Stable Style Transformer: Delete and Generate Approach with Encoder-Decoder for Text Style Transfer

Joosung Lee
Kakao Enterprise Corp., South Korea
rung.joo@kakaoenterprise.com

Abstract

Text style transfer is the task that generates a sentence by preserving the content of the input sentence and transferring the style. Most existing studies are progressing on non-parallel datasets because parallel datasets are limited and hard to construct. In this work, we introduce a method that follows two stages in non-parallel datasets. The first stage is to delete attribute markers of a sentence directly through a classifier. The second stage is to generate a transferred sentence by combining the content tokens and the target style. We experiment on two benchmark datasets and evaluate context, style, fluency, and semantic. It is difficult to select the best system using only these automatic metrics, but it is possible to select stable systems. We consider only robust systems in all automatic evaluation metrics to be the minimum conditions that can be used in real applications. Many previous systems are difficult to use in certain situations because performance is significantly lower in several evaluation metrics. However, our system is stable in all automatic evaluation metrics and has results comparable to other models. Also, we compare the performance results of our system and the unstable system through human evaluation. Our code and data are available at the link [1].

1 Introduction

Text style transfer is a task that generates a sentence while preserving the content in a given sentence but changing the source style. The style of the sentence refers to a predefined class (e.g. sentiment, formality, tense) and the content refers to the rest of the sentence except for the style. Lack of parallel data makes text style transfer task difficult. This problem cannot be solved by supervised learning because there are no right sentences.

[1] https://github.com/rungjoo/Stable-Style-Transformer

One previous method (Hu et al., 2017; Shen et al., 2017; Fu et al., 2018; Prabhumoye et al., 2018a; Logeswaran et al., 2018) of text style transfer is to learn latent representations to separate style and content from sentences. First, these approaches try adversarial training to learn a disentangled latent representation of the content and style. Secondly, a transferred sentence is generated from the decoder by combining the disentangled latent representation and the target style. However, the experimental results of (Lample et al., 2019) report that disentangled latent representation through adversarial training is hard to get and not necessary. Also, adversarial training is not effective to encode a sentence of various lengths into a vector representation of fixed length. Other methods of text style transfer do not depend on disentanglement. Dai et al. (2019a); Lample et al. (2019); Luo et al. (2019) do not attempt to find the disentangled latent representation in the sentence. Therefore, sentences with different styles are mapped to the same space. Xu et al. (2018a); Li et al. (2018a); Sudhakar et al. (2019); Wu et al. (2019) neutralize sentences by deleting style-dependent attribute markers. Remaining tokens resulting from the deletion of attribute markers are style independent, and then the content tokens and a style attribute are combined to generate the transferred sentence.

We propose an approach with two stages using Delete and Generate without adversarial training for disentanglement. (1) Attribute markers of a sentence are extracted by using a pre-trained classifier as a Delete model. Our method is model-agnostic and is not affected by the design of the classifier. Attribute markers found in a sentence are deleted. (2) A transferred sentence is generated by combining the target attribute and the content tokens after stage-1. The Generate model consists of an encoder and decoder with the Transformer structure.

In the method of deleting attribute markers, Li

Proceedings of The 13th International Conference on Natural Language Generation, pages 195–204,
Dublin, Ireland, 15-18 December, 2020.

et al. (2018b) deletes attribute markers via a statistical manner using a frequency ratio and Sudhakar et al. (2019); Xu et al. (2018b) delete attribute markers using attention weights of a classifier. Wu et al. (2019) deletes attribute markers by fusion of the frequency ratio and the attention weights. We introduce an intuitive delete method that uses a change in classifier probability. If a change in classifier probability is significant when limiting certain tokens in a sentence, the token is considered an attribute marker. Our method does not need to build attribute dictionaries or define attention weights like previous methods and easily control the trade-off between content and style.

We test our methods on two text style transfer datasets: sentiment of Yelp reviews and Amazon reviews. Evaluation metrics are conducted in terms of content, fluency, style accuracy, and semantic. The content and style accuracy are measured similarly to previous studies. Fluency is measured in two ways: general-fluency using pre-trained GPT-2 (Radford et al., 2019) and data-fluency using fine-tuned GPT-1 (Radford, 2018). Semantic is newly evaluated using BERTscore (Zhang* et al., 2020) in this paper. The goal of BERTscore is to evaluate semantic equivalence between two sentences. In this paper, we use a pre-trained model GPT and BERT (Devlin et al., 2019) that perform well in natural language processing/generation to evaluate transferred sentences with various automatic evaluations. Since automatic evaluations are not perfect evaluations of generated sentences, it is hard to know which system is the best, but we can determine which system has a problem. Comparative models are unstable in some evaluation metrics. But our proposed model has stable results for all automatic evaluations and is called SST (Stable Style Transformer). In addition, we first observe a point that can enhance the style controlling ability by generating sentences through latent space walking in the vector space of the style attribute token.

