Syntax Matters! Syntax-Controlled in Text Style Transfer

Zhiqiang Hu
University of Electronic Science and Technology of China
hzq950419@gmail.com

Roy Ka-Wei Lee
Singapore University of Technology and Design
roy_lee@sutd.edu.sg

Charu C. Aggarwal
IBM T. J. Watson Research Center
charu@us.ibm.com

Abstract

Existing text style transfer (TST) methods rely on style classifiers to disentangle the text's content and style attributes for text style transfer. While the style classifier plays a critical role in existing TST methods, there is no known investigation on its effect on the TST methods. In this paper, we conduct an empirical study on the limitations of the style classifiers used in existing TST methods. We demonstrate that the existing style classifiers cannot learn sentence syntax effectively and ultimately worsen existing TST models' performance. To address this issue, we propose a novel Syntax-Aware Controllable Generation (SACG) model, which includes a syntax-aware style classifier that ensures learned style latent representations effectively capture the syntax information for TST. Through extensive experiments on two popular TST tasks, we show that our proposed method significantly outperforms the state-of-the-art methods. Our case studies have also demonstrated SACG's ability to generate fluent target-style sentences that preserved the original content.

1 Introduction

Text Style Transfer (TST) is an increasingly popular natural language generation task that aims to change the stylistic properties (e.g., the sentiment of text) of the text while retaining its style-independent content (Hu et al., 2020). Due to the difficulty in obtaining training sentence pairs with the same content and differing styles, most existing methods are designed to perform TST in an unsupervised manner; the models only have access to non-parallel, but style-labelled sentences.

A popular TST approach is to leverage an adversarial learning autoencoder framework where a style classifier or discriminator is pre-trained to first disentangle the content and style latent representations, before using a decoder to generate the output sentence in the target style (Shen et al., 2017; Zhao et al., 2018; Fu et al., 2018; Chen et al., 2018). Another line of work proposed attribute-controlled generation methods where the style attribute latent vector is learned and combine with the latent representation of the text to generate output sentences in target style (Hu et al., 2017; Dai et al., 2019; Zhang et al., 2018a). Similar to the adversarial learning approach, the learning of the style attribute latent vector is guided using a pre-trained style classifier.

A common key component in the two aforementioned TST approaches is the usage of a style classifier. However, little is known about the effects of the style classifier on these models. For instance, is the style classifier effective in learning the style in the text? What aspects of the text style has the existing style classifier learned? Can the style classifiers distinguish text's syntax? Can the style classifier guide TST models to generate syntactically correct sentences and in the target style? This paper investigates these questions by conducting an empirical analysis of the style classifiers used in TST models.

Extending from our empirical study, we propose the Syntax-Aware Controllable Generation (SACG)[1] model, which includes a syntax-aware style classifier that ensure that the learned style latent representations effectively capture the syntax information for TST. Through extensive experiments with two popular TST datasets and human evaluation, we demonstrated SACG's ability to outperform the state-of-the-art baselines in the TST tasks.

2 Related Work

In recent years, studies on text style have attracted not only the linguist's attention but also that of many computer science researchers. Specifically,

[1]Code implementation: https://gitlab.com/bottle_shop/snlg/style/sacg

Proceedings of Recent Advances in Natural Language Processing, pages 566–575
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_064

computer science researchers are investigating the Text Style Transfer (TST) task that aims to change the text's stylistic properties while retaining its style-independent content. The recent comprehensive survey (Hu et al., 2020) summarizes the existing TST approaches.

Among these approaches, a popular line of research aims to infer a latent representation for an input sentence and manipulate the generated sentence's style based on this learned latent representation. Two techniques are commonly used to learn and manipulate the text's style latent representations: (1) adversarial learning and (2) attribute controlled generation. Shen et al. (2017) leverages an adversarial training scheme where a classifier is used to evaluate if an encoder is able to generate a latent content representation devoid of style. The text content latent representation is subsequently used to generate a specific style sentence using a style-dependent decoder. Similar works have been proposed where a classifier is pretrained to enable the adversarial learning process in TST models (Zhao et al., 2018; Fu et al., 2018; Chen et al., 2018; Logeswaran et al., 2018; Yin et al., 2019; Lai et al., 2019; Vineet et al., 2019).

Hu et al. (2017) proposed an attribute-controlled generation text style transfer model that utilized a Variational Autoencoder (VAE) (Kingma and Welling, 2013) to learn a sentence's latent representation z and leverage a style classifier to learn a style attribute vector s. Subsequently, z and s are input into a decoder to generate a target style sentence. Similar attribute-controlled generation methods have been proposed for the TST task (Dai et al., 2019; Zhang et al., 2018a; Li et al., 2019).

