# Exploring Non-Autoregressive Text Style Transfer

**Yun Ma**
Department of Computing
The Hong Kong Polytechnic University
mayun371@gmail.com

**Qing Li**
Department of Computing
The Hong Kong Polytechnic University
csqli@comp.polyu.edu.hk

## Abstract

In this paper, we explore Non-AutoRegressive (NAR) decoding for unsupervised text style transfer. We first propose a base NAR model by directly adapting the common training scheme from its AutoRegressive (AR) counterpart. Despite the faster inference speed over the AR model, this NAR model sacrifices its transfer performance due to the lack of conditional dependence between output tokens. To this end, we investigate three techniques, i.e., knowledge distillation, contrastive learning, and iterative decoding, for performance enhancement. Experimental results on two benchmark datasets suggest that, although the base NAR model is generally inferior to AR decoding, their performance gap can be clearly narrowed when empowering NAR decoding with knowledge distillation, contrastive learning, and iterative decoding.

## 1 Introduction

Text Style Transfer (TST) aims at altering a stylistic attribute (e.g., sentiment) of the given text to a target value, without changing the style-agnostic semantics. Due to the difficulty in collecting parallel training corpus, most exiting methods (Shen et al., 2017; Xu et al., 2018; Luo et al., 2019; Zhou et al., 2020) address the task in an unsupervised setting. Through techniques such as auto-encoding, back-translation, or adversarial learning, the task is converted to self-supervised problems and great empirical progress has been made. However, current methods employ autoregressive (AR) decoding which generates each output token conditioned on the previously generated ones, leading to low parallelizability and high latency for pragmatic use.

Recently, non-autoregressive (NAR) decoding has attracted much attention in neural machine translation (NMT) (Gu et al., 2018). NAR decoding eliminates the conditional dependencies among the output tokens and generates them in parallel, thus reducing the decoding time-complexity from $O(T)$ to $O(1)$ for outputs with length $T$. Without modeling the dependency, the advantage on decoding speed comes at the cost of reduced performance. To address this issue, knowledge distillation is employed to transfer the knowledge from AR models to NAR models (Gu et al., 2018). Furthermore, existing works resort to various regularization techniques (Wang et al., 2019) to constrain the output or Semi-AutoRegressive (SAR) decoding (Wang et al., 2018; Ghazvininejad et al., 2019) as a speed-performance tradeoff.

In this paper, we explore NAR decoding for unsupervised TST to enable faster inference with better parallelism. To the best of our knowledge, this is the first work to study NAR models for TST. Firstly, a base NAR model is proposed by directly adapting the widely used training objectives from AR models. As with NMT, the base NAR model underperforms the AR model. To narrow their performance gap, we propose to enhance NAR decoding from three perspectives: the data perspective by knowledge distillation, the regularization perspective by contrastive learning, and the speed-performance tradeoff perspective by iterative decoding. Experimental results on a sentiment transfer dataset and a formality transfer dataset demonstrate integrating these techniques can substantially improve the base NAR model.

## 2 Related Work

**Unsupervised Text Style Transfer.** One branch of methods disentangles the style and content by learning a style-agnostic representation, which can be either a latent vector (Shen et al., 2017; Fu et al., 2018; John et al., 2019; Yi et al., 2020) or a subsequence of the input with the style indicators removed (Li et al., 2018; Xu et al., 2018; Wu et al., 2019b; Sudhakar et al., 2019; Madaan et al., 2020). Another branch of methods (Lample et al., 2019; Dai et al., 2019) inspired by back-translation dynamically creates pseudo-parallel data to gradually



*Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9267–9278
November 7–11, 2021. 

refine the TST model. There are also reinforcement learning based methods (Luo et al., 2019; Wu et al., 2019a; Gong et al., 2019; Liu et al., 2021) which guide the model with different rewards corresponding to the evaluation criteria.

**Non-Autoregressive Decoding.** Since the proposal of NAR decoding in NMT (Gu et al., 2018), follow-up works focus on narrowing its gap with AR decoding while keeping its efficiency. One branch of methods transfers the knowledge from AR models to NAR models by knowledge distillation (Gu et al., 2018), attention alignment (Li et al., 2019), imitation learning (Wei et al., 2019), or reinforcement learning (Shao et al., 2019). Another branch introduces regularization terms such as similarity constraint (Wang et al., 2019), bag-of-ngram difference (Shao et al., 2020), and aligned cross-entropy (Ghazvininejad et al., 2020) to alleviate incorrect translations. Furthermore, SAR decoding is proposed, which seeks a tradeoff between AR and NAR decoding by adding AR layers on NAR models (Wang et al., 2018; Sun et al., 2019; Akoury et al., 2019), iterative refinement (Lee et al., 2018; Ghazvininejad et al., 2019), or insertion-based decoding (Stern et al., 2019; Gu et al., 2019).

