A SYNTAX-AWARE APPROACH FOR UNSUPERVISED TEXT STYLE TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised text style transfer aims to rewrite the text of a source style into a target style while preserving the style-independent content, without parallel training corpus. Most of the existing methods address the problem by only leveraging the surface forms of words. In this paper, we incorporate the syntactic knowledge and propose a multi-task learning based Syntax-Aware Style Transfer (SAST) model. Our SAST jointly learns to generate a transferred output with aligned words and syntactic labels, where the alignment between the words and syntactic labels is enforced with a consistency constraint. The auxiliary syntactic label generation task regularizes the model to form more generalized representations, which is a desirable property especially in unsupervised tasks. Experimental results on two benchmark datasets for text style transfer demonstrate the effectiveness of the proposed method in terms of transfer accuracy, content preservation, and fluency.

1 INTRODUCTION

Text style transfer is the task of transforming an input text by changing its stylistic attribute to a desired value while keeping the style-independent content unchanged. Focusing on different stylistic attributes, text style transfer has been addressed in many Natural Language Processing (NLP) applications such as sentiment translation (Xu et al., 2018), persona-based conversation (Li et al., 2016), text formalization (Rao & Tetreault, 2018), etc.

A descent text style transfer model is expected to produce outputs with the target style, without loss of the original semantics as well as readability. Unfortunately, since the parallel corpora with pairs of input and desired output are usually unavailable, models need to learn in an unsupervised setting, making it quite challenging to satisfy all of these criteria.

Previous dominant research line for unsupervised text style transfer disentangles the style and the content first, and then generates the output based on the disentangled content and the target style. The disentanglement is either achieved by learning an implicitly style-agnostic vector as the content via adversarial training (Shen et al., 2017; Fu et al., 2018; John et al., 2019), or by explicitly marking the text units as style-related or not through a dictionary or a pre-trained attention-based classifier (Li et al., 2018; Xu et al., 2018; Zhang et al., 2018a). Recently, Lample et al. (2019) found the disentanglement can be hardly met in practice and was not necessary for the transfer procedure. Departing from the disentangled representation, another research line (Zhang et al., 2018b; Lample et al., 2019; Luo et al., 2019) employs the back-translation technique to create online pseudo-parallel data, the quality of which can increase as training proceeds and in turn improve the model.

However, most existing methods only utilize the word information, thus may suffer from the data sparsity problem and fail to understand the underlying semantics of the input text. Take sentiment transfer as example, this limitation can lead to results simply appending an improper "great" to convert a sentence to positive sentiment. In this paper, we go beyond the word information and propose a Syntax-Aware Style Transfer (SAST) model for textual data by considering additional syntactic information such as part-of-speech (POS) tags. As illustrated in Figure 1, our SAST admits a multi-task learning scheme with the aim of generating aligned words and syntactic labels as the transferred outcome. Building on the encoder-decoder architecture, we couple the word emission and the syntactic label emission through shared hidden layers in the decoder. In this way, the model is encouraged to learn generalized representations explaining both the lexical knowledge and the syntactic knowledge, relieving possible word sparsity problem for style transfer.

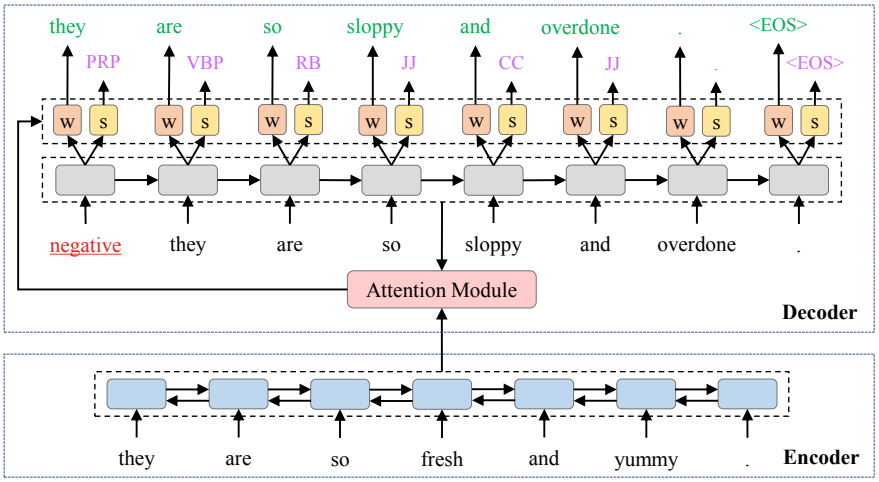

Figure 1: Model Overview.