2 Related Work

One line of text style transfer research (Shen et al., 2017; Fu et al., 2018; Hu et al., 2017; Prabhumoye et al., 2018b; Logeswaran et al., 2018) is to separate content and style from sentences through disentangled learning. Hu et al. (2017) uses the VAE model to derive the disentanglement of the content between the generated sentence and the original sentence through KL loss. Shen et al. (2017) introduce the aligned auto-encoder and the cross aligned auto-encoder using learning discriminators. Fu et al. (2018) propose a multi-decoder and StyleEmbedding model. The multi-decoder model has decoders for each style, and the style embedding model uses only one decoder by inserting style embedding into the decoder. The methods of Prabhumoye et al. (2018b); Logeswaran et al. (2018) used back-translation to learn latent representations.

The second line of text style transfer research is not to rely on learning for latent representation. The first approach (Xu et al., 2018b; Li et al., 2018b; Sudhakar et al., 2019; Wu et al., 2019) is to find and delete tokens called attribute markers that are highly related to style. Li et al. (2018b) uses the delete method of attribute markers as a statistical method based on frequency ratio, and Sudhakar et al. (2019); Xu et al. (2018b) use the attention scores of the Transformer classifier and LSTM classifier, respectively. Wu et al. (2019) deletes attribute markers by fusion of the frequency ratio and attention scores. The second approach (Dai et al., 2019b; Lample et al., 2019; Luo et al., 2019) does not attempt to control content and style separately. Therefore, sentences with different styles are encoded to gather in the same latent representation space. Dai et al. (2019b); Lample et al. (2019) are based on the learning method using cycle reconstruction loss. Lample et al. (2019) reported that disentanglement is not easy and that latent representations learned through adversarial training are unnecessary because learned latent representations depend on style. Unlike the previous models, (Luo et al., 2019) learns dual models in two directions: style1 (e.g. negative) to style2 (e.g. positive) and style2 (e.g. positive) to style1 (e.g. negative) by reinforcement learning.

In language model research, the RNN-based language model is weak in long dependency. Therefore, the recent study of text style transfer (Dai et al., 2019b; Sudhakar et al., 2019; Wu et al., 2019) has been conducted with Transformer (Vaswani et al., 2017) which is known to have good performance in language modeling. Dai et al. (2019b) uses a method of using the encoder and the decoder of the Transformer, and Sudhakar et al. (2019) uses a method of fine-tuning the decoder to the style transfer datasets with the pre-trained GPT-1 as an initial state. Wu et al. (2019) solved the problem of text style transfer in a similar way to Text Infill-