## 3 Non-Autoregressive Text Style Transfer

Let $\mathcal{S}$ denote all possible values for a stylistic attribute. A desired TST model $p_\theta(y|x,s)$ with parameters $\theta$ transforms an input text $x$ with source style $s_0 \in \mathcal{S}$ to an output $y$ with a given target style $s \in \mathcal{S}$ while preserving the style-agnostic semantics of $x$. In this section, we first propose a base NAR model (BaseNAR) for unsupervised TST (Section 3.1), then investigate three techniques, i.e., knowledge distillation (Section 3.2), contrastive learning (Section 3.3), and iterative decoding (Section 3.4), to enhance the performance of BaseNAR.

### 3.1 A Base NAR Model

At the core of NAR decoding is the conditional independence among output tokens. Formally,

$$p_\theta(y|x,s) = \prod_{t=1}^{T} p_\theta(y_t|x,s) \tag{1}$$

which, compared with AR decoding, removes the previous generated tokens $y_{<t}$ in the conditional variables for each timestamp.

Our BaseNAR consists of an encoder and a decoder, both adopting Transformer (Vaswani et al., 2017) based architecture. The encoder uses the standard Transformer encoder as with AR models. Following NAR models for NMT (Wang et al., 2019; Shao et al., 2020), the decoder differs from the standard Transformer decoder and AR models by (1) discarding the autoregressive mask in the self-attention layer, (2) incorporating a positional-attention layer, and (3) uniformly mapping the source words as the decoder input[1].

We optimize BaseNAR by three common losses from AR decoding based TST: self-reconstruction, cycle-reconstruction, and style compatibility.

**Self-Reconstruction.** When the target style $s = s_0$, the model is expected to reconstruct $x$. Formally, the self-reconstruction loss minimizes

$$\mathcal{L}_{\text{self}} = -\log p_\theta(x|x,s_0) \tag{2}$$

**Cycle-Reconstruction.** With $y \sim p_\theta(y|x,s)$, the model is expected to reconstruct $x$ when we feed $y$ as the input and $s_0$ as the target style. Formally, the cycle-reconstruction loss minimizes

$$\mathcal{L}_{\text{cycle}} = -\log p_\theta(x|y,s_0) \tag{3}$$

**Style Compatibility.** Let $p_\psi$ denote a pretrained style classifier with parameters $\psi$ to predict the style type for an input text. An output $y \sim p_\theta(y|x,s)$ is expected to be predicted as having style $s$. Formally, the style compatibility loss minimizes

$$\mathcal{L}_{\text{style}} = -\log p_\psi(s|y) \tag{4}$$

The full loss for our BaseNAR model is $\mathcal{L}_{\text{self}} + \mathcal{L}_{\text{cycle}} + \alpha\mathcal{L}_{\text{style}}$, where $\alpha$ is a hyper-parameter.

### 3.2 Knowledge Distillation

In NMT, NAR models (Gu et al., 2018) achieve improved performance by sequence-level knowledge distillation (Kim and Rush, 2016) from AR models. Specifically, a pseudo-parallel corpus is constructed by sampling a translation output from the AR model for each source input in the training dataset. The NAR model is then trained using this pseudo-parallel corpus instead of the original one.

In our text style transfer task, we follow the same scheme in NMT. Suppose $p_\phi(y|x,s)$ is a pretrained AR decoding based TST model with parameters $\phi$. For each input $x$ in the training set, we sample a pseudo-target $\tilde{y} \sim p_\phi(y|x,s)$. The NAR model is then optimized to minimize

$$\mathcal{L}_{\text{kd}} = -\log p_\theta(\tilde{y}|x,s) \tag{5}$$

[1] See Appendix C.1 for more details.

### 3.3 Contrastive Learning

Preliminary experiments show that BaseNAR suffers from the word omission problem. Inspired by Yang et al. (2019), we alleviate the problem by a contrastive learning-based regularization term. Specifically, the model is penalized if the probability for the desired output (*positive sample*) is not larger than that for an output with word omission errors (*negative sample*) by a margin $\eta$. The regularization can be paired with self-reconstruction, cycle-reconstruction, and knowledge distillation. For knowledge distillation, we minimize

$$\begin{aligned}\mathcal{R}_{\text{kd}} = &\max(\log p_\theta(\tilde{y}^*|x,s) + \eta - \log p_\theta(\tilde{y}|x,s), 0) \\ &- \log p_\theta(\tilde{y}^*|x,s) \end{aligned} \tag{6}$$

where $\tilde{y}^*$ is the negative sample generated from current model using length $|\tilde{y}| - 1$, the first term is the hinge loss, and the second term is to avoid instable results in case that minimizing $\log p_\theta(\tilde{y}^*|x,s)$ dominates the training. Regularizing self-reconstruction and cycle-reconstruction follows the same procedure.