We optimize the proposed SAST to achieve two goals: style conversion and content preservation. For style conversion, we enforce the model to generate outputs predicted as the target style by a pre-trained style classifier. For content preservation, we adopt the back-translation scheme following (Lample et al., 2019) but with enriched objectives on the syntactic labels: the output for an input under the transfer direction $s \rightarrow \tilde{s}$, when fed back into the model for the transfer direction $\tilde{s} \rightarrow s$, should reconstruct the input and the syntactic labels of the input. Moreover, besides the shared hidden layers, we further encourage the alignment between the generated words and syntactic labels by requiring the generated syntactic labels are consistent with the annotated syntactic labels (by an external NLP tool) for the generated words.

We evaluate our proposed model against state-of-the-art methods on two benchmark dataset: the sentiment dataset Yelp and the formality dataset GYAFC. Both automatic evaluation and human evaluation demonstrate our SAST can outperform the baselines in terms of transfer accuracy, semantic conservation, and fluency.

2 Related Work

Although unsupervised style transfer has gained remarkable success in the image domain (Johnson et al., 2016; Zhu et al., 2017; Choi et al., 2018), exploration for textual data is still limited due to the discreteness and complicated semantics of the natural language.

Earlier approaches transfer the input text based on the disentanglement between content and style. Hu et al. (2017) encourage the disentanglement by recovering the content and style from the output with an encoder and a style classifier. Shen et al. (2017) use adversarial discriminators to match the transferred samples with the real samples from the target style. Fu et al. (2018) match the encoded vectors from different styles. Johnson et al. (2016) seek for better separation by incorporating the content-orientated losses based on bag-of-words features, in addition to the style orientated losses in Fu et al. (2018). From a different perspective, Prabhumoye et al. (2018) assume a style-agnostic representation can be obtained via an external neural machine translation model.

Since implicitly disentangling the style and the content via hidden vectors is prone to semantic loss, explicit separation methods have been investigated. Xu et al. (2018); Zhang et al. (2018a) design a neutralization module to identify the indicators and an emotionalization module to stylize the neuralized text. Li et al. (2018) propose a delete-retrieve-generate pipeline which removes the style indicators based on term frequencies and then generates the output using the remaining content together with retrieved indicators from the target style corpus. Following this pipeline, recent works (Sudhakar et al., 2019; Wu et al., 2019) propose to employ the Transformer (Vaswani et al., 2017) based architecture BERT (Devlin et al., 2019) to extract the style indicators or generate the outputs.

Recent works start to skip the disentanglement step and address the text style transfer task with pseudo-parallel data created from back-translation as in unsupervised machine translation (Artetxe et al., 2018; Lample et al., 2018). Lample et al. (2019) use self-reconstruction and back-translation to optimize the model. Zhang et al. (2018b) initialize a pair of transfer models (in two directions) using phrase-based translation system and jointly update the models through iterative back-translation. Luo et al. (2019) further provide global supervision for the transferred output with rewards for style conversion and content preservation. Most recently, Dai et al. (2019) apply the Transformer to this task for its capability to capture long-term dependency.

3 SYNTAX-AWARE STYLE TRANSFER

We incorporate the syntactic knowledge and address the unsupervised text style transfer task with a multi-task learning framework: the input text is transformed into a sequence of words with aligned syntactic labels. The auxiliary syntactic label generation task encourages the model to learn generalized features which can benefit the prediction of both words and syntactic labels. Currently, our model allows only syntactic labels with one-to-one correspondence to the words. In this paper, we use the POS tags as the syntactic labels, leave other choices or multiple syntactic labels for future study.

3.1 PROBLEM DEFINITION

Consider a training corpus $\mathcal{D} = \{(x^i, s^i)\}_{i=1,\dots,N}$, where x^i is a text sequence and $s^i \in \mathcal{S}$ is the correponding style. $\mathcal{S} = \{s_j\}_{j=1,\dots,m}$ denotes all possible style types. As a preprocessing step, we extend the training corpus $\mathcal{D}$ to $\mathcal{D}^+ = \{(x^i, s^i, l^i)\}_{i=1,\dots,N}$ by providing an additional syntactic label sequence for each instance. For each text sequence $x = \{x_1, \dots, x_T\}$, we obtain its syntactic label sequence $l = \{l_1, \dots, l_T\}$ using an external NLP tool $\mathcal{F}$.