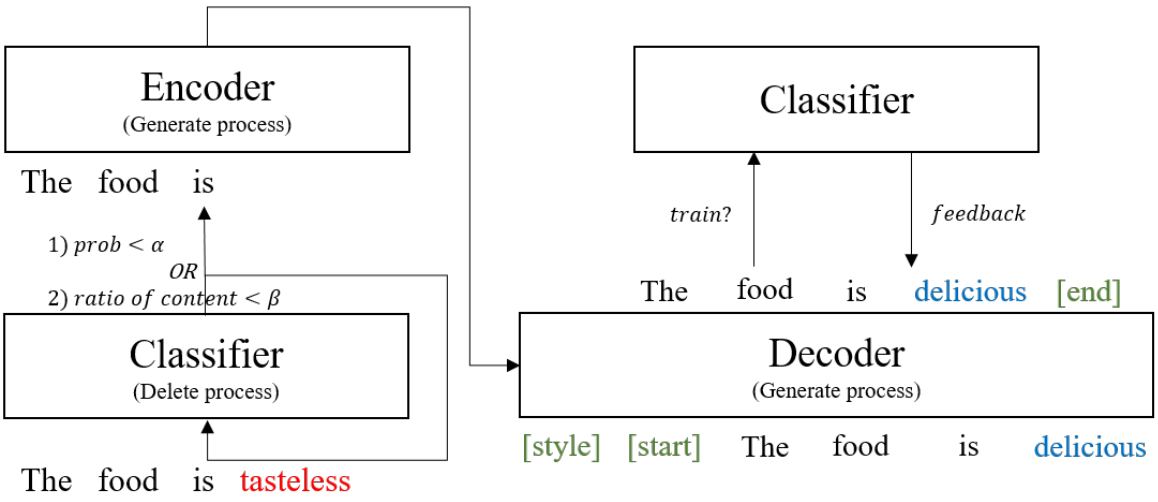

Figure 1: The proposed model framework consists of Delete and Generate process. Delete process is a method using a pre-trained classifier, and the Generate process consists of an encoder and a decoder. In the training time, our model receives feedback from the classifier's probability of the generated sentence.

ing or Cloze by presenting Attribute Conditional Masked Language Model (AC-MLM) using pre-trained BERT.

In this paper, we chose the first approach (Delete and Generate) that does not rely on latent representations in the second research line, referring to the results of Lample et al. (2019). Our system has a Transformer encoder and decoder because the style transfer task is given input text. If the system uses only a decoder such as Sudhakar et al. (2019), there is a disadvantage that it cannot include bidirectional encoding of the content token. Or, if only bidirectional encoders are used, such as AC-MLM, the position and length of the masking tokens to be filled in a sentence is not flexible.

3 Approach

In this section, we introduce our proposed method. The style transfer problem definition is described in Section 3.1. An overview of the model is shown in Section 3.2. The proposed generation process is introduced in Sections 3.3 and 3.4. The learning mechanism is described in Section 3.5.

3.1 Problem Statement

Given a dataset consist of sentence and label: $D = \{(\boldsymbol{x}_1, s_1), \cdots, (\boldsymbol{x}_N, s_N)\}$ where $\boldsymbol{x}_i$ is a sentence and s_i is a style attribute (e.g. sentiment) and N is the number of sentences in the dataset. Our goal is to train the model to generate a sentence $\boldsymbol{y}_i$ with a different style while preserving the content of the sentence $\boldsymbol{x}_i$. For example, if $\boldsymbol{x}_i$ is *"The food is salty and tasteless"* and s_i is *"negative"* attribute, then $\boldsymbol{y}_i$ is generated to mean *"The food is not salty and delicious"* which has a *"positive"* attribute. However the dataset is non-parallel, so the model cannot access $\boldsymbol{y}_i$ aligned with $\boldsymbol{x}_i$.

3.2 Model Overview

Our approach consists of two stages: Delete and Generate framework in Fig. 1. The first stage is the Delete process with a pre-trained style classifier. The pre-trained style classifier finds and deletes tokens that contain a lot of style attributes. The second stage is encoding the content tokens and combine them with a target style to generate a sentence. Both the encoder and the decoder have the Transformer structure, which is better than RNN and robust to long dependency.

3.3 Stage-1: Delete process

The stage-1 is the process of finding and deleting tokens for a given sentence and style attribute. In the previous studies, the strategies of deleting attribute markers are the frequency-ratio method and the classifier's attention score (or fusion of both). However, the frequency ratio method requires a pre-built vocabulary for the training dataset and it is difficult to understand contextual information. The attention score method has a limitation on the structure of the classifier, because it must learn the style classifier using self-attention regardless of accuracy. It is also unclear whether the attention score is directly proportional to the attribute.

We propose a novel method of removing attribute markers using a pre-trained classifier without a pre-built dictionary and attention scores. Our method is a model-agnostic method and it is more intuitive to find attribute markers than the previous method. Given an input sentence $\boldsymbol{x}$, the style probability follows:

$$p_{\boldsymbol{x}} = p_{\theta_C}(s|\boldsymbol{x}) \tag{1}$$

where p is a probability predicted by the classifier and s is style label. If we delete token t_i from the sentence $\boldsymbol{x}$, the style probability changes as follows:

$$p_{\boldsymbol{x},t_i} = p_{\theta_C}(s|\boldsymbol{x}, t_i) \tag{2}$$

where $\boldsymbol{x} = (t_1, t_2, \cdots, t_n)$ and n is the number of tokens in $\boldsymbol{x}$. The probability difference between Eq. 1 and Eq. 2 is defined as *Important Score(IS)* of the token(t_i):

$$IS^k_{t_i} = p_{\boldsymbol{x}^k} - p_{\boldsymbol{x}^k,t_i} \tag{3}$$

where $\boldsymbol{x}^k$ is the remained tokens after k tokens are deleted. The value of $IS^k_{t_i}$ determines how much the token t_i affects the style classifier. The token is deleted in order of the largest *IS*, and the Delete process ends if only one of the following two conditions: (1) $p_{\boldsymbol{x}^k}$ is less than α, or (2) the ratio of content tokens is less than β. α is a hyperparameter that determines that a sentence no longer has a source style attribute. β is a hyperparameter that determines how much of the content is preserved. The two hyperparameters make it easy to control the trade-off of content and style, and the experimental results are explained in Section 4.8.