In the aforementioned methods, pretrained style classifiers played a vital role in guiding the TST task. However, these style classifiers are often pretrained without considering the syntax of sentences. We postulate that syntax is an important aspect of text style, especially in text formality style transfer. This paper empirically demonstrates the importance of modeling syntax in the TST task and proposes a novel syntax-aware TST method that outperforms state-of-the-art TST methods.

3 Empirical Study

Before presenting our proposed method, we first conduct an empirical study on the style classifiers used in existing TST methods. The goal is to examine the style classifiers' ability to learn the syntax style information in a given text.

Classifier	Test set	ACC	F	I
TextCNN	GYAFC	88.6	91.3	86.4
	Disordered	85.3	84.9	85.5
RNN	GYAFC	85.6	84.6	86.4
	Disordered	82.2	74.8	87.8
Transformer	GYAFC	84.9	86.7	83.7
	Disordered	82.9	80.5	84.6

Table 1: Style classifiers performance on *GYAFC* test set and corresponding *Disordered* test set. **ACC** refers to the accuracy on both formal and informal sentences, **F** refers to the accuracy on formal sentences, and **I** refers to the accuracy to informal sentences.

TextCNN (Kim, 2014), RNN (Cho et al., 2014), and Transformer (Vaswani et al., 2017) are popular style classifiers used in many TST models (Dai et al., 2019; Vineet et al., 2019; Luo et al., 2019; Li et al., 2019; Zhang et al., 2018b). In this study, we train the three style classifiers on *GYAFC* (Rao and Tetreault, 2018), which is a popular formality transfer dataset used in many TST studies. We first train and test the classifiers using the original GYAFC training and test set. Next, we perturb the sentence structure of the text in the GYAFC test set by disordering the sentences' word order. The underlying intuition is that there should be syntax differences between formal and informal sentences, and the style classifiers should be able to learn the syntactic style information. Therefore, perturbing the test set's sentence structure should worsen classification accuracy as the syntactic information in the text is corrupted.

The empirical experiment results show that syntax plays a crucial role in text's formality. Table 1 shows the results of our empirical experiments. We observed a small 2.9% decrease in style classification accuracy in the *disordered* test set compared to the original GYAFC test set. We further examined the style classifiers' performance in different classes. We noted that the classification accuracy for formal sentences sharply decreased as we disordered the test sentences' word order. However, such observations are not made for informal sentences; the classification accuracy remained fairly constant even when word order was disrupted in informal sentences. From the observations, we postulate that the style classifiers may have focused on the attribute words to predict the style of sentences while neglecting the syntactic information in their style predictions. Furthermore, the style classifiers may have regarded the perturbed sentences as in-

formal ones. Nevertheless, the syntax of informal sentences should be different from the perturbed sentences. The similar classification performance on perturbed sentences demonstrated the style classifiers' ineffectiveness in capturing different formality styles' syntax information. More importantly, the style classifier's inability to learn syntax information could misguide the TST model's decoder to generate fragmented sentences, especially when transferring sentences to the informal style.

4 Methodology

This section proposes the Syntax-Aware Controllable Generation (SACG) model, which addresses the ineffectiveness of existing TST methods in handling sentence structure when transferring text style. We first introduce Graph Convolutional Networks (GCNs). Subsequently, we explain how the GCNs are utilized to extract sentence structure information in our syntax-classifier and syntax-encoder, which are the two main components in our proposed SACG model. Finally, we describe the learning process of our SACG model.

4.1 GCN and Sentence Structure Representation

As a variant of convolutional neural networks (LeCun et al., 1998), graph convolutional networks (GCN) (Kipf and Welling, 2017) is designed for graph data and it has demonstrated effectiveness in modeling text data via syntactic dependency graphs (Marcheggiani and Titov, 2017). Consider a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V}$ (where $|\mathcal{V}| = n$ is the number of vertices in $\mathcal{G}$) is the set of graph node and $\mathcal{E}$ is the set of graph edges. Given a feature matrix $X \in \mathbb{R}^{n \times d}$, where row $x_i \in \mathbb{R}^d$ corresponds to a feature for vertex i, the propagation rule of a GCN is given as

$$H^{(l+1)} = \sigma(AH^{(l)}W^{(l)}), \quad (1)$$

where $H^{(l)} \in \mathbb{R}^{n \times d_l}$ is the feature matrix of the l-th layer and d_l is the number of features for each node in the l-th layer. $H^{(0)} = X$, $W^{(l)}$ is the weight matrix between the l-th and $(l+1)$-th layers, $A \in \mathbb{R}^{n \times n}$ is the adjacency matrix associated with the graph $\mathcal{G}$, and $\sigma(\cdot)$ is a non-linear activation function, such as ReLU or Leaky ReLU. In essence, a GCN takes in a feature matrix X as an input and extract a latent feature matrix $H^{(L)}$ as the output, where L is the number of layers in GCN.