The objective of our SAST is to learn a joint conditional probability distribution $P_\theta(\tilde{x}, \tilde{l}|x, \tilde{s})$ to produce a text sequence $\tilde{x}$ with aligned syntactic labels $\tilde{l}$, conditioned on a given input text x and a target style $\tilde{s}$. The output text $\tilde{x}$ should possess the style $\tilde{s}$ while retaining the content (i.e., semantics not exhibiting the style) of x. We denote the marginal probability distribution of $\tilde{x}$ as $P^w_\theta(\tilde{x}|x, \tilde{s})$, which is the objective for models only leveraging word information.

3.2 MODEL OVERVIEW

We adopt the sequence-to-sequence encoder-decoder architecture (Sutskever et al., 2014) for our SAST model. For both the encoder and the decoder, we use the recurrent neural networks with the Gated Recurrent Units (GRU) (Cho et al., 2014). The attention mechenism (Bahdanau et al., 2015) is also utilized. An overview of our model architecture is shown in Figure 1.

3.2.1 ENCODER

The encoder, designed as a bi-directional GRU, maps the input text to the latent semantic space. Formally, given the input text $x = \{x_1, \dots, x_T\}$, we arrive at a set of hidden vectors $\mathbf{h} = \{\mathbf{h}_1, \dots, \mathbf{h}_T\}$ with $\mathbf{h}_t = [\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}]$, where $\overrightarrow{\mathbf{h}_t} = \overrightarrow{\text{GRU}}(\mathbf{h}_{t-1}, \mathbf{x}_t)$ and $\overleftarrow{\mathbf{h}_t} = \overleftarrow{\text{GRU}}(\mathbf{h}_{t+1}, \mathbf{x}_t)$ are the hidden outputs from the forward GRU and the backward GRU respectively.

3.2.2 DECODER

The decoder, designed as a uni-directional GRU, aims to generate a sequence of (word, syntactic label) pairs conditioned on the encoded hidden vectors $\mathbf{h}$ and the target style $\tilde{s}$. Following Lample et al. (2019), the style information is involved as the start token of the decoding process. Formally, at the k-th step, the hidden state $\mathbf{g}_k$ is computed as

$$\mathbf{g}_k = \begin{cases} \text{GRU}([\overrightarrow{\mathbf{h}_T}; \overleftarrow{\mathbf{h}_1}], \tilde{\mathbf{s}}), & \text{if } k = 1 \\ \text{GRU}(\mathbf{g}_{k-1}, \tilde{\mathbf{x}}_{k-1}), & \text{otherwise.} \end{cases} \tag{1}$$

We then apply the attention mechenism to assign an attention weight $\alpha^t_k = \text{softmax}(\mathbf{h}_t^\top \mathbf{W} \mathbf{g}_k)$ to each source word x_t, where W is a parameter matrix. Based on the attention weights, a context

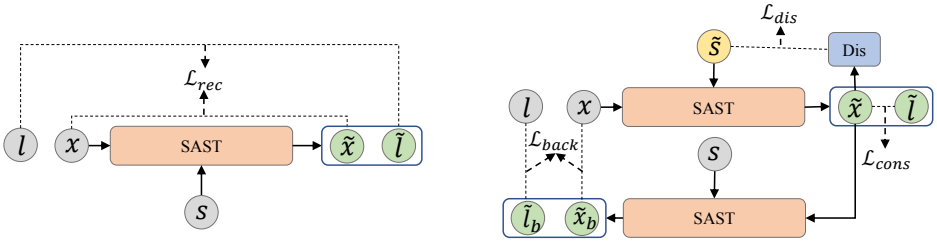

Figure 2: Training procedure of the proposed SAST. **Left**: the pre-training stage. **Right**: the regular training stage.

vector $\mathbf{c}_k$ is obtained as the weighted average of $\mathbf{h}$: $\mathbf{c}_k = \sum_t \alpha_k^t \mathbf{h}_t$. The hidden state $\mathbf{g}_k$ and the context vector $\mathbf{c}_k$ is further concatenated as an attentional hidden state $\mathbf{z}_k = [\mathbf{g}_k; \mathbf{c}_k]$. Assuming the next word $\tilde{x}_k$ and the next syntactic label $\tilde{l}_k$ are conditionally independent given $\mathbf{z}_k$, we predict $\tilde{x}_k$ and $\tilde{l}_k$ by:

$$P_\theta^w(\tilde{x}_k|\tilde{x}_{<k}, \tilde{l}_{<k}, x, \tilde{s}) = \text{softmax}(\mathbf{U}_w \mathbf{z}_k + \mathbf{b}_w) \tag{2}$$

$$P_\theta^l(\tilde{l}_k|\tilde{x}_{<k}, \tilde{l}_{<k}, x, \tilde{s}) = \text{softmax}(\mathbf{U}_l \mathbf{z}_k + \mathbf{b}_l) \tag{3}$$

where $\mathbf{U}_w$, $\mathbf{U}_l$, $\mathbf{b}_w$, and $\mathbf{b}_l$ are model parameters.

