Xu et al., 2018a; Li et al., 2018; Wu et al., 2019; Madaan et al., 2020). However, their success was limited to specific situations where the style words are explicit. When the negative sentiment is expressed implicitly, as in "*The only thing I was offered was a free dessert!!!*" this approach cannot have the desired effect.

A second direction to address the semantic loss has been the use of back-translation (Prabhumoye et al., 2018; Lample et al., 2019; He et al., 2020) and/or reinforcement learning to preserve the input content (Gong et al., 2019; Luo et al., 2019). Generally, these techniques involve a more complex model training, adding another layer of difficulty to obtain a strong style transfer model with robust performance.

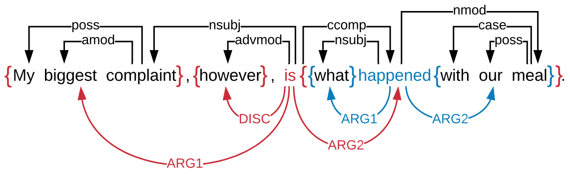

Figure 1: A parsed sentence with dependency arcs (top) and edges that show semantic roles (bottom).

In this paper, we propose a new idea to preserve the semantics by *highlighting* the core information that should be preserved. This is performed during the input text encoding stage. As shown in Figure 1, we propose to include two types of structures—

Proceedings of The 13th International Conference on Natural Language Generation, pages 113–119,
Dublin, Ireland, 15-18 December, 2020.

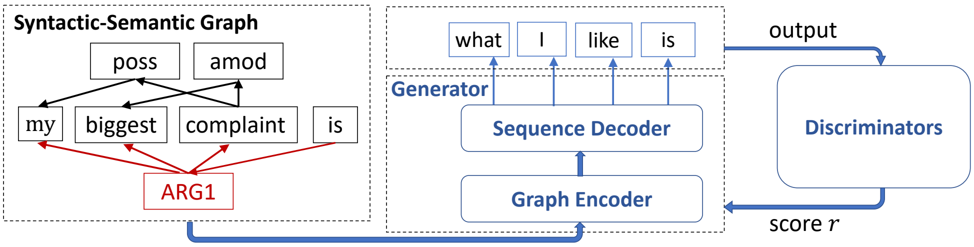

Figure 2: Text style transfer model with syntactic-semantic graph.

semantic roles and dependency trees—to represent the core semantic and syntactic information. Semantic roles directly capture the key information in a sentence, such as its subject and object. In the example of Figure 1, "my biggest complaint" is identified as the subject of "is". Meanwhile, a dependency tree captures finer-grained word-level relations. The relations "poss" and "amod" (possessive pronoun and adjectival modifier respectively) in this example reveal the syntactic structure within the phrase "my biggest complaint".

As a different way of encoding the input, we consider a sentence, along with its dependency and semantic role annotations, as a graph. We then use a Graph Neural Network (GNN) (Marcheggiani and Titov, 2017) to encode the sentence, noting the reported success of GNNs in representing syntactic and semantic structures (Marcheggiani and Titov, 2017; Song et al., 2018; Beck et al., 2018; Xu et al., 2018b). For the overall architecture, the proposed GNN layers can be stacked onto (or replace) the encoder of an existing system. Previous advances on style transfer were achieved by new designs of the decoder or the learning framework (Shen et al., 2017; Fu et al., 2018). Our approach can be considered to be orthogonal to these previous designs.

Preliminary experiments with available text style transfer models on benchmark datasets validate the utility of our input-encoding approach for preserving input semantic information when compared with a strong baseline (Gong et al., 2019) without such an encoding. We include the details of our implementation in the supplementary material.

2 Baseline

The baseline we consider for this study is a recent model (Gong et al., 2019) based on a generator-discriminator framework. The generator transfers sentences from the source style to the target style and is in a tight feedback loop with a set of discriminators that evaluate the quality of the transferred sentences. The reported results revealed its competitive style transfer performance on available benchmark datasets.