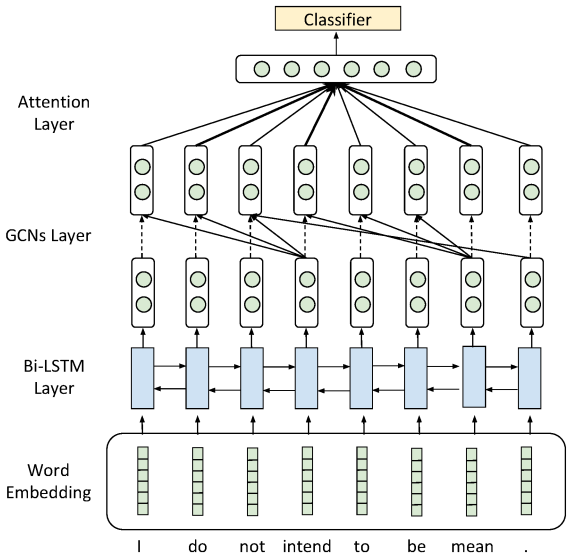

Figure 1: Architecture of syntax-aware style classifier.

Our goal is to extract and utilize sentence structure information to guide our SACG model to generate more plausible sentences. The syntactic relations between words in a sentence can be represented using dependency trees (Marcheggiani and Titov, 2017). A dependency tree can be regarded as a directed graph, and the GCNs can be used to extract the latent representation of sentence structure from the dependency trees. Previous studies have attempted to use GNCs to learn syntactic representation from dependency trees (Marcheggiani and Titov, 2017; Bastings et al., 2017). However, many of these existing techniques are over-parameterized, especially on huge datasets. To overcome this limitation, we employ a simpler approach where an adjacency matrix incorporated with directions is used to represent a sentence's structure. Specifically, the adjacency matrix A is used to represent the dependency relations of all words in the sentence. The column words are head words, and the row words are dependents. We set the element A_{ij} to 1 if there is a dependency between the i-th word (head) and the j-th word (dependent). Similar to (Marcheggiani and Titov, 2017), we add a self-loop for each node in the graph, where all diagonal elements of A are set to 1.

4.2 Syntax-Aware Style Classifier

In this subsection, we propose syntax-aware style classifier D to encode the syntactic information from the dependency trees better.

Figure 1 shows the architecture of our proposed syntax-aware style classifier. We first encode the to-

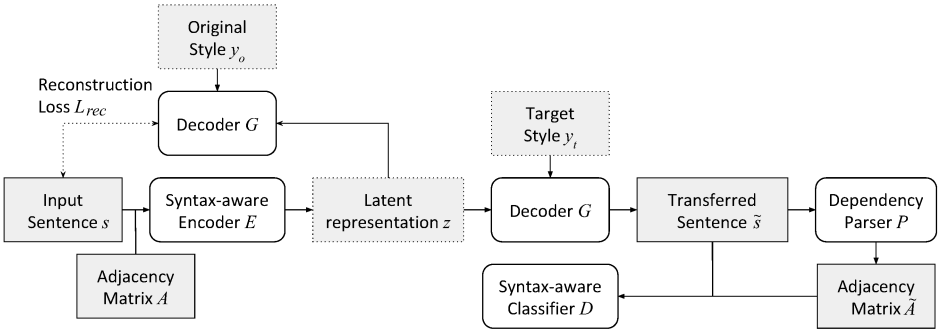

Figure 2: Framework of the Syntax-Aware Controllable Generation (SACG) model.

