Fluency Matters! Controllable Style Transfer with Syntax Guidance

Ji-Eun Han[1] **and Kyung-Ah Sohn**[1,2,*]
[1]Department of Artificial Intelligence, Ajou University
[2]Department of Software and Computer Engineering, Ajou University
{hanji0514, kasohn}@ajou.ac.kr
* Corresponding author

Abstract

Unsupervised text style transfer is a challenging task that aims to alter the stylistic attributes of a given text without affecting its original content. One of the methods to achieve this is controllable style transfer, which allows for the control of the degree of style transfer. However, an issue encountered with controllable style transfer is the instability of transferred text fluency when the degree of the style transfer changes. To address this problem, we propose a novel approach that incorporates additional syntax parsing information during style transfer. By leveraging the syntactic information, our model is guided to generate natural sentences that effectively reflect the desired style while maintaining fluency. Experimental results show that our method achieves robust performance and improved fluency compared to previous controllable style transfer methods.

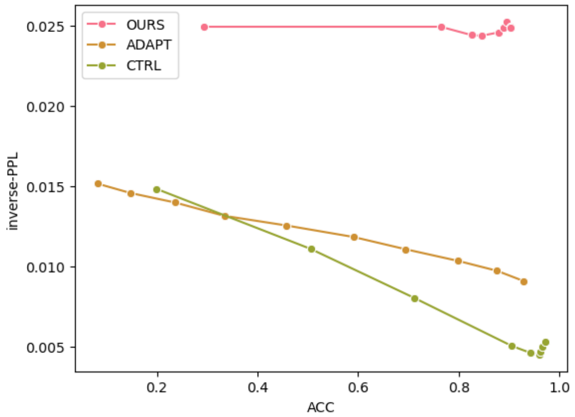


Figure 1: Comparison of inverse-perplexity between controllable style transfer models as the style transfer degree changes. A higher inverse-perplexity score indicates better fluency of generated text. Each dot represents the inverse-perplexity corresponding to style control degrees from 1 to 10. The x-axis represents the transfer accuracy.

1 Introduction

Text style transfer has been garnering increasing interest in the field of natural language generation. Its applicability spans a wide range of tasks, including data augmentation (Chen et al., 2022), stylistic writing for marketing purposes (Kaptein et al., 2015; Jin et al., 2020), and natural chatbot response generation (Kim et al., 2019).

Text style transfer aims to modify a given text to represent a target style attribute. Key considerations for this task include ensuring that the generated text: (i) reflects the desired style attribute, (ii) preserves style-irrelevant content, and (iii) generates a sentence that seems natural to humans. Target style attributes can include various styles such as sentiment, formality, politeness, offensiveness, and genre. In this work, we primarily focus on sentiment as the target style attribute.

Some approaches, such as those by Jhamtani et al. (2017), Carlson et al. (2018), and Wang et al. (2019b), train their models using parallel datasets consisting of pairs of source text and transferred text. However, collecting human-generated transferred text can be both time-consuming and costly. As a result, mainstream research has primarily focused on unsupervised methods that rely solely on source text.

Unsupervised methods for text style transfer can be broadly categorized into two approaches: disentanglement and entanglement. Hu et al. (2018), Shen et al. (2017), and John et al. (2018) proposed models that disentangle content and style in the latent space. However, content and style cannot be entirely separated. As a result, rather than separating content and style, an alternative approach was proposed that uses entangled latent representation.

In the entanglement approach, style information is used to overwrite the latent representation of the source text, resulting in the text reflecting the target style. Multiple approaches have been proposed to achieve this, including the use of back-translation loss (Sennrich et al., 2016a) or a combination of reconstruction, cycle loss, and style classification

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 162–171
July 14, 2023 ©2023 Association for Computational Linguistics

loss in the method proposed by Dai et al. (2019).

Expanding upon the entanglement approach, Wang et al. (2019a) and Kim and Sohn (2020) attempted to control the degree of style during the transfer process. The advantage of these models lies in their ability to generate diversely transferred sentences with varying degrees of style. However, although these models transfer sentences effectively, the generated sentences often lack naturalness. To address this issue, we have endeavored to improve the fluency of our model by incorporating additional syntax information.