Generator. The generator is a typical seq2seq model with a sequence encoder and a sequence decoder (Bahdanau et al., 2015). The encoder adopts a Gated Recurrent Unit (GRU) to take in an input sentence with words $\{x_1, \dots, x_N\}$ and produce the encoder states $\{\mathbf{h}_1, ..., \mathbf{h}_N\}$ sequentially. The sequence decoder is another GRU with the attention mechanism (Luong et al., 2015). At each time step t, the decoder updates its hidden state $\mathbf{s_t}$ with the target token generated at time $t-1$. It then predicts the current token y_t using the current decoder state and the weighted sum of the encoder states, where the weights are produced by the attention mechanism.

Discriminators. Three discriminators are included, each serving to judge one aspect of the quality of the generated target sentences from among meaning preservation, transfer strength, and fluency. The meaning preservation is evaluated using the word mover's distance (Kusner et al., 2015), which calculates the similarity score r^{sem}. The style discriminator predicts the likelihood r^{style} of a generated sentence in the target style as the style quality. Moreover, a pre-trained neural language model evaluates the fluency by estimating the log-probability r^{lm} of each generated sentence. The overall evaluation score that served as the feedback to the generator, r, was a weighted average of the three evaluation metrics to account for the fact that they may not be in the same scale.

$$r = \alpha r^{\text{sem}} + \beta r^{\text{style}} + \eta r^{\text{lm}}, \qquad (1)$$

where α, β and γ are weighting coefficients.

Reinforcement learning (RL). RL trains the generator with the feedback received from the discriminators (scores given by the discriminators to its generated sentences) (Gong et al., 2020). Under the RL framework, generating a target sentence

	Negative-to-Positive				Positive-to-Negative			
	Semantic	Style	Overall	Perplexity	Semantic	Style	Overall	Perplexity
CA	0.919	0.842	0.440	119	0.914	0.821	0.433	148
MD	0.831	0.972	0.448	**104**	0.824	0.864	0.422	**92**
RL	0.864	0.964	0.456	126	0.842	0.952	0.447	101
GT	0.896	0.943	**0.459**	126	0.863	0.974	**0.458**	99

Table 1: Model performance of style transfer on Yelp dataset (GT is RL with the proposed enhancement).

was formulated as making a sequence of actions, where an action is a token produced at a decoding step. Taking the decoding step t as an example, the decoder state $\mathbf{s_t}$ contains the information of the input source sentence and the partial target sentence already generated by the model. An action a_t is the generated token of the target sentence at step t, and a reward Q_t is defined to reflect how good the action is.

The reward $Q(\mathbf{s}_t, a_t)$ of taking action a_t in state $\mathbf{s}_t$ was estimated by sampling complete target sentences with their first $t-1$ tokens fixed. With $r(\mathbf{s}_\tau, a_\tau)$ as the average score over the sampled sentences with the first τ tokens fixed, the reward was defined as

$$Q(\mathbf{s}_t, a_t) = \sum_{\tau=t}^{T} \gamma^{\tau-t}(r(\mathbf{s}_\tau, a_\tau) - r(\mathbf{s}_{\tau-1}, a_{\tau-1})), \tag{2}$$

where γ was a discounting factor set to 0.9.

The generator was parameterized by θ and denoted as G_θ. The total reward $J(G_\theta)$ of generating a target sentence of T tokens was

$$J(G_\theta) = \sum_{t=1}^{T} \sum_{a_t \in V} p_\theta(a_t|\mathbf{s}_t) Q(\mathbf{s}_t, a_t), \tag{3}$$

where $p_\theta(a_t|x_t)$ is the probability for producing the token a_t in state $\mathbf{s}_t$. Policy gradient was applied to update the generator parameters θ with $J(G_\theta)$.