kens in a sentence of size n as $s = \{w_1, ..., w_n\}$ in the word embedding layer, where w_i is the i-th step input of Bi-LSTM. GCN has a limitation in capturing dependencies between nodes far away from each other in the graph. Therefore, instead of performing the graph convolution on the static word embeddings, we perform the GCN operations on top of the Bi-LSTM hidden states (Marcheggiani and Titov, 2017). As such, the GCN will only need to model the relationships for fewer hops. The Bi-LSTM states $H_{lstm} = \{h_{lstm,1}, ..., h_{lstm,n}\}$ serve as input $x_i = h_{lstm,i}$ to GCN, where $h_{lstm,i}$ is the concatenation of the forward and backward hidden states. We feed the hidden states into a L-layer GCN to obtain the hidden representations of each token, which are directly influenced by its neighbors no more than L edges apart in the dependency tree. Formally, the hidden representation of node i at the $(l+1)$-th layer of GCN is computed by the following equation:

$$h_i^{(l+1)} = \sigma(\sum_{j=1}^{n} A_{ij} W^{(l)} h_j^{(l)} + b^{(l)}) \quad (2)$$

where A is the adjacency matrix of dependency tree, $W^{(l)}$ and $b^{(l)}$ are the model parameters, and σ is an activation function. We obtain the hidden representation $h_i^{(L)}$ of node i after L GCN layers.

We noted that some node representations are more informative by gathering information from syntactically related neighbors through GCN. Thus, we utilize scaled dot-product attention (Vaswani et al., 2017) and averaging to aggregate the node representations to sentence representation:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (3)$$

where Q, K, V represent queries, keys, and values, respectively, $\frac{1}{\sqrt{d_k}}$ is the scaling factor. In practice, we feed the output $H^{(L)}$ of GCN to Q, K, V. Finally, we obtain the style prediction by feeding the sentence representation into a fully connected neural network followed by the softmax operation.

4.3 Syntax-aware Controllable Generation

Figure 2 shows the framework of our proposed Syntax-Aware Controllable Generation (SACG) model. For each input sentence s with attribute y_o and the corresponding adjacency matrix A, the syntax-aware encoder E encodes s to a latent representation $z = E(s, A)$. E is designed to extract sentence structure using the feature extractor of our proposed syntax-aware classifier. Subsequently, a decoder G decodes transferred sentence $\tilde{s} = G(z, y_t)$ or input sentence $s = G(z, y_o)$ based on the attribute controlling code y_t or y_o. We employ the Stanford neural dependency parser Stanza (Zhang et al., 2020) to generate the dependency tree for transferred sentences, and the corresponding adjacency matrix $\tilde{A}$. The transferred sentence $\tilde{s}$ and the corresponding adjacency matrix serve as the input of the syntax-aware classifier D, and the classifier will evaluate if the transferred sentence has the desired style.

We train the SACG model with classification loss L_{cla} and reconstruction loss L_{rec}.

Classification Loss L_{cla}: The classification loss ensures the transferred sentence is in the target style. To this end, we apply the pretrained syntax-aware classifier to guide the updates of related parameters such that the output sentence is predicted to be in the target style:

$$L_{cla} = -\mathbb{E}_{(s,y_o)\sim D}[logP(y_t|G(\tilde{s}), \tilde{A})] \quad (4)$$

where $G(\tilde{s})$ denotes a soft generated sentence based on Gumbel-Softmax distribution (Jang et al., 2017a) and the representation of each word is

defined as the weighted sum of word embeddings with the prediction probability at the current timestep. $\tilde{A}$ denotes the corresponding adjacency matrix of transferred sentence $\tilde{s}$.

Reconstruction Loss L_{rec}: The reconstruction loss attempts to preserve the original content information in the transferred sentences. Specifically, the loss function constricts the model to capture informative features to reconstruct the original sentence using the learned representations. Formally, we define L_{rec} as follows:

$$L_{rec} = -logP(s|z, y_o) \tag{5}$$

Where z denotes the hidden representation extracted by our syntax-aware encoder, and y_o denotes the original style of input sentence s.

Putting them together, the final joint training loss L is as follows:

$$L = L_{rec} + \lambda L_{cla} \tag{6}$$

Where λ is a balancing hyper-parameter to ensure that the transferred sentence has the target style while preserving the original content.

5 Experiments

5.1 Experiment Setting

Datasets. We evaluate our model on two popular style transfer tasks: (1) Sentiment transfer, and (2) formality transfer. The representative Yelp [2] restaurant reviews dataset (Shen et al., 2017) is selected for the sentiment transfer task. Following the same data preprocessing steps proposed in (Shen et al., 2017), reviews with a rating above 3 are considered positive, and those below 3 are negative. We adopt the same train, development, and test split as (Shen et al., 2017). Rao et al. (2018) released the GYAFC [3] (Grammarly's Yahoo Answers Formality Corpus) dataset to facilitate the formality style transfer task. We adopt the *Family&Relationship* (F&R) domain data for our experiments. Although it is a parallel dataset, the alignments are only used for evaluation and not for model construction. Table 2 shows the training, validation, and test splits of the Yelp and GYAFC datasets used in our experiments.