Figure 1 highlights that our method substantially outperforms the models proposed by Kim and Sohn (2020) (ADAPT) and Wang et al. (2019a) (CTRL) in terms of perplexity, a metric used to measure the fluency of generated sentences. It should be noted that in the figure, we have inverted the perplexity score, meaning that a higher score indicates better fluency. The comparison reveals that as both the style degree and the accuracy of the transferred style increase, the inverted perplexity score declines in both models. However, our model maintains stable perplexity scores. This suggests that the incorporation of syntax parses helps to preserve the syntactic structure of transferred sentences across diverse levels of accuracy. We found that the other models tend to prioritize generating more tokens containing the target style to enhance accuracy, regardless of fluency. As a result, the generated sentences become less fluent as the style degree increases.

To enhance the fluency of controllable text style transfer, we extract syntax parses from constituency parse trees and encode them into syntactic embeddings. After encoding, we concatenate these embeddings with semantic and style embeddings.

Our experimental results on two datasets demonstrate that our method outperforms several text style transfer baselines. Specifically, our model shows remarkable performance in relation to perplexity. Furthermore, we present an ablation study and qualitative analysis. We also evaluate the syntax preservation capability among controllable models to validate the effectiveness of incorporating syntax parses. Our contributions are suggested as follows:

- We propose a novel approach to enhance the fluency of the controllable text style transfer task. We place emphasis on the fluency of the generated text, ensuring that it sounds natural as if written by a human. By incorporating additional syntax information as a model input, we effectively improve the model's fluency regardless of the transfer strength.
- We validate the effectiveness of our approach by conducting experiments utilizing automatic evaluation metrics. Moreover, we analyze our method with respect to syntax preservation and fluency. The results show that our method helps the model comprehend the syntactic structure of the input sentences and serves as a constraint, steering the model towards generating more natural text.
- We present text-level outputs and compare them to outputs from controllable text style transfer baselines, demonstrating that our model generates fluent sentences while preserving both the syntactic structure and content integrity of the input text.

2 Related Work

Entangle-based text style transfer
One of the approaches employed in unsupervised text style transfer is entanglement. Rather than dividing the latent representation of an input text into content and style components, the entanglement approach directly integrates the input text's latent representation with target style information. Subramanian et al. (2019) use back-translation loss (Sennrich et al., 2016a) to enable learning in two steps: first, the model transfers the input sentence x reflecting the target style s', and second, it reconstructs the output from the previous step with the original style s. Dai et al. (2019) train their model with both reconstruction and cycle loss. Additionally, a style classifier is used to incorporate a style classification loss during training.

Controllable style transfer
Controllable style transfer involves adjusting the magnitude of style transfer strength in the transferred text. Wang et al. (2019a) proposed the Fast-Gradient-Iterative-Modification algorithm to modify the latent representation of the input text to follow the target style. A modification weight is used to control the transfer strength.

Similarly, Kim and Sohn (2020) use the modification weight and train style embeddings to control the style transfer strength. Two style embeddings – positive and negative – are trained in training time. By multiplying these embeddings by the