3 Model

We propose to enrich input sentences with syntactic and semantic structures, and to encode the resulting graphs using a Graph Neural Network (GNN). For a fair comparison with the baseline model, we replaced the GRU encoder of the baseline generator with our GNN encoder and kept the other modules unchanged. Our proposed model is called the Graph Transfer (GT) model. Figure 2 demonstrates our style transfer model with the proposed graph encoder. The input sentence is first parsed as a syntactic-semantic graph. The graph encoder encodes rich information of the graph into dense representations, which are then fed to a sequence decoder for sentence generation. The transferred sentences are sent to the discriminators for evaluation, and the scores of these sentences serve as the training signals for the generator.

3.1 Syntactic-Semantic Graph

To jointly leverage information from both syntactic and semantic structures that were automatically produced by off-the-shelf toolkits, We include both into a syntactic-semantic graph,. In particular, the graph nodes are the words in the sentence and the relation tags (e.g. "ARG1") between word pairs. Directed edges are assigned to node pairs, and the direction is determined by the parsers. Using the parse in Figure 1 as an example, we see the relation tag "*ARG1*" connects the phrase "my biggest complaint" and the verb "is". In the syntactic-semantic graph shown in Figure 2, we add an edge from "*is*" to "*ARG1*" , and also add edges from "*ARG1*" to "*my*", "*biggest*" and "*complaint*" respectively.

3.2 Graph Encoding with GNN

We use a GNN to encode our syntactic-semantic graphs. It adopts an iterative message-passing mechanism, where directly connected graph nodes pass information to each other for their state updates. As a result, these graph nodes absorb rich contextual information. Taking node i in iteration k as an example, the node first collects information along incoming edges, and obtains its forward neighbor representation $\mathbf{m}_{k,i}^{\text{fwd}}$

$$\mathbf{m}_{k,i}^{\text{fwd}} = \frac{1}{|N_i^{\text{fwd}}|} \sum_{j \in N_i^{\text{fwd}}} \mathbf{W}_k^{\text{fwd}} \mathbf{h}_{k-1,j} + \mathbf{b}_k^{\text{fwd}}, \tag{4}$$

where N_i^{fwd} is the set of incoming neighbors for node i, and $\mathbf{W}_k^{\text{fwd}}$ and $\mathbf{b}_k^{\text{fwd}}$ are model parameters. Next, the forward hidden state of node i is generated using its neighbor representations:

$$\mathbf{h}_{k,i}^{\text{fwd}} = \text{ReLU}[\tilde{\mathbf{W}}_k^{\text{fwd}} \mathbf{h}_{k-1,i}^{\text{fwd}} + \tilde{\mathbf{b}}_k^{\text{fwd}}, \mathbf{m}_{k,i}^{\text{fwd}}], \tag{5}$$

	Negative-to-Positive	Positive-to-Negative
Orig	unfortunately , our experience did not live up to others' experiences .	but still love this place every time regardless !
CA	unfortunately , our food has nothing to be some great restaurants .	but avoid this place is just every time time time !
MD	pizza is always good , and the staff is great .	staff is bad , we have all the food back .
RL	unfortunately , their experience is always to be coming back	but i would not get this every time time !
GT	however , our experience did such an incredible job .	this place has gone down hill .

Table 2: Examples of transferred sentences by different models.

where matrix $\tilde{\mathbf{W}}_k^{\text{fwd}}$ and bias $\tilde{\mathbf{b}}_k^{\text{fwd}}$ are model parameters. Similarly, its backward hidden state $\mathbf{h}_{k,i}^{\text{bwd}}$ is calculated from all outgoing neighbors, and the overall hidden state $\mathbf{h}_{k,i} = [\mathbf{h}_{k,i}^{\text{fwd}}, \mathbf{h}_{k,i}^{\text{bwd}}]$ is their concatenation. After a total number of K iterations, each node collects the information of all its neighbors within a distance of K, and they are used as the final encoder states (i.e. $\mathbf{h}_i$).

In the encoding stage, a sequence encoder such as a GRU network only allows information to be propagated sequentially within a sentence. Because of this, the encoding process could result in an information loss when long-range dependencies are present. Conversely, a GNN encoder allows a direct interaction between distant words that are semantically or syntactically related (Zhang et al., 2018), thereby serving the style transfer process.