### 3.4 Iterative Decoding

Iterative decoding is based on the Conditional Masked Language Model (CMLM) (Ghazvininejad et al., 2019). CMLM masks a subsequence of a given target sequence and predicts this masked subsequence conditioned on the remaining observed tokens and the source input. In our task, $p_\theta(y|x,s)$ is reformulated as $p_\theta(y_{\text{mask}}|x,s,y_{\text{obs}}) = \prod_{t\in y_{\text{mask}}} p_\theta(y_t|x,s,y_{\text{obs}})$, and the loss functions $\mathcal{L}_{\text{self}}, \mathcal{L}_{\text{cycle}}, \mathcal{L}_{\text{kd}}, \mathcal{R}_{\text{kd}}$ are also reformulated accordingly. For instance, $\mathcal{L}_{\text{kd}}$ is reformulated as

$$\mathcal{L}_{\text{CMLM-kd}} = -\log p_\theta(\tilde{y}_{\text{mask}}|x,s,\tilde{y}_{\text{obs}}) \tag{7}$$

Other losses follow a similar reformulation.

During inference, we iteratively refine the prediction by the mask-predict scheme (Ghazvininejad et al., 2019). Given the target length $T$, let $K$ denote the total number of iterations. In each iteration $k \in \{0,\dots,K-1\}$, we obtain $y^k_{\text{obs}}$ by masking $n^k = \lfloor T \cdot \frac{K-k}{K} \rfloor$ tokens with the lowest probabilities $p_t^{k-1}$ in previous prediction $y^{k-1}$, except for $k=0$ where all tokens are masked. The model then repredicts the masked tokens and updates the prediction and probabilities: for masked tokens,

$$\begin{aligned} y_t^k &= \arg\max_w p_\theta(y_t = w|x,s,y^k_{\text{obs}}) \\ p_t^k &= \max_w p_\theta(y_t = w|x,s,y^k_{\text{obs}}) \end{aligned} \tag{8}$$

while for unmasked ones, $y_t^k = y_t^{k-1}$, $p_t^k = p_t^{k-1}$.

## 4 Experiments

### 4.1 Setup

**Dataset.** The models are evaluated on the Yelp dataset (Li et al., 2018) for sentiment transfer and the GYAFC dataset (Rao and Tetreault, 2018) for formality transfer. Yelp consists of business reviews in a positive or negative style, and GYAFC consists of sentences from Yahoo Answers in a formal or informal style. See Appendix A for more details.

**Evaluation Metrics.** The models are evaluated on three aspects: transfer accuracy (TA), content preservation (CP), and language fluency (LF). Both automatic and human evaluation are employed. For automatic evaluation, transfer accuracy is measured by a pretrained style classifier; content preservation is measured by the BLEU score between the model outputs and the human references; and language fluency is measured by the perplexity of model outputs on a pretrained language model. For human evaluation, we sample 100 test samples from both datasets. Three human annotators are invited to score each model output from 1 (worst) to 5 (best) for each aspect. See Appendix B for more details.

**Model Variants.** The following model variants are evaluated: BaseAR (the AR counterpart of BaseNAR), BaseNAR, and variants empowering BaseNAR with knowledge distillation (KD), contrastive learning (CL), and iterative decoding (ID), namely BaseNAR+KD, BaseNAR+CL, BaseNAR+ID, BaseNAR+KD+CL, BaseNAR+KD+ID, BaseNAR+CL+ID, and BaseNAR+KD+CL+ID. See Appendix C for their implementation details[2].

### 4.2 Results and Analysis

Table 1 and Table 2 show the automatic and human evaluation results of different models on Yelp and GYAFC. For comparison, we also provide automatic evaluation results for two state-of-the-art methods:

- DRL (Luo et al., 2019): a reinforcement learning framework which jointly trains the source-to-target and the target-to-source transfer models as a dual task. The framework is optimized by a style reward and a content reward together with the pseudo-parallel data created through back-translation.

[2]We will release our code soon at https://github.com/sunlight-ym/nar_style_transfer.

| Model | DI | ACC ↑ \| TA ↑ | BLEU ↑ \| CP ↑ | PPL ↓ \| LF ↑ |
|---|---|---|---|---|
| DRL (Luo et al., 2019) | $O(T)$ | 89.0 \| - | 55.2 \| - | 48.6 \| - |
| SR (Zhou et al., 2020) | $O(T)$ | 87.6 \| - | 60.4 \| - | 44.8 \| - |
| BaseAR | $O(T)$ | 89.7 \| 4.0 | 54.0 \| 4.2 | 46.2 \| 4.1 † |
| BaseNAR | $O(1)$ | 90.5 \| 4.0 | 53.1 \| 4.1 | 61.0 \| 3.5 |
| BaseNAR+KD | $O(1)$ | 89.1 \| 3.9 | 56.7 \| 4.1 | 53.1 \| 3.6 † |
| BaseNAR+CL | $O(1)$ | 90.2 \| 4.0 | 54.9 \| 4.2 | 57.2 \| 3.6 † |
| BaseNAR+ID | $O(K)$ | 89.3 \| 4.0 | 53.2 \| 4.0 | 55.9 \| 3.7 † |
| BaseNAR+KD+CL | $O(1)$ | 90.9 \| 4.1 | 57.2 \| 4.3 † | 50.3 \| 3.9 † |
| BaseNAR+KD+ID | $O(K)$ | 91.3 \| 4.1 | 57.1 \| 4.3 † | 51.9 \| 3.9 † |
| BaseNAR+CL+ID | $O(K)$ | 89.8 \| 4.0 | 55.3 \| 4.3 † | 53.6 \| 4.0 † |
| BaseNAR+KD+CL+ID | $O(K)$ | 91.4 \| 4.1 | 57.7 \| 4.3 † | 48.9 \| 4.0 † |