Baselines. We benchkmark SACG against 12 state-of-the-art TST models:*ARAE* (Zhao et al., 2018), *DualRL* (Luo et al., 2019), *DAST, DAST-C* (Li et al., 2019), *PFST* (He et al., 2020), *DRLST* (Vineet et al., 2019), *DeleteOnly, Template, Del&Retri* (Li et al., 2018), *DIRR* (Liu et al., 2021), and *HPAY* (Kim and Sohn, 2020).

Dataset	Attributes	Train	Dev	Test
Yelp	Positive	267K	38K	76K
	Negative	176K	25K	50K
GYAFC	Informal	51K	2.7K	1.3K
	Formal	51K	2.2K	1K

Table 2: Dataset statistics for Yelp and GYAFC.

Training. The experiments were performed on an Ubuntu 18.04.4 LTS system with 24 cores, 128 GB RAM, and Nvidia RTX 2080Ti. The word embeddings of 300 dimensions are learned from scratch. We use a single Bi-LSTM layer followed by 2 GCN layers. The hidden dimension of the latent representation z is set to 500, and the learnable vectors with 200 dimensions represent the style labels. The decoder is initialized by a concatenation of the latent representation z and attribute controlling code y. The syntax-aware style classifier is pretrained for evaluation and guiding the decoder's generation. After pretraining, the parameters of the classifier are fixed. We use the Gumbel-softmax to back-propagate the loss through discrete tokens from the classifier to the encoder-decoder model (Jang et al., 2017b). We empirically set the learning rate to 1×10^{-5} and the balancing parameter λ to 1.

5.2 Automatic Evaluation

We evaluate the proposed model and baselines on three criteria commonly used in TST studies: *transfer strength*, *content preservation*, and *fluency*.

Transfer strength. A TST model's transfer strength or its ability to transfer text style is commonly measured using *style transfer accuracy* (Hu et al., 2020). A syntax-aware style classifier is first pre-trained to predict the style label of the input sentence. The classifier is subsequently used to approximate the style transfer accuracy of the sentences' transferred style by considering the target style as the ground truth.

Content preservation. To quantitatively measure the amount of original content preserved after the style transfer operation, we employed four metrics used in previous work (Fu et al., 2018; Vineet et al., 2019; He et al., 2020):

- *BLEU*: The BLEU score (Papineni et al., 2002) is used to compare the style transferred sentences with the human references provided

[2] https://github.com/shentianxiao/language-style-transfer
[3] https://github.com/raosudha89/ GYAFC-corpus

Model	ACC(%)	BLEU	CS	WO	PPL	G-Score
ARAE (Zhao et al., 2018)	76.2	2.2	0.903	0.042	35	0.71
DeleteOnly (Li et al., 2018)	18.7	16.2	0.945	0.431	74	1.11
Template (Li et al., 2018)	44.7	19.0	0.943	0.509	102	1.32
Del&Retri (Li et al., 2018)	50.7	11.8	0.934	0.345	74	1.21
DualRL (Luo et al., 2019)	59.8	18.8	0.944	0.447	266	1.12
DAST (Li et al., 2019)	78.3	14.3	0.934	0.350	352	1.01
DAST-C (Li et al., 2019)	79.2	13.8	0.927	0.328	363	0.98
DRLST (Vineet et al., 2019)	49.8	2.7	0.909	0.342	**31**	1.06
PFST (He et al., 2020)	48.3	16.5	0.940	0.393	116	1.25
HPAY (Kim and Sohn, 2020)	43.1	10.4	0.942	0.418	92	1.17
DIRR (Liu et al., 2021)	71.8	18.2	0.942	0.451	145	1.28
SACG (ours)	**84.1**	**21.1**	**0.962**	**0.591**	73	**1.69**
Human0	**84.6**	24.6	**0.942**	**0.393**	**24**	**2.00**
Human1	83.8	24.3	0.931	0.342	27	1.89
Human2	83.6	24.6	0.932	0.354	27	1.91
Human3	82.1	**24.7**	0.931	0.354	27	1.90

Table 3: Performance of models on GYAFC dataset (Formality Transfer Task).