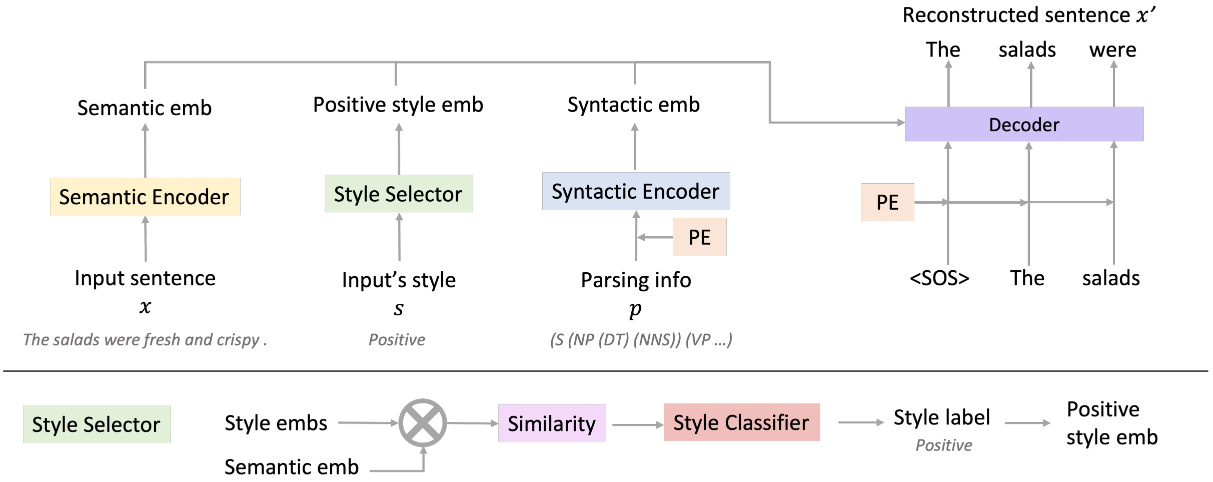


Figure 2: The architecture of our proposed model, consisting of four components: semantic encoder, syntactic encoder, style selector, and decoder. The *upper* figure shows the overall model architecture. The input sentence x, its style label s, and syntax parse p are provided to the model. The semantic encoder, style selector, and syntactic encoder each output an embedding. A concatenated latent representation of the three embeddings—semantic, syntactic, and style—is then fed to the decoder, which generates the reconstructed sentence x'. The *bottom* figure shows Style Selector which selects the style embedding of input x.

modification weight, the model can generate style-controlled text.

Our model adopts the approach by Kim and Sohn (2020) but with the additional step of incorporating style embeddings alongside semantic and syntactic embeddings. The integration of all three types of embeddings, along with the additional syntax information, enables our model to generate more sophisticatedly controlled and natural text.

Syntax-guided generation

Syntax-guided generation generally uses additional syntax information, particularly in machine translation and paraphrasing. In both tasks, syntax information is typically derived from constituency parse trees. After the parse tree has been extracted, it is linearized and then provided to the model along with the input text.

In machine translation, Yang et al. (2020) predict soft target templates and use them to provide syntactical guidance during the translation procedure. Sun et al. (2021) and Huang and Chang (2021) utilize syntax templates to generate syntactically controlled paraphrases that conform to these templates. Sun et al. (2021) use a ranker and retriever to select target parse templates and then generate texts according to the templates. Huang and Chang (2021) train a parse generator to generate diverse syntax templates.

Previous research has explored the importance of syntax in text style transfer. Hu et al. (2021) demonstrated that previous style classifiers were incapable of learning syntax and could worsen models' performance, especially in formality transfer. They employed Graph Convolutional Networks (GCNs) to extract syntactic information and used it to train both syntax-classifier and syntax-encoder. Rather than relying on GCNs for incorporating syntactic information, our approach extracts syntax information from the constituency parse trees. Subsequently, we combine the encoded linearized parse information with semantic and style embeddings.

3 Proposed Method

We formulate the syntax-guided text style transfer as follows: given an input text x, its corresponding style label s, and syntax parse p as model inputs, we train our model using an autoencoder to reconstruct x while preserving the style s. Training the model based on reconstruction is necessary in an unsupervised setting due to the lack of a parallel dataset. The actual style transfer takes place during inference time.

3.1 Model Architecture

Figure 2 shows our overall model architecture. Our model consists of four key components: i) a *semantic encoder* that encodes the input text x; ii) a *syntactic encoder* that encodes the input text's syntax parse p; iii) a *style selector* that chooses the appropriate style embedding se for the input text

x; iv) a *decoder* that generates either reconstructed sentences or transferred sentences.

Semantic encoder. The semantic encoder converts the input text x into a semantic embedding z_{sem}. We represent each token in the input text as $x_1, x_2, ..., x_n$, where n is the number of tokens in x. The semantic encoding process is expressed as follows:

$$z_{sem} = (z_1^{sem}, z_2^{sem}, ..., z_n^{sem}) = Enc_{sem}((x_1, x_2, ..., x_n))$$

where Enc_{sem} represents the semantic encoder. We do not use positional encoding from Transformer (Vaswani et al., 2017) for the semantic embedding, but we apply it to the syntactic encoder. This leads to a semantic embedding that is less affected by word order and thus mainly captures the meaning of the text. In other words, the semantic embedding without positional encoding functions similarly to a bag of words representation. Previous studies have shown that bag of words representation can be effective in various tasks. For example, Xu et al. (2010) demonstrated that generating abstract summaries using only keywords in a bag of words is feasible. In addition, Tao et al. (2021) showed that neural models can successfully reconstruct sentences from an unordered bag of words.