3.4 Stage-2: Generate process

Our model generates a transferred sentence with the encoder and the decoder of the Transformer.

3.4.1 Encoder

All content tokens given as a result of Delete process are input to a bidirectional self-attention the Transformer encoder. Explicitly, the Transformer encoder maps content tokens $\boldsymbol{x}^c = (t_1, \cdots, t_m)$ to the continuous representation $\boldsymbol{z} = (z_1, \cdots, z_m)$ as follow:

$$(z_1, \cdots, z_m) = Encoder(t_1, \cdots, t_m; \theta_E) \tag{4}$$

3.4.2 Decoder

In order to generate a sentence with the desired style, two special tokens, *style* and *start*, are initially input to the decoder in Fig. 1. The position of special tokens is always fixed in front, so we did not add positional embedding. We use teacher-forcing at training time and no teacher-forcing at test time to generate sentences. If the generated token is the special token *end*, the Generate process ends. The decoder auto-regressively predicts the conditional probability of the next step token as follows:

$$softmax(\boldsymbol{y}_j) = p_{\theta_D}(t'_j|t'_1, \cdots, t'_{j-1}, \tilde{s}, \boldsymbol{z}) \tag{5}$$

where $\boldsymbol{y}_j$ is the logit vector of the decoder, $\tilde{s}$ is a desired style and t'_j is the predicted token in j step.

3.5 Training

Since we only have non-parallel datasets, we can't do supervised learning about transferred sentences. Therefore, we train SST to minimize two losses according to style conditions: s (source style) or $\hat{s}$ (target style).

3.5.1 Reconstruction loss

SST reconstructs the original sentence $\boldsymbol{x}$ conditioned on $\boldsymbol{x}^c$ and source style s. Reconstruction loss follows the equation:

$$\mathcal{L}_{rec} = -log\ p_{\theta_E,\theta_G}(\boldsymbol{x}|\boldsymbol{x}^c, s) \tag{6}$$

In non-parallel datasets, the reconstruction loss cannot be calculated if the style of the generated sentence is $\hat{s}$.

3.5.2 Style loss

If the model is only trained with reconstruction loss, the decoder will not see how to transform the style. Therefore, a discrepancy occurs between training time and test time. To learn how to generate sentence $\hat{\boldsymbol{x}}$ with a target style $\hat{s}$, we introduce style loss as follows:

$$\mathcal{L}_{style} = -log\ p_{\theta_C}(\hat{\boldsymbol{x}} = \hat{s}|\boldsymbol{x}^c, \hat{s}) \tag{7}$$

Style loss is measured by a pre-trained classifier to determine whether the transferred sentence has a $\hat{s}$. Since the generated sentence is a discrete space, we utilize soft-embedding of predicted tokens to optimize through style loss. When the SST is trained, the parameters of the classifier are not finetuned.

3.6 Model Details

The Transformer encoder and decoder consist of 3 layers, and each layer has 4 heads. The style classifier consists of 5 convolution filters based on Kim (2014). Text is tokenized using Byte-Pair-Encoding, and (word, style, position) embeddings

Dataset	Style	Train	Dev	Test
Yelp	Positive	270K	2000	500
	Negative	180K	2000	500
Amazon	Positive	277K	985	500
	Negative	278K	1015	500

Table 1: (Sentiment) Dataset statistics

are 256-dimensional vectors. In the Delete process, (α, β) is $(0.7, 0.5)$ at training time and observes the trade-off of content and style by changing parameters during test time.

4 Experiments

4.1 Dataset

In this paper, we test our model on two datasets, YELP and AMAZON, which are provided in Li et al. (2018b). The Yelp dataset is for business reviews, and the Amazon dataset is product reviews. Both datasets are labeled negative and positive and statistics are shown in Table 1.