Table 1: Automatic and human evaluation results on Yelp. The left side of "|" is the automatic evaluation result and the right side is the human evaluation result. $K = 4$ in our experienments. DI: decoding iterations. †: result significantly better than BaseNAR with p-value < 0.1 for both automatic and human evaluation.

| | DI | ACC ↑ \| TA ↑ | BLEU ↑ \| CP ↑ | PPL ↓ \| LF ↑ |
|---|---|---|---|---|
| DRL (Luo et al., 2019) | $O(T)$ | 73.1 \| - | 41.9 \| - | 86.8 \| - |
| SR (Zhou et al., 2020) | $O(T)$ | 72.4 \| - | 46.0 \| - | 48.9 \| - |
| BaseAR | $O(T)$ | 73.9 \| 3.7 † | 45.5 \| 4.1 † | 50.0 \| 4.0 † |
| BaseNAR | $O(1)$ | 67.7 \| 3.1 | 43.8 \| 3.8 | 65.7 \| 3.3 |
| BaseNAR+KD | $O(1)$ | 68.9 \| 3.4 † | 47.6 \| 4.2 † | 59.8 \| 3.7 † |
| BaseNAR+CL | $O(1)$ | 67.3 \| 3.2 | 43.9 \| 3.9 | 64.8 \| 3.4 |
| BaseNAR+ID | $O(K)$ | 66.1 \| 2.9 | 45.3 \| 4.0 † | 64.9 \| 3.6 |
| BaseNAR+KD+CL | $O(1)$ | 69.4 \| 3.5 † | 47.2 \| 4.3 † | 57.8 \| 3.9 † |
| BaseNAR+KD+ID | $O(K)$ | 70.2 \| 3.5 † | 47.6 \| 4.1 † | 59.7 \| 3.7 † |
| BaseNAR+CL+ID | $O(K)$ | 66.7 \| 2.8 | 45.4 \| 4.2 † | 61.3 \| 3.7 † |
| BaseNAR+KD+CL+ID | $O(K)$ | 71.4 \| 3.6 † | 47.5 \| 4.3 † | 57.6 \| 3.9 † |

Table 2: Automatic and human evaluation results on GYAFC. The left side of "|" is the automatic evaluation result and the right side is the human evaluation result. $K = 4$ in our experienments. DI: decoding iterations. †: result significantly better than BaseNAR with p-value < 0.1 for both automatic and human evaluation.

- SR (Zhou et al., 2020): a sequence-to-sequence model which predicts the output words as well as their relevance to the target style. The word relevance is further utilized to ensure style relevance consistency and content preservation.

On all metrics, BaseAR has comparable performance with the two methods thus serves as a decent baseline to evaluate the NAR models[3]. For our NAR variants, we have the following observations:

First, compared with BaseAR, BaseNAR has a clear disadvantage towards language fluency on Yelp and all metrics on GYAFC, proving that the removed conditional dependencies do degrade model performance.

Second, knowledge distillation can provide a significant improvement over BaseNAR in cases where BaseNAR underperforms BaseAR by a large margin. In particular, on GYAFC which has longer sentences and larger variance but fewer training samples, the relationships among output tokens becomes harder to be inferred, making BaseNAR inferior to BaseAR on all metrics. In this situation, the pseudo-parallel data distilled from an AR model provide considerable complementary knowledge to BaseNAR. Thus the gap between BaseNAR and BaseAR is clearly narrowed. In contrast, on aspects where BaseNAR and BaseAR have limited performance gap, most of the knowledge distilled from an AR model can be already captured by BaseNAR and thus less helpful.

Third, contrastive learning can generally lead to a small improvement on all metrics. While the improvement is quite limited compared with knowledge distillation and can be neglectable especially for automatic evaluation, the benefits turn to be more visible when utilized together with knowledge distillation.

[3]For the two baselines, we conduct the evaluation on the transferred outputs provided by the original papers. Since we independently train our own style classifier to justify the transfer accuracy, the ACC values in Table 1 and Table 2 for the baselines can be different from the results in their papers.

| | Yelp: **positive** → **negative** | GYAFC: **formal** → **informal** |
|---|---|---|
| Input | they were extremely friendly and reasonably priced . | that is if you truly adore them . |
| BaseAR | they were extremely rude and flavorless . | that s if u realy luv them |
| BaseNAR | they were extremely rude over priced . | that is u realy adore them |
| BaseNAR+KD | they were extremely rude and flavorless . | that s if u truly luv them |
| BaseNAR+CL | they were extremely rude over priced . | that s if you realy adore them :p |
| BaseNAR+ID | they were extremely over priced . | that is if you truly adore them |
| BaseNAR+KD+CL | they were extremely rude and flavorless . | that s if u truly luv them |
| BaseNAR+KD+ID | they were extremely rude and flavorless . | that s if u truly them |
| BaseNAR+CL+ID | they were extremely rude and over priced . | that is if you truly adore them |
| BaseNAR+KD+CL+ID | they were extremely rude and flavorless . | that s if u truly luv them |

Table 3: Ouputs of different models on exemplary sentences from Yelp and GYAFC.