Syntactic encoder. The goal of the syntactic encoder is to produce a syntactic embedding z_{syn} by taking the linearized syntax parse $p = \{p_1, p_2, ..., p_k\}$ as input. This can be expressed as follows:

$$z_{syn} = (z_1^{syn}, z_2^{syn}, ..., z_k^{syn}) = Enc_{syn}((p_1, p_2, ..., p_k))$$

To ensure that the syntax parse includes the information about the order of words, we utilize a Transformer encoder with positional encoding.

Style selector. We define two types of style embeddings: positive and negative. The style selector predicts the style of the input text x and then selects the appropriate style embedding. The process involves three phases, represented at the bottom of Figure 2. In the first phase, we calculate the similarities between each style embedding and the semantic embedding of x. To accomplish this, we use the dot product. In the second phase, we predict the style label of x by utilizing a style classifier C_θ. Finally, in the third phase, we select the final style embedding of x. This is achieved by leveraging the predicted style label in the second phase to select the proper style embedding of x.

Decoder. To generate the reconstructed text x', we concatenate the semantic, syntactic, and style embeddings of the input text x, and feed the resulting concatenated embedding to the Transformer decoder. The decoder then generates x' autoregressively. This process can be represented as follows:

$$\begin{aligned} x' &= (x'_1, x'_2, ..., x'_m) \\ &= Dec(concat(z_{sem}, z_{syn}, z_{style})) \end{aligned}$$

3.2 Training

Since we do not have access to a parallel dataset for this task, we train our model in an unsupervised manner by combining the reconstruction loss from a Transformer-based autoencoder with a style classification loss from a style classifier.

Reconstruction loss. We employ a Transformer-based autoencoder. We calculate the reconstruction loss by comparing the reconstructed sentence to the original sentence. The reconstruction loss is represented as follows:

$$L_{res} = \sum_{i=1}^{n} logP(x'_i = x_i | \bar{x}, p_x, s_x, x'_1, ..., x'_{i-1})$$

where $\bar{x}$ represents an unordered list containing all tokens in the input text x, while p_x represents the syntax parse, and s_x represents the style label of x. Additionally, x'_i represents the generated i-th token, with $x'_1, ..., x'_{i-1}$ being the previously autoregressively generated tokens. By considering the relationships between the semantic, syntactic, and style embeddings, our model gains the ability to reconstruct the input text x.

Style classification loss. In the second phase of the style selector, we utilize a style classifier denoted as C_θ to predict the sentiment of the input text. The classifier is comprised of simple linear layers. Style embeddings SE_i contain two embeddings: a positive embedding and a negative embedding in this task. The similarity between the semantic embedding z_{sem} and each style embedding in SE_i is given as an input of the classifier. Since the gold label is already provided in the training data, we calculate the loss by comparing the predicted label to the gold label y. This procedure is based on the following loss function:

$$L_{style}(C_\theta(Sim(z_{sem}, se_i)), y) = -\sum_{i=1}^{k} \bar{q}_i log(q_i)$$

where C_θ denotes the style classifier, Sim is the similarity calculation performed via dot product and se_i refers to one of the style embeddings in SE_i. $\bar{q}_i$ represents the true style label probability distribution, while q_i represents the predicted style label probability distribution. By optimizing this L_{style} loss function, we train the style embeddings.

Joint training loss. Reconstructing the input text is influenced by the style embedding since the style embedding is concatenated with the semantic and syntactic embeddings. Therefore, we train the autoencoder and the style classifier together using the joint loss as follows:

$$L = L_{res} + L_{style}$$

This approach allows the model to learn to reconstruct the input text while also considering the style information.

3.3 Inference

During inference, the semantic embedding is adjusted to perform style transfer. We use the style embedding that was learned during training. The style transfer operation is represented as follows:

$$z'_{sem} = z_{sem} + w \cdot se'_i$$

where z_{sem} is the semantic embedding, w is a style transfer weight, and se'_i represents the style embedding of the target style. The hyperparameter w controls the degree of style transfer. Following the adjustment of z'_{sem}, it is concatenated with the syntactic and style embeddings before being input into the decoder.

4 Experiment

4.1 Dataset

We evaluate our model with Yelp and Amazon datasets, which are commonly used in unsupervised text style transfer. Table 1 presents the number of data samples for the train, validation, test split in Yelp and Amazon datasets. Each dataset contains human transferred references.