4.2 Human References

Human references are used to measure human-BLEU and BERTscore. We used 2 Yelp human references and 1 Amazon human reference. **Yelp**: Li et al. (2018b) provides 1 human reference and 3 additional human references in Luo et al. (2019). We used 2 human references, one from Li et al. (2018b) and one (the best performance in automatic evaluation) from Luo et al. (2019), to increase reliability. **Amazon**: We used the human reference provided by Li et al. (2018b).

4.3 Previous Method

We compare the previous models with three approaches. The first comparisons are CrossAligned (Shen et al., 2017), [StyleEmbedding, multi-decoder] (Fu et al., 2018), and BackTranslation (Prabhumoye et al., 2018b), which attempt to separate content and style through latent representation learning. The second comparisons are [DeleteOnly, DeleteAndRetrieve] (Li et al., 2018b), UnpariedRL (Xu et al., 2018b) and [B-GST, G-GST] (Sudhakar et al., 2019), which delete attribute markers and then generate the sentence. [TemplateBased, RetrieveOnly] (Li et al., 2018b) return the target sentence through retrieve without generating. The final comparison is DualRL (Luo et al., 2019), which does not distinguish between content and style.

4.4 Automatic Evaluation

We evaluated the systems in 4 ways and results are shown in Table 2 and 3.

4.4.1 Content

Content preserving intensity is measured by G-BLEU, the geometric mean of self-BLEU and human-BLEU, as in previous works. A high BLEU score indicates that the model is good at content preservation.

In the Yelp dataset, RetireveOnly and BackTranslation are considered unstable models because G-BELU score is too low compared to other systems. In the Amazon datasets, CrossAligned and RetrieveOnly are too low compared to other systems.

4.4.2 Attribute

Most style transfer studies measure style accuracy using a classifier. We also evaluate style accuracy with a classifier (note that this is different from the one used in training).

In the Yelp dataset, StyleEmbedding, multi-decoder, and UnpairedRL have quite a low accuracy. In the Amazon datasets, StyleEmbedding, DeleteOnly, and DeleteAndRetrieve are unstable in style transfer.

4.4.3 Fluency

Fluency is considered the perplexity of the transferred sentence. We use GPT-1 and GPT-2, which is known to perform well as a language model. General-fluency (g-PPL) is measured using pretrained GPT-2 and data-fluency (d-PPL) is measured using GPT-1 (instead of GPT-2 due to GPU memory) finetuned to the dataset. General-Fluency is a general view because the language model is not fitted to the data, and data-fluency is an evaluation metric in terms of the specific data of style transfer tasks. The total-fluency (t-PPL) is the geometric mean of d-PPL and g-PPL, and lower values indicate better fluency.

In the Yelp dataset, TemplateBased is unstable because t-PPL is much larger than other systems. In the Amazon dataset, it is determined that the fluency of B-GST and G-GST is unstable.

4.4.4 Semantic

Semantic is measured using BERTscore. Unlike BLEU and ROUGE, BERTscore is an evaluation metric defined in continuous space. Pretrained model is used to calculate cosine similarity by extracting the contextual token embed-

dings from a human reference and a transferred sentence. BERTscore solves the limitations of previous metrics and measures a better correlation between the reference and the candidate. The original BERTscore ranged from 0 to 1, but we rescale it from 0 to 100 to clearly see the difference.

We set the unstable threshold as a margin point lower than the mean of all systems. The margin point is a gap between an average and a lower bound with 95% confidence considering all systems as the gaussian distribution in the BERTscore evaluation. CrossAligned, multi-decoder, RetrieveOnly, and BackTranslation have limitations on Yelp datasets. CrossAligned, multi-decoder, and RetrieveOnly have limitations on Amazon datasets.

SST : For comparison with other systems, we select the α and β of the appropriate trade-off points for style transfer and content preservation. When experimenting with the Yelp datasets, SST model is evaluated in two cases where (α, β) is (0.7, 0.5) and (0.7, 0.75). SST$_{(0.7,\ 0.5)}$ changes styles better with style accuracy of 79.5%, but SST$_{(0.7,\ 0.75)}$ has better performance on other metrics. In the Amazon datasets, SST model is evaluated when (α, β) is (0.6, 0.5). The effects of α and β are discussed in detail in Section 4.8.