Model	ACC(%)	*self*-BLEU	CS	WO	PPL	G-Score
ARAE (Zhao et al., 2018)	83.2	18.0	0.874	0.270	79	1.35
DeleteOnly (Li et al., 2018)	84.2	28.7	0.893	0.501	130	1.53
Template (Li et al., 2018)	78.2	48.1	0.850	0.603	250	1.50
Del&Retri (Li et al., 2018)	88.1	30	0.897	0.464	88	1.66
DualRL (Luo et al., 2019)	79.0	**58.3**	0.970	**0.801**	117	1.98
DAST (Li et al., 2019)	90.7	49.7	0.961	0.705	181	1.76
DAST-C (Li et al., 2019)	93.6	41.2	0.933	0.560	274	1.49
DRLST (Vineet et al., 2019)	91.2	7.6	0.904	0.484	**65**	1.36
PFST (He et al., 2020)	85.3	41.7	0.902	0.527	94	1.78
HPAY (Kim and Sohn, 2020)	86.5	31.2	0.886	0.450	85	1.66
DIRR (Liu et al., 2021)	**94.2**	52.6	0.957	0.715	292	1.63
SACG (ours)	93.0	57.7	**0.971**	0.778	74	**2.23**

Table 4: Performance of models on Yelp dataset (Sentiment Transfer Task).

in the GYAFC dataset.

- *self-BLEU*: The *self*-BLEU score is adopted by comparing the style transferred sentence with its original sentence. This metric is used when human reference is not available.

- *Cosine Similarity*: Fu et al. (2018) calculated the cosine similarity between original sentence embedding and transferred sentence embedding. The two sentences' embeddings should be close to preserve the semantics of the transferred sentences.

- *Word Overlap*: Vineet et al. (Vineet et al., 2019) employed a simple metric that counts the unigram word overlap rate of the original and style transferred sentences.

Fluency. Generating fluent sentences is a common goal for most natural language generation models. GPT-2 (Radford et al., 2019) is a large-scale transformer-based language model that is pre-trained on large text corpus. We fine-tuned GPT-2 on the GYAFC and Yelp datasets and use the model to measure the perplexity (PPL) of transferred sentences. The sentences with smaller PPL scores are considered more fluent.

Geometric Mean (G-Score): We compute the geometric mean of *ACC*, *self-BLEU*, *BLEU*, *CS*, *WO* and *1/PPL*. Notably, we take the inverse of the calculated perplexity score because a smaller PPL score corresponds to better fluency.

5.2.1 Automatic Experiment Results

Table 3 shows the performance of the proposed SACG model and baselines on the formality transfer task. SACG has achieved the best G-Score, outperforming the state-of-the-art baselines. Nevertheless, we noted that none of the TST models could score well on all evaluation metrics. Many of the baselines can only perform well on transfer strength or content preservation, but not on both evaluation criteria. SACG has outperformed the baselines in G-Score, and achieve 84.1% transfer accuracy and 21.1 average BLEU score. The GYAFC dataset also provided the performances of four human references performing the formality transfer task on the test set. The BLEU score of

Model	Style(%)	Content	Fluency
DualRL	28.5	4.09	4.52
DAST	27.5	3.22	3.68
PFST	24.0	3.91	4.54
Del&Retri	25.5	2.61	3.23
SACG	**44.5**	**4.39**	**5.07**

Table 5: Human evaluation results on GYAFC dataset.

Model	TED	Model	TED
DRLST	19.2	DeleteOnly	18.2
ARAE	18.1	Template	17.9
DualRL	15.2	Del&Retri	21.0
DAST	16.6	HPAY	18.4
PFST	15.5	DIRR	15.5
DAST-C	16.9	SACG (ours)	**13.2**

Table 6: Average Tree Edit Distance (TED) of constituency tree between TST model generated sentences and 4 human references in GYAFC.

each human reference is calculated with the other three human references. Interestingly, we observe that SACG's performance on the three TST evaluation criteria is comparable and close to human references' performance.

Similar results were observed for the sentiment transfer task. Table 4 shows the performance of the proposed SACG model and baselines on the Yelp dataset. We computed the self-BLEU scores as no human references are provided for the Yelp test set. Similarly, SACG outperformed the baselines in G-score. We observe that the average style transfer accuracy in Yelp is 86.3%, which is significantly higher than GYAFC's average score of 66.0%. The difference in the average style transfer accuracy highlights the challenge of the formality transfer task. We also noted that most models performed better in this task compared to the formality transfer task. Nevertheless, the trade-off phenomenon between transfer strength and content preservation is still observed in the sentiment transfer task.