Dataset	Train	Valid	Test
Yelp	443,259	1,000 / style	500 / style
Amazon	554,997	1,000 / style	500 / style

Table 1: Details of Yelp and Amazon datasets.

Yelp. The dataset consists of restaurant reviews on Yelp. The reviews include scores that range from 1 to 5. Each sentence is labeled with the sentiment, either positive or negative according to the score. Sentences with scores of 1 and 2 are labeled as negative, and 4 and 5 are labeled as positive. We use the preprocessed version of the dataset from Li et al. (2018).

Amazon. The dataset contains product reviews from Amazon. The same labeling scheme as the Yelp dataset is used. We use the dataset from He and McAuley (2016).

4.2 Evaluation Metric

We evaluate the performance of our model by comparing it to previous works using three commonly used metrics.

Accuracy measures how well the transferred sentences conform to the target style. To calculate the accuracy, we use a fasttext classifier (Joulin et al., 2016) that is trained on each training dataset. A higher accuracy indicates better model performance.

Content preservation metric evaluates the model's ability to maintain the meaning of the input text, regardless of its stylistic attributes. We measure this using the BLEU score (Papineni et al., 2002), which quantifies how much the transferred sentences overlap with human-written sentences. A higher BLEU score indicates greater similarity between the two sentences. To compute the BLEU-2 score, we utilize the nlg-eval[1](Sharma et al., 2017) package.

Fluency shows how natural the transferred text is. We use perplexity (PPL) as a measure of fluency. In our work, GPT-2 language model (Radford et al., 2019) is used. The GPT-2 model is fine-tuned with the training data of each dataset, and it calculates the 3-gram PPL score.

4.3 Baseline Models

To evaluate the effectiveness of our model, we compare it with several unsupervised text style transfer models. These models can be categorized into two groups based on their ability to control the degree of style transfer.

Uncontrollable models

1) Cross-Align (Shen et al., 2017): this model disentangles style and the content of the input text using a variational autoencoder. It uses an alignment approach to match the input and the transferred text. **2) StyleEmb** (Fu et al., 2017): this model also disentangles the latent into the style and

[1] https://github.com/Maluuba/nlg-eval

Model		Yelp			Amazon		
		ACC↑	**BLEU↑**	**PPL↓**	**ACC↑**	**BLEU↑**	**PPL↓**
Human reference		73.4	100.0	42.3	42.7	100.0	71.3
(1)	Cross-Align	74.5	21.5	66.9	82.9	8.6	27.5
	StyleEmb	8.8	33.9	61.6	44.5	24.6	114.3
	DeleteAndRetrieve	79.0	16.0	69.4	50.2	42.4	83.3
	Style transformer	84.9	**42.3**	164.0	62.0	42.3	104.6
	RACoLN	87.4	42.2	55.8	**90.1**	**52.1**	100.2
	PromptAndRerank 0-shot	52.2	21.4	65.4	43.8	32.5	91.7
	PromptAndRerank 4-shot	61.2	30.2	57.7	50.0	30.4	68.5
(2)	Controllable-transfer	71.1	35.5	124.2	55.0	36.0	109.6
	Adaptive-StyleEmb	**87.6**	33.9	101.3	74.1	34.19	90.6
	Ours	82.5	18.8	**40.9**	76.8	22.44	**26.8**

Table 2: Evaluation results conducted on the Yelp and Amazon datasets. We divided the models into two groups: (1) uncontrollable models, (2) controllable models. We selected the style transfer weight for models in (2) based on the geometric mean of the accuracy and BLEU score.

the content part using an adversarial network. It uses style embeddings that control the generated styles. **3) DeleteAndRetrieve** (Li et al., 2018): this model first removes the stylistic attributes in the input text and transfers the input by replacing those attributes with retrieved target attribute markers. The model is based on recurrent neural networks. **4) Style transformer** (Dai et al., 2019): unlike other models mentioned above, it overwrites the latent representations with target stylistic attributes. The model architecture is based on Transformer. **5) RACoLN** (Lee et al., 2021): this model is implemented using a gated recurrent unit architecture, and it utilizes a reverse attention mechanism to preserve the content of the input text during style transfer. **6) PromptAndRerank** (Suzgun et al., 2022): pre-trained language models are utilized to generate transferred text. We use the zero-shot and few-shot results from EleutherAI's GPT-J-6B using curly brackets as delimiters.