Fourth, iterative decoding mainly improves language fluency. However, it can degrade the transfer accuracy on GYAFC. An explanation is that, as the model's prediction also relies on the partial outputs in addition to the source words and target style under iterative decoding, the dependency on the target style is harder to be captured with more conditioned variables. Furthermore, since the masked tokens are selected based on their probabilities, the correctly predicted tokens (which reflect the target style) can be re-masked due to lower probabilities compared with tokens in other positions. Fortunately, the degradation will diminish when knowledge distillation is used.

Table 3 demonstrates the qualitative results for different model variants on samples from Yelp and GYAFC. BaseNAR suffers from the word omission problem, i.e., omitting "and" for the Yelp sample and "if" for the GYAFC sample. This problem can hurt content preservation and language fluency. Further, BaseNAR makes limited changes in producing an informal sequence on GYAFC, explaining its lower transfer accuracy in Table 1 and Table 2. Variants with knowledge distillation can produce results closer to BaseAR. Variants with iterative decoding can generate more fluent results but can do worse in transfer accuracy (e.g., BaseNAR+ID only removes the ending punctuation for the GYAFC sample). Using contrastive learning only brings marginal improvement, e.g., only tackling the word omission on GYAFC but not on Yelp. However, using contrastive learning together with knowledge distillation can generally lead to better results. See Appendix D for more examples.

To summarize, the gap between AR and NAR decoding can be clearly narrowed when the NAR model is enhanced by knowledge distillation, contrastive learning, and optional iterative decoding (without iterative decoding, the model has no significant performance difference but is more efficient).

### 4.3 Discussions

Our work differs from other NAR works by exploring NAR in an unsupervised TST task. Knowledge distillation mainly solves the multimodality problem in the supervised NMT domain while alleviates the problem of lacking ground-truth for training in TST. Iterative decoding is very effective in NMT while has limited help and leads to reduced transfer accuracy in TST. Without ground-truth, the NAR model in TST has more word omission problems, instead of word repetition problems in NMT. The contrastive learning loss, which is not studied in NAR for other tasks, thus is introduced to penalize high log-probability of outputs with word omission. We expect the contrastive learning loss can be adapted to reduce the word repetition problems in other tasks.

## 5 Conclusion

In this paper, we propose NAR decoding for unsupervised text style transfer to pursue faster inference. On top of a base model, we explore how knowledge distillation, contrastive learning, and iterative decoding can narrow the performance gap towards AR decoding.

## Acknowledgements

We thank all the reviewers for their valuable comments and suggestions. The research work described in this paper has been supported by the Hong Kong Research Grants Council under the general research fund scheme (project number: PolyU 11204919).

## References


Nader Akoury, Kalpesh Krishna, and Mohit Iyyer. 2019. Syntactically supervised transformers for faster neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1269–1281, Florence, Italy. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007, Florence, Italy. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 663–670. AAAI Press.

Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. 2020. Aligned cross entropy for non-autoregressive machine translation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3515–3523. PMLR.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. 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 6112–6121, Hong Kong, China. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. 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 11179–11189.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for text style transfer. In *Annual Meeting of the Association for Computational Linguistics*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Guillaume Lample, Sandeep Subramanian, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Zhuohan Li, Zi Lin, Di He, Fei Tian, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Hint-based training for non-autoregressive machine translation. 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 5708–5713, Hong Kong, China. Association for Computational Linguistics.

Yixin Liu, Graham Neubig, and John Wieting. 2021. On learning text style transfer with direct rewards. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4262–4273, Online. Association for Computational Linguistics.

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 2019, Macao, China, August 10-16, 2019*, pages 5116–5122. ijcai.org.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, 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*, pages 1869–1881, Online. Association for Computational Linguistics.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality 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 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Chenze Shao, Yang Feng, Jinchao Zhang, Fandong Meng, Xilin Chen, and Jie Zhou. 2019. Retrieving sequential information for non-autoregressive neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3013–3024, Florence, Italy. Association for Computational Linguistics.

Chenze Shao, Jinchao Zhang, Yang Feng, Fandong Meng, and Jie Zhou. 2020. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 198–205.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6830–6841.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5976–5985. PMLR.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "transforming" delete, retrieve, generate approach for controlled text style transfer. 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 3269–3279, Hong Kong, China. Association for Computational Linguistics.

Zhiqing Sun, Zhuohan Li, Haoqing Wang, Di He, Zi Lin, and Zhi-Hong Deng. 2019. Fast structured decoding for sequence models. 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 3011–3020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Chunqi Wang, Ji Zhang, and Haiqing Chen. 2018. Semi-autoregressive neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 479–488, Brussels, Belgium. Association for Computational Linguistics.

Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Non-autoregressive machine translation with auxiliary regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5377–5384.

Bingzhen Wei, Mingxuan Wang, Hao Zhou, Junyang Lin, and Xu Sun. 2019. Imitation learning for non-autoregressive neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1304–1312, Florence, Italy. Association for Computational Linguistics.

Chen Wu, Xuancheng Ren, Fuli Luo, and Xu Sun. 2019a. A hierarchical reinforced sequence operation method for unsupervised text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4873–4883, Florence, Italy. Association for Computational Linguistics.

Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019b. Mask and infill: Applying

masked language model to sentiment transfer. In *International Joint Conference on Artificial Intelligence*.

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. 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, Melbourne, Australia. Association for Computational Linguistics.

Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. Reducing word omission errors in neural machine translation: A contrastive learning approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6191–6196, Florence, Italy. Association for Computational Linguistics.

Xiaoyuan Yi, Zhenghao Liu, Wenhao Li, and Maosong Sun. 2020. Text style transfer via learning style instance supported latent space. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3801–3807. ijcai.org.

Chulun Zhou, Liangyu Chen, Jiachen Liu, Xinyan Xiao, Jinsong Su, Sheng Guo, and Hua Wu. 2020. Exploring contextual word-level style relevance for unsupervised style transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7135–7144, Online. Association for Computational Linguistics.

# A Dataset Details

For both Yelp and GYAFC, we use the same train/dev/test split as in our state-of-the-art baselines (Luo et al., 2019; Zhou et al., 2020). In particular, for the GYAFC dataset, we use the data in Family & Relationship domain and ignore the available alignment information in the corpus to target at unsupervised text style transfer. Table 4 present the statistics of the Yelp dataset[4] and the GYAFC dataset[5].

| Dataset | Style | Train | Dev | Test |
|---|---|---|---|---|
| Yelp | positive | 270K | 2000 | 500 |
| | negative | 180K | 2000 | 500 |
| GYAFC | formal | 51K | 2247 | 500 |
| | informal | 51K | 2788 | 500 |

Table 4: Dataset statistics.

# B Evaluation Details

## B.1 Automatic Evaluation

**Transfer Accuracy.** The pretrained style classifier is learned on the training corpus of the text style transfer task and follows the TextCNN (Kim, 2014) architecture. The accuracy of the classifier on the test set is 97.8% on Yelp and 88.2% on GYAFC, respectively. A transferred result is regarded as accurate if the pretrained classifier predicts it as having the target style during transfer.

**Content Preservation.** For the Yelp dataset, we use the extended human references provided by Luo et al. (2019). For the GYAFC dataset, we use the human references from the original paper (Rao and Tetreault, 2018). As a result, each test sample is associated with four human references on both datasets. The BLEU score is calculated by the `multi-bleu.perl`[6] script.

**Language Fluency.** The pretrained language model is learned on all text sequences from the training corpus of the text style transfer task and adopts the Gated Recurrent Units (GRU) (Cho et al., 2014) architecture with a single layer and 512 hidden units.

## B.2 Human Evaluation

For each test sample, an annotator is provided with the source input, the target style, and the transferred outputs from all compared models as in Li et al. (2018). The transferred outputs are shuffled for different test samples so that the annotator is unaware of the source model for these outputs.

The annotators are trained by exemplar annotation provided by the authors before evaluation. The final Fleiss' kappa score is 0.79 on Yelp and 0.77 on GYAFC.

# C Implementation Details

## C.1 Model Architecture

All the NAR models adopt the same Transformer based encoder-decoder architecture, and the BaseAR model only differs from the NAR models by the following three differences discussed in Section 3.1.

[4] https://github.com/lijuncen/Sentiment-and-Style-Transfer/tree/master/data/yelp

[5] https://github.com/raosudha89/GYAFC-corpus

[6] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

- The NAR model discards the autoregressive mask in the self-attention layer. Since the NAR model removes the conditional dependency among the output tokens, the causal mask where the position $t$ can only attend to positions $1 \dots t-1$ is no longer needed. Following Gu et al. (2018), we set the masks to prevent a position from attending to itself.

- The NAR model incorporates a positional-attention layer in the decoder, which has been shown to facilitate local reordering in decoding (Gu et al., 2018). The positional-attention layer, placed between the self-attention layer and the inter-attention layer, takes the position embeddings as queries and keys while the decoder states as values.

- The NAR model uniformly maps the source words as the decoder input to enrich the information on the decoder side. Specifically, position $t$ in the decoder input takes the word embedding of the source token in position $i = \text{round}(\frac{T_x}{T_y} \cdot t)$, where $T_x$ and $T_y$ denote the lengths of source input and target output, respectively.

Both the encoder and the decoder use a Transformer structure with $d_{\text{model}} = d_{\text{hidden}} = 128$, $n_{\text{head}} = 4$, $n_{\text{layer}} = 2$. Following existing works (Lample et al., 2019), the target style is treated as a special start token in decoder. Both the style classifier for automatic evaluation and the pretrained $p_\psi$ in the style compatibility loss follow the TextCNN (Kim, 2014) architecture but are independently trained. To backpropagate the gradients from $p_\psi$ to $\theta$, we approximate $y$ in Eq. 4 with the softmax distribution sequence from which $y$ should be sampled.