4.5 Human Evaluation

Table 4 shows human evaluation results for content, fluecy, and style. Comparison models, StyleEmbedding and TemplatedBased, each have weaknesses in attribute and fluency. BackTranslation has weaknesses in content and semantic in automatic evaluation. In the yelp test set, we randomly sampled 250 and gave 6 people hired through the Amazon mechanical turk [2] evaluate content, fluency, and style between 1 and 5 points. As a result, BackTranslation and StyleEmbedding show the worst results for content, fluency, and style, respectively. Since humans evaluate fluency from a general point of view, the fluency performance of BackTranslation, which is poor in overall performance, and TemplateBased, which has poor t-PPL performance, are similarly bad. We confirm that our system has adequate performance in human evaluation as well as automatic evaluation.

4.6 Result Analysis

Human systems do not obtain the highest performance scores except for human-BLEU and BERTscore, which are calculated using human references. *But which of the sentences in human and machines is actually realistic? Probably human.* It is difficult to determine the best system with only automatic evaluation, but it is possible to determine which system is stable or unstable. If a system has significantly lower performance during the evaluation, it is considered unstable. The stable systems in the Yelp dataset are SST, DeleteOnly, DeleteAndRetireve, DualRL, B-GST, and G-GST. For the Amazon dataset, the stable systems are SST and TemplateBased. For all the metrics in both datasets, the stable systems are SST and DualRL. In automatic evaluation, DualRL outperforms SST, but DualRL does not share the model parameters of positive to negative and negative to positive tasks. Therefore, direct comparison is difficult because DualRL is regarded as two models.

We trained SST by changing the random seed of the model initialization several times and found that SST can always yield similar and comparable results. SST can be inferred as a stable system for the following reasons: (1) **G-BLEU**: Delete and Generate approaches show the stable performance of G-BLEU because the methods generate a sentence based on content tokens. There is no guarantee that content tokens will always be maintained, but content tokens help the generator. (2) **Attribute**: Our delete process is a method of determining whether certain tokens are deleted with *Important Score*. The direct and model-agnostic deletion is effective for neutralizing sentences. SST also improves a style accuracy by adding style control loss. (3) **Fluency**: TemplatedBased, B-GST, and G-GST show non-ideal fluency in d-PPL. TemplatedBased is considered unstable because it simply inserts attribute tokens of training data when generating test sentences. Since B-GST and G-GST use pre-trained GPT, they also have the ability to predict the distribution of tokens that are not in training data. The ability to predict generalized tokens is usually helpful, but can sometimes be harmful to d-PPL. SST, the Transformer encoder-decoder structure, learns only the distribution of given data and therefore has a stable d-PPL. (4) **Semantic**: Transformer language modeling is known to perform better on various tasks than RNN. Even in the style transfer task, the Transformer-based structures seem to reflect the linguistic characteristics.

We observed that unstable systems performed poorly in human evaluation as well in automatic

[2] https://www.mturk.com/

	Content			Attribute	Fluency			Semantic
Model	s-BLEU	h-BLEU	G-BLEU	Classifier(%)	d-PPL	g-PPL	t-PPL	BERTscore
SST (0.7, 0.5)	39.05	10.85	20.58	**79.5**	185.26	321.84	244.18	88.72
SST (0.7, 0.75)	49.09	12.66	**24.93**	70.4	197.82	295.9	**241.94**	**90.65**
CrossAligned	17.02	4.34	8.59	74.8	69.13	319.1	148.53	**88.12**
StyleEmbedding	71.8	13.65	31.3	**8.9**	121.66	379.81	214.96	90.56
multi_decoder	40.81	8.24	18.33	**46.4**	201.59	642.13	359.79	**88.35**
TemplateBased	48.67	12.86	25.02	79.7	3258.19	375.62	**1106.28**	89.71
DeleteOnly	33.94	9.29	17.75	84.8	171.66	279.55	219.06	89.28
DeleteAndRetrieve	34.48	9.82	18.4	87.7	137.04	343.75	217.04	89.39
RetrieveOnly	0.88	0.43	**0.61**	98.4	150.54	150.62	150.58	**86.33**
BackTranslation	0.67	0.52	**0.59**	96.2	30.53	148.77	67.39	**87.36**
UnpairedRL	42.29	10.6	21.17	**47.7**	328.8	735.1	491.63	88.51
DualRL	58.72	17.71	32.25	86.8	87.72	273.73	154.96	92.14
B_GST	43.45	13.49	24.21	86.1	165.59	184.02	174.57	91.78
G_GST	43.94	13.28	24.15	77.2	441.38	274.25	347.92	91.15
human: DRG	26.97	53.35	37.93	72.8	121.17	153.45	136.36	95.83
human: DualRL	36.79	33.02	34.86	77	178.63	196.15	187.19	95.83
input copy	100	21.01	45.84	3.5	69.72	131.91	95.9	93.18

Table 2: Automatic evaluation results of the Yelp dataset (s: self, h: human, G: geometric mean, f: fine-tuned, p: pre-trained). The red indicates that the evaluation score is significantly worse than other systems. Our model is referred to as SST(α, β). The bold black indicates the better performance of our systems for the four metrics that determine it is a stable system.