Controllable models
1) Controllable-transfer (Wang et al., 2019a): it modifies the latent representation of the input text iteratively until the desired degree of style transfer is achieved. **2) Adaptive-StyleEmb** (Kim and Sohn, 2020): it controls the style of the input text by adding style embeddings learned during training to the input latent representation. For these two models, we used pretrained models provided by the authors to get the model outputs.

4.4 Implementation Details

We apply byte pair encoding (Sennrich et al., 2016b) for tokenization and utilize the Stanford CoreNLP parser (Manning et al., 2014) to obtain constituency parses. The maximum token length of the input sentences is 40 and the max token length of linearized syntax parses is 180. Word embeddings are initialized using GloVe (Pennington et al., 2014). The encoder and decoder architecture of our model is implemented with standard Transformer architecture (Vaswani et al., 2017) with its default parameters. We employ the Adam optimizer with a learning rate of 1e-4 and a weight decay of 1e-5. The word dropout probability is set to 0.4. The training process is carried out for 10 epochs.

5 Results

5.1 Quantitative Evaluation

Table 2 presents the results of our quantitative evaluation on the Yelp and Amazon datasets. For controllable text style transfer models, there are multiple output candidates that can be generated by varying the style transfer weight. To select the best candidate, we choose the output with the highest geometric mean of the accuracy and BLEU score.

Our model demonstrates competitive accuracy on the Yelp dataset compared to both controllable and uncontrollable text style transfer models. Notably, our model achieves the lowest PPL score among all the compared models, with a score of 40.9, which is close to the PPL of human reference 42.3. On the Amazon dataset, our model achieves the highest accuracy among controllable models and also shows the lowest PPL score. Overall, these results suggest that our proposed method effectively improves the fluency of transferred text while maintaining high accuracy, although there is

	Negative→Positive
Input	other than that , food here is pretty **gross** .
Controllable-transfer	other than that , food is here pretty **fun makes you delicious** .
Adaptive-StyleEmb	other than that , food here is pretty **good and enjoy warm** .
Ours	other than that , food here is pretty **good** .
	Positive→Negative
Input	the service is **friendly and attentive**.
Controllable-transfer	the service **was not less but then disappointed had the wait fries**.
Adaptive-StyleEmb	the service is **then rude and had old fill that is your worse**.
Ours	the service is **slow and rude**.

Table 3: Comparison of transferred outputs at the text-level in controllable models. Bolded text indicates differences from the input text.

room for improvement in terms of the BLEU score.

5.2 Qualitative Evaluation

In Table 3, we compare transferred outputs from the controllable style transfer models. To select the optimal style transfer weight for the controllable text style transfer models, which is a hyperparameter, we use the same criterion used in 5.1, selecting the weight that shows the highest geometric mean of the accuracy and BLEU score. In the first sample, where a negative sentence is transferred into a positive one, our model is able to convert the token *gross* to *good* while preserving the content of the sentence. In contrast, the other compared models, suggested by Wang et al. (2019a) and Kim and Sohn (2020), generate some tokens that are not present in the original input sentence, such as *fun makes you delicious* and *good and enjoy warm*.

In the second sample, where a positive sentence is transferred into a negative one, our model is able to effectively transfer the sentiment of the input sentence by converting *friendly and attentive* into *slow and rude* while maintaining the naturalness and fluency of the sentence. The other compared models are also able to transfer the input into a negative sentiment. However, their outputs are less natural and fluent compared to ours.

These results indicate that our proposed method is highly effective in transferring the sentiment of the input text to the target style, while ensuring that the content and fluency of the transferred text are maintained.

5.3 Ablation Study

To further demonstrate the importance of incorporating syntax parse information, we conduct an ablation study. Table 4 shows the impact of concatenating syntactic and style embedding on the three evaluation metrics of the transferred text. Our proposed model (Model 1) outperformed the other models across all three metrics. When we excluded the syntactic embedding (Model 2), the resulting transferred text was less fluent, as evidenced by an increase of approximately 5 points in PPL. Similarly, when we removed both syntactic and style embeddings, the performance of the model dropped significantly, particularly in terms of accuracy and PPL. Conversely, adding the syntactic embedding to Model 4 (Model 3) resulted in a substantial decrease in PPL. These results underscore the crucial role of syntax parsing information in generating fluent and natural transferred text.