5.3 Human Evaluation

To further evaluate SACG's performance in generating syntactically correct sentences in target style, we conducted a human-based evaluation study. Specifically, we first randomly sampled 200 sentences from the GYAFC dataset. Next, we perform text style transfer for the sampled sentences using SAGC and four competitive baselines. Finally, we recruited two linguistics researchers (i.e., participants) to evaluate the style-transferred sentences generated by the TST models. The participants are asked to evaluate the generated sentences on the three criteria discussed in the earlier section. Specifically, for *Transfer Strength*, participants are asked to indicate if the generated sentences are in the target style (i.e., a binary true/false indicator). For *Content Presentation*, the participants are asked to rate the amount of content preserved in the generated sentences using a 6-point Likert scale. 1: no content presented, and 6: all content are preserved. Similarly, for *Fluency*, the participants are asked to rate fluency in the generated sentences using a 6-point Likert scale. 1: too many grammatical errors, and 6: perfect and fluent sentence.

To minimize biases, we do not display the models' names and we shuffled the order of the models when displaying their generated sentence. Therefore, the participants do not know which model generates a particular sentence.

5.3.1 Human Experiment Results

Table 5 shows the human evaluation results. For the transfer style, we compute the models' style transfer accuracy using the binary feedback from the participants. We compute the models' average 6-point Likert scores for content preservation and fluency criteria. SACG is observed to outperform the baselines in all three criteria. SACG is also rated to generate more syntactically sound and fluent sentence compared to the baselines. To check for participant bias, we compute the inter-annotator agreement between the participants. The Cohen's kappa coefficients on style transfer strength, content preservation, and fluency are 0.54, 0.76, and 0.72, respectively. The participants have substantially high agreement on the content presentation and fluency. However, the participants' agreement for style transfer strength is moderate as text formality is subjective, and the participants are only asked to perform binary indication.

5.4 Syntax Evaluation

As human references are available in the GYAFC dataset, we compare the syntax of the sentences generated by the TST models with the human references. Specifically, we compute the constituency tree edit distance (TED) to measure the syntactic similarity between generated sentences and human references. The intuition is that the TST model that could generate sentences with similar syntactic structure as the human references would likely have learned the syntactic information associated with the text formality style. To compute the constituency TED, we parse the sentences using Stan-

Model	ACC(%)	*self*-BLEU	BLEU	CS	WO	PPL
	GYAFC					
SACG	84.1	-	21.1	0.962	0.591	73
SACG w/o Syntax-aware Encoder	83.8	-	20.3	0.957	0.544	83
SACG w/o Syntax-aware Encoder & Classifier	78.7	-	15.6	0.943	0.446	223
	Yelp					
SACG	93.0	57.7	-	0.971	0.778	74
SACG w/o Syntax-aware Encoder	92.6	56.4	-	0.964	0.720	85
SACG w/o Syntax-aware Encoder & Classifier	89.3	49.1	-	0.943	0.697	230

Table 7: Results of ablation study.

	From formal to informal(GYAFC)	From positive to negative (Yelp)
Source	also , i dislike it when my father is unhappy .	We will definitely come back here!
DualRL	also i thrilled...	We will not come back here!
DAST	also, i r it when my father is men!	We will normally joke back here?
PFST	so i miss it when my father is 18.	We will not come back here again.
SACG (ours)	i also hate it when my father is unhappy !!	We will not come back here!

Table 8: Example outputs on the GYAFC and Yelp datasets. Grammatical errors are colored.

ford CoreNLP and compute the TED between constituency parsing trees.

Table 6 shows the syntax evaluation results. We noted that SACG outperformed the baseline in generating sentences that are syntactically similar to human references. This superior performance in both the formality transfer task and syntax evaluation suggests that SACG is able to learn the syntax information of formal and informal text to perform better text formality transfer.

5.5 Ablation Study

We also conducted an ablation study to further examine the importance of syntax-aware classifier and encoder in the SACG model. Table 7 shows the results of our ablation study. In the "w/o syntax-aware encoder" setting, we replace the syntax-aware encoder with a one-layer GRU (Cho et al., 2014). We noted a small decrease in performance for both formality transfer and sentiment transfer tasks when the encoder is replaced. In the "w/o syntax-aware encoder & classifier" setting, we further replace the syntax-aware classifier with a TextCNN (Kim, 2014) classifier. Interestingly, we observe a sharp decrease in performance for both formality transfer and sentiment transfer tasks. In particular, the absence of the syntax-aware encoder and classifier greatly worsens the fluency of the sentences. Our ablation study noted that the syntax-aware encoder and classifier play vital roles in ensuring SACG generates fluent target-style sentences that preserve the original content.