	Content			Attribute	Fluency			Semantic
Model	s-BLEU	h-BLEU	G-BLEU	Classifier(%)	d-PPL	g-PPL	t-PPL	BERTscore
SST (0.6, 0.5)	45.47	20.34	**30.41**	**66.5**	4.51	367.73	**40.72**	**89.17**
CrossAligned	0.76	0.61	**0.68**	74.8	1.11	119.37	11.51	**85.95**
StyleEmbedding	32.03	12.95	20.37	**42.4**	3.42	369.24	35.54	87.39
multi_decoder	16.48	6.61	10.44	70.3	1.39	343.72	21.86	**86.09**
TemplateBased	68.54	33.79	48.12	64.8	5.36	368.41	44.44	90.65
DeleteOnly	57.48	28.56	40.52	**50**	2.78	251.24	26.43	90.55
DeleteAndRetrieve	60.75	30.83	43.28	**52.4**	2.43	221.92	23.22	90.92
RetrieveOnly	2.82	1.23	**1.86**	82.3	5.65	135.22	27.64	**85.54**
B_GST	58.21	25.47	38.5	59.1	12448.44	193.73	**1552.94**	91.23
G_GST	51.02	21.1	32.81	57.3	18106	458.93	**2882.6**	89.48
human: DRG	47.67	100	69.04	46.9	12.38	132.18	40.45	100
input copy	100	47.6	68.99	18.5	3.76	188.33	26.61	93.77

Table 3: Automatic evaluation results of the Amazon dataset. Evaluation metrics are the same as Yelp, but BackTranslation, UnpariedRL, and DualRL do not provide results from Amazon datasets.

Model	Content	Fluency	Style
SST(0.7, 0.75)	3.32	3.37	3.3
BackTranslation	**2.69**	**3.15**	2.99
TemplatedBased	3.18	3.16	3.19
StyleEmbedding	3.56	3.49	**2.88**

Table 4: Human evaluation results. The higher the number, the better. Red means the worst result in the corresponding evaluation term.

evaluation. However, since performing human evaluation every time is expensive, choosing a stable system with automatic evaluation can be helpful.

Table 5 shows the samples of the generation of the models, which shows the lack of comparison models. In Yelp's negative to positive example, there are only SST and DualRL models that change the style while preserving content that includes *taste* and *price* of the food. In Yelp's positive to negative example, the *professionals* word contains a combination of style and content. In this case, the deletion and generation framework has the disadvantage of corrupting content information.

4.7 Ablation Study

If we use style loss for SST training, Table 6 shows that the style accuracy has 4 point gain. Fluency

	Yelp (negative to positive)	Yelp (positive to negative)
Input (source)	the food was so-so and very over priced for what you get .	these two women are professionals .
SST	the service is so-so and very reasonably priced for what you get .	these two women are rude .
CrossAligned	the food was fantastic and very very nice for what you .	these two dogs are hard down .
StyleEmbedding	the food was so-so and very over priced for what you get .	these two pot everywhere was .
DeleteOnly	the food was so-so and very over priced for what you get .	i would n't like these two women are professionals .
DeleteAndRetrieve	the service is fantastic and the food was so-so and the food is very priced for what you get .	these two scam women are professionals .
Back-translation	the food is delicious and the staff are very good for me .	this place is just not good .
UnpariredRL	the food was so-so and very over priced for what great qualities .	these two women are great .
DualRL	the food was surprising and very reasonably priced for what you get .	these two women are unprofessional .
B-GST	the food was amazing - so fresh and very good for what you get .	these two women are terrible liars .
G-GST	the food was priced right - so nice and very good for what you get .	these two women are condescending .
Human_DRG	the food was great and perfectly priced	these two women are not professionals .
Human_DualRL	the food was good and the price is low .	these two women are not professionals at all
	Amazon (negative to positive)	Amazon (positive to negative)
Input (source)	i have to lower the rating another notch .	it seems to be of very good quality in its build .
SST	love the rating another one ,	it seems to be of very poor quality in its build .
CrossAligned	i would recommend this for the price .	it s not be for a good game for my phone .
StyleEmbedding	i have to get by a one market .	it seems to be the num_extend is good nice high cases .
DeleteOnly	i have to lower the rating and it fits into another notch .	i have previously charged num_num different bt headsets that last num_num hours longer .
DeleteAndRetrieve	i have to lower the rating another notch and i love it .	initially it was very good quality in its build .
B-GST	i have lower levels for the other notch .	it seems to be of very good quality in taste .
G-GST	i have lower the steel another notch .	it seems to be of very good value in return .
Human_DRG	i have to raise the rating another notch .	it seems to be of very poor quality in its build