	Syn emb	Style emb	ACC	BLEU	PPL
(1)	O	O	**82.5**	**18.8**	**40.9**
(2)	X	O	57.3	13.7	45.8
(3)	O	X	21.0	14.6	41.3
(4)	X	X	16.0	16.1	58.6

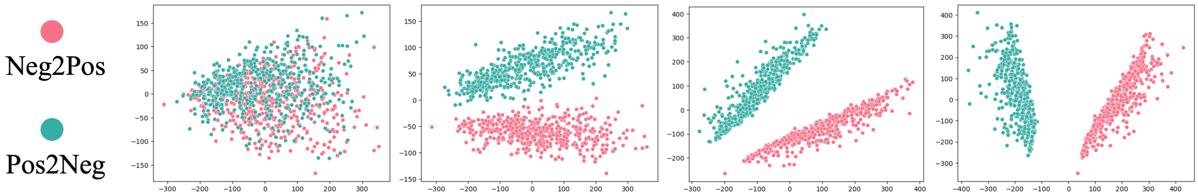
Table 4: Ablation study of the impact of concatenating syntax and style embeddings. We set the style transfer weight w of each model with the geometric mean of accuracy and BLEU score.

Model	Original	Human-transferred
Ours	**92.0**	**67.4**
Controllable-transfer	71.4	56.6
Adaptive-StyleEmb	71.4	56.9

Table 5: Syntax similarity of each model using a metric based on weighted ROUGE scores. It compares the linearized syntax parses of generated sentences to those of reference sentences. Our approach was compared to two controllable text style transfer baselines on the Yelp dataset to demonstrate its ability to preserve syntax while transferring style.

5.4 Syntax Preservation

As demonstrated in 5.2, the controllable style transfer models tend to generate more tokens than the input text in order to incorporate more stylistic



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 3$

Figure 3: Semantic embedding visualization differing style weight w.

attributes. While this approach can contribute to higher accuracy, it may compromise the fluency of the output. Therefore, we conducted an experiment to assess the syntax preservation capabilities.

Syntax preservation is determined by the similarity between the syntax parse of the source text and that of the reference text. We employ the syntax parse similarity measure using weighted ROUGE scores (Lin, 2004) proposed in Sun et al. (2021).

$$S(p_{src}, p_{ref}) = a * ROUGE1 + b * ROUGE2 \\ + c * ROUGEL$$

We set $a = 0.2$, $b = 0.3$, $c = 0.5$, following previous work. We applied the style transfer to the test set and compared the transferred output with two types of references: the original test set and human-transferred references. Table 5 demonstrates that the output generated by our model is considerably more similar to both references. This finding suggests that concatenating syntax parses aids the model in retaining the syntactic structures even though the sentiment has transferred.

5.5 Syntax-guided Reconstruction Ability

We evaluate the impact of syntax information on the reconstruction ability of our model, which is trained using reconstruction loss. To assess the pure reconstruction ability, we exclude style information from all models.

Model	ACC	Self-BLEU	PPL
Ours	**3.0**	**90.9**	**36.9**
Controllable-transfer	4.1	78.7	53.4
Adaptive-StyleEmb	5.4	71.5	63.2

Table 6: Impact of syntax parses on model's reconstruction ability evaluated on Yelp dataset.

The results presented in Table 6 highlight the impact of additional syntax information on the reconstruction ability of our model. To evaluate this ability, we use the self-BLEU metric which measures the similarity between the original input text and the reconstructed text, where a higher score indicates better reconstruction ability. Conversely, for accuracy, a lower score indicates better reconstruction ability since it is the accuracy for style transfer. The PPL is calculated using GPT-2 language model. Our findings indicate that incorporating additional syntax parses not only enhances transfer capability but also improves reconstruction ability.

5.6 Embedding visualization

We visualize semantic embeddings in Figure 3 using PCA (Wold et al., 1987) after they are transferred using learned style embeddings. Red dots represent positive sentences that were originally negative, while green dots indicate negative sentences that were originally positive. At w=0, the two colors of dots are entangled. However, as the transfer weight increases, these embeddings gradually separate. At w=3, it is evident that the embeddings are completely transferred and distinctly separated. This implies that the style transfer weight effectively controls the degree of transfer.