4 Experiments

Dataset. We focus on the task of sentiment transfer, retaining the setting of the Yelp dataset as (Shen et al., 2017). The dataset contains $176,878$ negative and $267,314$ positive sentences for training, $25,278$ negative and $38,205$ positive sentences for development, and $50,278$ negative and $76,392$ positive sentences for testing. We construct syntactic-semantic graphs with the Stanford dependency parser (Manning et al., 2014) and the semantic role labeler of the AllenNLP (Gardner et al., 2018).

Baselines. We compared our model (*GT*) with three state-of-the-art models for text style transfer:
(1) Reinforcement learning based model (RL). *RL* is the baseline summarized in Section 2.
(2) Cross alignment model (CA). *CA* transfers the text style by combining content representation with style information (Shen et al., 2017).
(3) Multi-decoder model (MD). *MD* disentangles content from style, and adopts multiple decoders to produce outputs of various styles (Fu et al., 2018).

Implementation. We include the implementation details of our model in Appendix A.1.

4.1 Automatic Evaluation

Evaluation metrics. We use the same automatic evaluation metrics from prior studies to evaluate the outputs in terms of semantic preservation, style transfer strength and fluency. Semantic preservation s_{sem} is measured by a sentence-similarity metric based on pre-trained GloVe embeddings (Fu et al., 2018). Transfer strength s_{style} is measured by the percentage of generated sentences that can be correctly classified into the target style by a pre-trained style classifier (Fu et al., 2018). Considering the trade-off between semantic preservation and transfer strength, Fu et al. (2018) proposed an overall score $s_{\text{overall}} = \frac{s_{\text{sem}} * s_{\text{style}}}{s_{\text{sem}} + s_{\text{style}}}$, with higher scores indicating better generation quality. Similar to Gong et al. (2019), we use the perplexity estimated by a pre-trained RNN-based language model to quantify fluency, and lower perplexity indicates higher fluency.

Results. Table 1 shows the metric scores of each model for the transfer in two directions. Looking at the overall score, our model (*GT*) outperforms all the models for positive-to-negative transfer. It achieves a comparable performance with the *RL* baseline for negative-to-positive transfer, while still outperforming the other models. In particular, *GT* shows a largely improved semantic score compared to *RL* on both tasks. In terms of perplexity, *GT* is comparable to *RL*.

model	Semantic	Style	Overall	Perplexity
Both	0.863	0.974	0.458	99
w. Dep	0.847	0.957	0.449	91
w. SRL	0.891	0.81	0.424	124

Table 3: Ablation study of style transfer with graph encoder.

Table 2 gives examples of transferred sentences. The row *Orig* shows the original sentences. For both tasks, we notice that *MD* generates irrelevant words, such as “pizza” and “staff”, that were not mentioned in the input. This behavior is reflected in its low semantic score. *CA* fails to transfer the

	Negative-to-Positve			Positive-to-Negative		
	Semantic	Style	Fluency	Semantic	Style	Fluency
CA wins	0.20	0.20	0.20	0.18	0.12	0.20
MD wins	0.08	0.18	0.18	0.10	0.24	0.12
RL wins	0.16	0.14	0.20	0.14	0.20	0.18
GT wins	**0.42**	**0.26**	**0.24**	**0.34**	**0.26**	**0.22**
Tie	0.14	0.22	0.18	0.24	0.18	0.28

Table 4: Percentage of model wins and ties in human evaluation.

sentiment for both cases, which is consistent with its low style score. *RL* is successful in changing the sentiment for both cases, but it largely misses input content, such as “this place”. We note that the addition of syntactic and semantic information helps to preserve most of the original content, reflected in *GT* outperforming *MD* in content preservation by a large margin. Both our model (GT) and *CA* have high semantic scores, but our model does better than *CA* in style changing. More transferred examples are shown in Appendix A.2.

Ablation analysis. We incorporate syntactic and semantic information into style transfer by both dependency parsing and semantic role labeling. To explore the benefits of each part, we performed an ablation study in the task of positive-to-negative transfer. Table 3 compares the performance of models with only dependency parsing (Dep) and with only semantic role labeling (SRL).