Table 5: Examples of comparison of generated sentences of AI systems. SST is our model. Attributes are colored. Red is negative and blue is positive.

	Con	Attr	Flu	Sem
Model	G-BLEU	Cls(%)	t-PPL	BERTscore
SST (0.7, 0)	19.11	82.2	306.65	89.96
- Style loss	19.78	78.2	341.51	89.84

Table 6: Ablation result of style loss in the Yelp dataset. (Con: content, Attr: attribute, Flu: Fluency, Sem: Semantic)

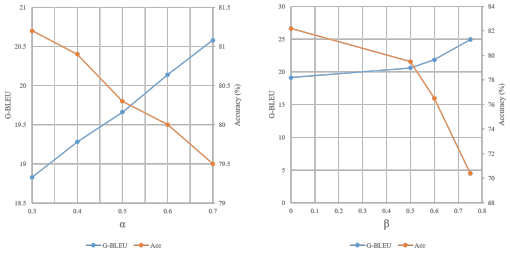

Figure 2: (a) Trade-off curve of G-BLEU and Style accuracy according to α (at $\beta = 0.5$) in Yelp (b) Trade-off curve of G-BLEU and Style accuracy according to β (at $\alpha = 0.7$) in Yelp.

and semantic are slightly better. It is observed that style loss improves the data-fluency, resulting in better total fluency. However, style loss decreases G-BLUE slightly by allowing the transferred sentence to change the attribute better.

4.8 Trade-off between Content and Style

With α and β we can simply adjust the trade-off of content and style. The results of Yelp are shown in

Source(negative)	when i was finally there , i was very disappointed .
after deletion	when i finally , i very .
style: negative	when i finally left , i was very disappointed .
↓	when i finally left , i was very disappointed.
↓	when i finally walked in , i was very disappointed .
↓	when i finally got , i was very happy .
style: positive	when i finally got , i was very happy .

Table 7: One sample of the Yelp dataset. SST generates a sentence from style vector space to negative to positive

Fig. 2. Smaller α and β allow the model to focus on style changes, while larger α and β allow the model to focus on content preserving. The trade-off of content and style changes linearly with α and is sensitive to β. The appropriate α and β depend on datasets.

4.9 Latent Space Walking

In this section we observe the transferred sentences according to the weight of positive and negative in the continuous style vector space. Ideally, a neutral sentence should be generated when the style attribute has the same weight for negative and positive. An example is shown in Table 7. A lot of data, like this example, don't show a neutral sentence even if the style has the same weight for the negative and positive. If we train our model to reflect this problem, we can expect better style control.

5 Conclusion and Future Work

We propose Stable Style Transformer (SST) that rewrites the sentences with Delete and Generate. SST is a system that can be used in the real world with overall stable results compared to other comparable systems. We show that filtering out unstable systems through human evaluation is expensive, so selecting a stable system through automatic evaluation can be helpful. The proposed direct and model-agnostic deletion method allows the classifier to intuitively delete attribute markers and easily handle the trade-off of content and style. In future work, we will study solutions for the case where attribute markers also contain content in the deletion and generation framework.

References

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019a. 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.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019b. 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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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.

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.

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

Juncen Li, Robin Jia, He He, and Percy Liang. 2018a. 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.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018b. 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.

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, Xu Sun, and Zhifang Sui. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5116–5122. International Joint Conferences on Artificial Intelligence Organization.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018a. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018b. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Alec Radford. 2018. Improving language understanding by generative pre-training.

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

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.

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)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Mask and infill: Applying masked language model for sentiment transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5271–5277. International Joint Conferences on Artificial Intelligence Organization.

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

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018b. 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.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. {BERTS}core: Evaluating text generation with {bert}. In *International Conference on Learning Representations*.