5.6 Case Study

We conduct some case studies by presenting randomly sampled examples and the corresponding style transferred output of SACG and the top three baselines ranked by G-Score. Table 8 shows the example outputs on the GYAFC and Yelp datasets. For the Yelp dataset, we observe that DualRL, PFST, and SACG are able to transfer the sentiment of the source sentence correctly. The generated sentences are also fluent and have preserved the original content (i.e., going back to a venue). The formality transfer task is observed to be more challenging, as we noted that most of the baselines could not generate acceptable output sentences. The baselines have generated output sentences with grammatical errors, making it harder to judge if the style has been successfully transferred. Albeit the difficulty of the task, SACG is able to generate a fluent sentence that preserved the original content.

6 Ethical Considerations

TST algorithms have many real-world applications. For example, these algorithms can improve target marketing messages' persuasiveness and integrate into writing tools to improve users' writing style. However, TST algorithms inherently run the risk of being misused for document forgery, impersonation, and sock-puppeting. To mitigate these risks, we will add access control to our code repository, and we would share our codes after the requester has acknowledged our ethical disclaimer.

7 Conclusion

In this paper, we empirically examined the style classifier used in existing TST models and demonstrated that the existing style classifier could not learn the text syntax effectively. We proposed SACG, a novel deep generative framework that considers syntax when learning style latent representation. We conducted extensive experiments on two benchmark datasets and benchmarked SACG against competitive TST models. The automatic and human-based evaluation experiment results showed that SACG outperforms state-of-the-art methods. Our case studies also demonstrated that SACG is able to generate fluent target-style sentences that preserved the original content. For future work, we will continue to explore other methods to improve the structural representations of text and incorporate them to perform better TST.

Acknowledgement

This research is supported by Living Sky Technologies Ltd, Canada under its research exploratory funding initiatives. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of Living Sky Technologies Ltd, Canada.

References

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967.

Liqun Chen, Shuyang Dai, Chenyang Tao, Haichao Zhang, Zhe Gan, Dinghan Shen, Yizhe Zhang, Guoyin Wang, Ruiyi Zhang, and Lawrence Carin. 2018. Adversarial text generation via feature-mover's distance. In *Advances in Neural Information Processing Systems*, pages 4666–4677.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuan-Jing 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.

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*.

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. In *International Conference on Learning Representations (ICLR)*.

Zhiqiang Hu, Roy Ka-Wei Lee, Charu C Aggarwal, and Aston Zhang. 2020. Text style transfer: A review and experimental evaluation. *arXiv preprint arXiv:2010.12742*.

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. JMLR. org.

Eric Jang, Shixiang Gu, and Ben Poole. 2017a. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Eric Jang, Shixiang Gu, and Ben Poole. 2017b. Categorical reparametrization with gumbel-softmax. In *Proceedings International Conference on Learning Representations 2017*. OpenReviews.net.

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.

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.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Chih-Te Lai, Yi-Te Hong, Hong-You Chen, Chi-Jen Lu, and Shou-De Lin. 2019. Multiple text style transfer by using word-level conditional generative adversarial network with two-phase training. 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 3570–3575.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Dianqi Li, Yizhe Zhang, Zhe Gan, Yu Cheng, Chris Brockett, Bill Dolan, and Ming-Ting Sun. 2019. Domain adaptive 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 3295–3304.

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.

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.

Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*, pages 5103–5113.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019*.

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*, pages 1506–1515.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

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

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.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

John Vineet, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434.

Di Yin, Shujian Huang, Xin-Yu Dai, and Jiajun Chen. 2019. Utilizing non-parallel text for style transfer by making partial comparisons. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5379–5386. AAAI Press.

Ye Zhang, Nan Ding, and Radu Soricut. 2018a. Shaped: Shared-private encoder-decoder for text style adaptation. In *Proceedings of NAACL-HLT*, pages 1528–1538.

Yuhao Zhang, Yuhui Zhang, Peng Qi, Christopher D. Manning, and Curtis P. Langlotz. 2020. Biomedical and clinical english model packages in the stanza python nlp library. *arXiv preprint arXiv:2007.14640*.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. 2018b. Style transfer as unsupervised machine translation. *arXiv*, pages arXiv–1808.

Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M Rush, and Yann LeCun. 2018. Adversarially regularized autoencoders. In *35th International Conference on Machine Learning, ICML 2018*, pages 9405–9420. International Machine Learning Society (IMLS).