The graph encoder with only syntactic information from dependency parser does well in changing text style while falling behind in content preservation. It achieves lower perplexity (i.e., higher fluency) than the encoder with both dependency parsing and semantic role labeling. A possible explanation is that the syntactic information provided by the dependency parser plays an important role in style and grammaticality.

The encoder with only semantic role labeling trades off its style transfer strength for content preservation. This is consistent with our linguistic intuition that semantic roles centrally capture sentence meaning.

4.2 Human Evaluation

Human evaluation of the output complements the evaluation using automatic metrics for transfer quality. Accordingly, we sampled 50 positive and 50 negative sentences from the Yelp corpus, and corresponding outputs from all models. Two raters with native-like English proficiency selected the best sentence(s) among all candidates for the dimensions of content preservation, transfer strength and fluency separately. The best sentence(s) received a score of 1, and the others received a 0. Ties were allowed, i.e., multiple transferred outputs could receive a 1 for the same input. Each output was scored along each dimension by averaging its scores from the two raters.

Table 4 reports the percentage of times when each model won and when multiple models tied for positive-to-negative and negative-to-positive transfer respectively. We note that *GT* outperforms all baselines in all evaluation aspects. Further discussions of the human evaluation results are available in Appendix A.3.

5 Conclusion

We empirically demonstrated how including rich syntactic and semantic information can help to preserve content during text style transfer. Toward this, we compared competitive style transfer models with and without enriching inputs via syntactic and semantic structures on a benchmark dataset. We found that instead to an input text of a sequence of words alone, encoding the input sentence’s syntactic-semantic graph via a graph neural network serves to explicitly highlight the sentence’s core meaning.

In this work, we have focused on the generator to improve the performance of text style transfer. One of our future directions is to incorporate better semantic metrics into the discriminators so that the training loss could measure the preservation of semantic information more accurately.

Acknowledgments

This work was supported by the IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR)—a research collaboration as part of the IBM AI Horizons Network. We would like to thank the anonymous reviewers for their constructive comments and suggestions. We also thank Raghavendra Bhat for data annotations.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283, Melbourne, Australia. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.

Hongyu Gong, Suma Bhat, Lingfei Wu, JinJun Xiong, and Wen-mei Hwu. 2019. Reinforcement learning based text style transfer without parallel training corpus. 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 3168–3180.

Hongyu Gong, Yelong Shen, Dian Yu, Jianshu Chen, and Dong Yu. 2020. Recurrent chunking mechanisms for long-text machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6751–6761. Association for Computational Linguistics.

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. 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 424–434. Association for Computational Linguistics.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. 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 1865–1874.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Xu Sun, and Zhifang Sui. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5116–5122.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabás Póczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W. Black, and Shrimai Prabhumoye. 2020. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1869–1881. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1506–1515. 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.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the*

56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 866–876.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, pages 6830–6841.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Chen Wu, Xuancheng Ren, Fuli Luo, and Xu Sun. 2019. A hierarchical reinforced sequence operation method for unsupervised text style transfer. 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 4873–4883. Association for Computational Linguistics.

Jingjing Xu, SUN Xu, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018a. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 979–988.

Kun Xu, Lingfei Wu, Zhiguo Wang, Mo Yu, Liwei Chen, and Vadim Sheinin. 2018b. Exploiting rich syntactic information for semantic parsing with graph-to-sequence model. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 918–924.

Cheng Yang, Maosong Sun, Xiaoyuan Yi, and Wenhao Li. 2018a. Stylistic chinese poetry generation via unsupervised style disentanglement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3960–3969.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018b. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, pages 7287–7298.

Yue Zhang, Qi Liu, and Linfeng Song. 2018. Sentence-state lstm for text representation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 317–327.

Ganbin Zhou, Ping Luo, Rongyu Cao, Fen Lin, Bo Chen, and Qing He. 2017. Mechanism-aware neural machine for dialogue response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*.