### C.2 Hyper-parameters

We tune the hyper-parameters on the development set. As a result, the balancing weight $\alpha$ is set to $0.1$, the number of iterations $K$ in iterative decoding is set to $4$, and the margin $\eta$ in contrastive learning is set to $1$.

We implement all models using PyTorch and conduct the experiments on a single Nvidia's GTX 1080Ti GPU. Each model is trained for 100,000 iterations with a batch size of 64 on Yelp and 32 on GYAFC. The Adam algorithm (Kingma and Ba, 2015) with a learning rate of 0.001 is used for optimization.

### C.3 Technical Details

**Target Length.** During inference, the target length needs to be provided in advance. Models in NMT usually train a target length predictor during training with the available ground-truth outputs. However, this strategy cannot be adapted to our unsupervised task. Fortunately, on both the sentiment transfer task and the formality transfer task, the desired transferred result only involves local changes towards the source text thus has a similar length with the source length. Therefore, motivated by Wang et al. (2019), we generate a transferred result for each $T \in [T_x - B, T_x + B]$, where $T_x$ denotes the length of the source input. As a result, we obtain $2B + 1$ candidates and select the candidate with the highest log-probability (assigned by the decoder) as the final result. In our experiments, we set $B$ to 2.

**Knowledge Distillation.** For models with knowledge distillation, i.e., BaseNAR+KD, BaseNAR+KD+CL, BaseNAR+KD+ID, BaseNAR+KD+CL+ID, we eliminate the self-reconstruction loss and the cycle-reconstruction loss from the full loss, as preliminary experiments demonstrate there is no performance degradation with this elimination. The reason should be that, knowledge distillation can provide more reliable and direct gradients to the model, leaving the weak supervision from self-reconstruction and cycle-reconstruction as redundant information.

**Contrastive Learning.** For models with contrastive learning, i.e., BaseNAR+CL, BaseNAR+KD+CL, BaseNAR+CL+ID, BaseNAR+KD+CL+ID, the contrastive learning based regularization will only be involved for the last 30% training iterations. Consistent with previous contrastive learning works, earlier involvement of the regularization may lead to unstable training and cannot bring performance improvement. As discussed in Section 3.3, the contrastive learning based regularization can be paired with self-reconstruction, cycle-reconstruction, and knowledge distillation. However, based on our preliminary experiments, (1) when knowledge distillation is used, we only pair this regularization with knowledge distillation, and (2) when knowledge distillation is not used (thus we cannot use $\mathcal{R}_{\text{kd}}$), we only pair this regularization with cycle-reconstruction, as more sophisticated setting cannot bring further improvement.

**Iterative Decoding.** For models with iterative decoding, i.e., BaseNAR+ID, BaseNAR+KD+ID, BaseNAR+CL+ID, BaseNAR+KD+CL+ID, all losses except the style compatibility loss will be reformulated to fit the CMLM scheme. During training, we randomly mask $n$ ($0 \leq n \leq T$) tokens for a target sequence with length $T$ and then optimize the model by predicting these masked tokens. One problem here is that, for the style compatibility loss, we need to generate an output $y$, however, there is not partial target sequence to utilize. We have considered two strategies: one strategy is to assume all tokens are masked; and the other strategy is that we first go through an inference stage to get an output $\hat{y}$, then randomly mask and repredict $n$ tokens in $\hat{y}$, and the repredicted tokens and the unmasked tokens are mixed as the input $y$ for the style classifier. Our preliminary experiments show that the second strategy can always achieve much better results, so we stick to this strategy when iterative decoding is used.

## D Addtional Qualitative Results

Table 5 and Table 6 present addtional qualitative results on Yelp and GYAFC, respectively.