6 Conclusion

In this paper, we proposed a controllable, syntax-guided text style transfer model. We improved the fluency of transferred sentences, irrespective of the style transfer strength, by incorporating syntax parses and concatenating their embeddings with semantic and style embeddings. Our approach outperformed previous controllable models on two datasets in terms of consistent PPL scores and natural sentence generation while preserving context. However, our model yielded lower BLEU scores compared to other controllable style transfer models. Future work aims to improve content preservation capabilities while maintaining performance across varying style transfer weights.

Limitation

Our proposed method demonstrates stable perplexity even as the style transfer weight changes, but it yields a lower BLEU score compared to other controllable style transfer models. We hypothesize that the lower BLEU score may be attributed to the fact that the BLEU score calculation is based on just one human-written transferred sentence option per source sentence. This lower score could be a result of our model generating diverse sentences that do not necessarily overlap with the provided human-written references.

Ethics Statement

There are several ethical considerations that must be taken into account when developing a text style transfer model. One important consideration is the risk of the generated text being used to spread hate speech or misinformation. It is also crucial to ensure that the model does not exhibit bias towards a particular demographic, which could result in harmful outcomes. Another potential ethical concern is the misuse of the model for malicious purposes, such as generating negative comments or fake news. These issues need to be addressed to ensure that the development and use of the model align with ethical principles and values.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2022R1A2C1007434), and also by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-2018-0-01431) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

References

Keith Carlson, Allen Riddell, and Daniel Rockmore. 2018. Evaluating prose style transfer with the bible. *Royal Society Open Science*, 5(10):171920.

Shuguang Chen, Leonardo Neves, and Thamar Solorio. 2022. Style transfer as data augmentation: A case study on named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1827–1841, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2017. Style transfer in text: Exploration and evaluation.

Ruining He and Julian McAuley. 2016. Ups and downs. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee.

Zhiqiang Hu, Roy Ka-Wei Lee, and Charu C. Aggarwal. 2021. Syntax matters! syntax-controlled in text style transfer.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2018. Toward controlled generation of text.

Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence-to-sequence models.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, Lisa Orii, and Peter Szolovits. 2020. Hooks in the headline: Learning to generate headlines with controlled styles.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2018. Disentangled representation learning for non-parallel text style transfer.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification.

Maurits Kaptein, Panos Markopoulos, Boris de Ruyter, and Emile Aarts. 2015. Personalizing persuasive technologies: Explicit and implicit personalization using persuasion profiles. *International Journal of Human-Computer Studies*, 77:38–51.

Heejin Kim and Kyung-Ah Sohn. 2020. How positive are you: Text style transfer using adaptive style embedding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2115–2125, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Soomin Kim, Joonhwan Lee, and Gahgene Gweon. 2019. Comparing data from chatbot and web surveys: Effects of platform and conversational style on survey response quality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA. Association for Computing Machinery.

Dongkyu Lee, Zhiliang Tian, Lanqing Xue, and Nevin L. Zhang. 2021. Enhancing content preservation in text style transfer using reverse attention and conditional layer normalization.

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

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment.

Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text style transfer.

Jiao Sun, Xuezhe Ma, and Nanyun Peng. 2021. AESOP: Paraphrase generation with adaptive syntactic control. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5176–5189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2022. Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models.

Chongyang Tao, Shen Gao, Juntao Li, Yansong Feng, Dongyan Zhao, and Rui Yan. 2021. Learning to organize a bag of words into sentences with neural networks: An empirical study. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1682–1691.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Ke Wang, Hang Hua, and Xiaojun Wan. 2019a. Controllable unsupervised text attribute transfer via editing entangled latent representation.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2019b. Harnessing pre-trained neural networks with rules for formality 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 3573–3578, Hong Kong, China. Association for Computational Linguistics.

Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Songhua Xu, Shaohui Yang, and Francis Lau. 2010. Keyword extraction and headline generation using novel word features. In *Proceedings of the AAAI conference on artificial intelligence*, volume 24, pages 1461–1466.

Jian Yang, Shuming Ma, Dongdong Zhang, Zhoujun Li, and Ming Zhou. 2020. Improving neural machine translation with soft template prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5979–5989, Online. Association for Computational Linguistics.