| | **positive → negative** |
|---|---|
| Input | the prices were the best and worth it . |
| BaseAR | the prices were the only good and worth it . |
| BaseNAR | the prices were the worst worth it . |
| BaseNAR+KD | the prices were the only and not worth it . |
| BaseNAR+CL | the prices were the worst worth it . |
| BaseNAR+ID | the prices were not worth it . |
| BaseNAR+KD+CL | the prices were the worst and not worth it . |
| BaseNAR+KD+ID | the prices were the worst and not worth it . |
| BaseNAR+CL+ID | the prices were the worst and not worth it . |
| BaseNAR+KD+CL+ID | the prices were the worst and not worth it . |
| Input | this place has been making great sushi and sashimi for years . |
| BaseAR | this place has been making @num mins for years . |
| BaseNAR | this place has been making bad and sashimi for years . |
| BaseNAR+KD | this place has been making bad sushi and sashimi for @num years . |
| BaseNAR+CL | this place has been making bad sushi and sashimi for years . |
| BaseNAR+ID | this place has been making horrible sushi for years . |
| BaseNAR+KD+CL | this place has been making bad sushi and sashimi for years . |
| BaseNAR+KD+ID | this place has been making bad sushi and sashimi for years . |
| BaseNAR+CL+ID | this place has been making horrible sushi and sashimi for years . |
| BaseNAR+KD+CL+ID | this place has been making bad sushi and sashimi for years . |
| | **negative → positive** |
| Input | this branch is getting worse and worse . |
| BaseAR | this branch is getting better and better . |
| BaseNAR | this branch is getting and incredible . |
| BaseNAR+KD | this branch is getting better and better . |
| BaseNAR+CL | this branch is getting and better . |
| BaseNAR+ID | this branch is getting exceptional . |
| BaseNAR+KD+CL | this branch is getting better and better . |
| BaseNAR+KD+ID | this branch is getting better and better . |
| BaseNAR+CL+ID | this branch is getting better . |
| BaseNAR+KD+CL+ID | this branch is getting better and better . |
| Input | this is the worst panda express location there is ! |
| BaseAR | this is the best panda express location there is ! |
| BaseNAR | this is the best panda location there is ! |
| BaseNAR+KD | this is the best panda express location there is ! |
| BaseNAR+CL | this is the best panda express location there is ! |
| BaseNAR+ID | this is the best panda express location is there ! |
| BaseNAR+KD+CL | this is the best panda express location there is ! |
| BaseNAR+KD+ID | this is the best panda express location there is ! |
| BaseNAR+CL+ID | this is the best panda express location there is ! |
| BaseNAR+KD+CL+ID | this is the best panda express location there is ! |

Table 5: Ouputs of different models on exemplary sentences from YELP.

| | **informal → formal** |
|---|---|
| Input | yes i m not one of those people but i know there are lots of them |
| BaseAR | yes , i am not one of those people but i know there are lots of them . |
| BaseNAR | yes i am not one of those people but i know there are lots of . |
| BaseNAR+KD | yes , i am not one of those people but i know there are lots of them . |
| BaseNAR+CL | yes i am not one of those people but i know there are lots of " |
| BaseNAR+ID | yes i am not one of those people but i know there are lots . |
| BaseNAR+KD+CL | yes , i am not one of those people but i know there are lots of them . |
| BaseNAR+KD+ID | yes , i am not one of those people but i know there are lots of them . |
| BaseNAR+CL+ID | yes i am not one of those people but i know there are lots of them . |
| BaseNAR+KD+CL+ID | yes , i am not one of those people but i know there are lots of them . |
| Input | no u should nt leave them ... just teach them what to do to please u better ... |
| BaseAR | no , you should not leave them . just teach them what to do . please you better . |
| BaseNAR | no you should not leave them ... just teach what to do to please you better . |
| BaseNAR+KD | no , you should not them . just teach them what to do to please you better . |
| BaseNAR+CL | no you should not leave them " just teach what to do to please you better " |
| BaseNAR+ID | no " should not leave them " just teach them what to do to please " better ... " |
| BaseNAR+KD+CL | no , you should not leave them . just teach them what to do to please you better . |
| BaseNAR+KD+ID | no , you should not leave them . just teach them what to do to please you better . |
| BaseNAR+CL+ID | no " should not leave them " just teach them what to do " please " |
| BaseNAR+KD+CL+ID | no , you should not leave them . just teach them what to do to please you better . |
| | **formal → informal** |
| Input | you should find someone who does not hate you . |
| BaseAR | u should find someone who does nt hate you . |
| BaseNAR | you should find someone who does nt hate you . |
| BaseNAR+KD | u should find someone who does nt hate u |
| BaseNAR+CL | u should find someone who does nt hate you . |
| BaseNAR+ID | you should find someone who does nt hate you . |
| BaseNAR+KD+CL | u should find someone who does nt hate u . |
| BaseNAR+KD+ID | u should find someone who does nt hate u |
| BaseNAR+CL+ID | you should find someone who does nt hate you . |
| BaseNAR+KD+CL+ID | u should find someone who does nt hate u . |
| Input | do not allow her to dominate your life for she will simply have to learn to deal with it . |
| BaseAR | do nt allow her to dominate your life for your life she will jus have to learn to deal with it . |
| BaseNAR | do nt allow her to ur life for she will dont have to learn deal with it . |
| BaseNAR+KD | do nt allow her to dominate your life for she will jus have to learn to deal with it |
| BaseNAR+CL | do nt allow her to dominate ur life for she dont have to learn to deal with it . |
| BaseNAR+ID | do nt allow her to dominate your life for she will definately have to learn to deal with it :p |
| BaseNAR+KD+CL | do nt allow her to dominate your life for she will jus have to learn to deal with it |
| BaseNAR+KD+ID | do nt allow her to dominate your life for she will jus have to learn to deal with it . |
| BaseNAR+CL+ID | do nt allow her to dominate your life for she will definately have to learn to deal with it :p |
| BaseNAR+KD+CL+ID | do nt allow her to dominate ur life for she will jus have to learn to deal with it |

Table 6: Ouputs of different models on exemplary sentences from GYAFC.