The shared hidden state $\mathbf{z}_k$ between the word prediction task and the syntactic label prediction task allows the interaction and complementation between lexical information and syntactic information, thus facilitating learning more generalized features.

Note that we only feed the generated token without the generated syntactic label from previous step as the input for the current step. The reasons are three-fold: (1) the knowledge for previous generated syntactic labels is encoded in $\mathbf{g}_{k-1}$; (2) our preliminary experiments did not show further improvements by feeding the syntactic label as input; (3) using only the word as input makes zero additional cost during inference compared to works not using the syntactic information. As a result, we can eliminate $\tilde{l}_{<k}$ from the conditions of Equation 2 and 3. The joint probability distribution $P_\theta(\tilde{x}, \tilde{l}|x, \tilde{s})$ and the marginal probability distribution $P_\theta^w(\tilde{x}|x, \tilde{s})$ can be formulated as

$$P_\theta(\tilde{x}, \tilde{l}|x, \tilde{s}) = \prod_{k=1}^{K} P_\theta^w(\tilde{x}_k|\tilde{x}_{<k}, x, \tilde{s}) P_\theta^l(\tilde{l}_k|\tilde{x}_{<k}, x, \tilde{s}) \tag{4}$$

$$P_\theta^w(\tilde{x}|x, \tilde{s}) = \prod_{k=1}^{K} P_\theta^w(\tilde{x}_k|\tilde{x}_{<k}, x, \tilde{s}) \tag{5}$$

3.3 Learning Algorithm

We first warm up the model with a pre-training stage based on self-reconstruction. Then we optimize the model with three losses during the regular training stage: a discrimination loss to encourage correct style transformation, a back-translation loss to ensure content preservation, and a consistency loss to enforce the alignment between the generated words and syntactic labels. Figure 2 shows the training procedure for these two training stages.

3.3.1 Self-Reconstruction as Pre-training

For non-parallel corpora, self-reconstruction can be used to guide the model for meaningful output. Given the input text x and its original style s, if the target style $\tilde{s} = s$, the model should reconstruct x as the output. In our multi-task learning based SAST, for $\tilde{s} = s$, the model is optimized to reconstruct both the text x and the syntactic labels l. Formally, the self-reconstruction loss is defined as

$$\mathcal{L}_{\text{rec}}(\theta) = \mathbb{E}_{(x,s,l)\sim\mathcal{D}^+}[-\log P_\theta(x, l|x^*, s)] \tag{6}$$

To avoid learning a trivial solution by copying source words, following Shen et al. (2017); Lample et al. (2019), we feed the model with a corrupted variant x^* (by word removal/permutation) of x.

Algorithm 1 The training algorithm for Syntax-Aware Style Transfer (SAST).

Input: non-parallel training corpus $\mathcal{D}$ and an external NLP tool $\mathcal{F}$
Pre-train a style discriminator P_ϕ on $\mathcal{D}$
Expand $\mathcal{D}$ to $\mathcal{D}^+$ by adding syntactic labels using $\mathcal{F}$
Make m splitions $\mathcal{D}^+[s_j]_{j=1,\dots,m} = \{(x^i, s^i, l^i) | (x^i, s^i, l^i) \in \mathcal{D}^+ \wedge s^i = s_j\}$
Pre-train θ on $\mathcal{D}^+$ by minimizing $\mathcal{L}_{\text{rec}}(\theta)$
for each iteration $r = 1, 2, \dots, L$ **do**
    for each style $s \in \mathcal{S}$ **do**
        Sample a minibatch of samples $\mathcal{B} = \{(x^i, s^i, l^i)\}_{i=1,\dots,n}$ from $\mathcal{D}^+[s]$
        Sample a target style $\tilde{s} \in \mathcal{S}$ with $\tilde{s} \neq s$
        Generate words $\tilde{x}^i \sim P_\theta^w(\tilde{x}|x^i, \tilde{s})$ for $\forall (x^i, s^i, l^i) \in \mathcal{B}$
        Compute the discrimination loss $\mathcal{L}_{\text{dis}}(\theta)$ based on Equation 7
        Compute the back-translation loss $\mathcal{L}_{\text{back}}(\theta)$ based on Equation 8
        Compute the consistency loss $\mathcal{L}_{\text{cons}}(\theta)$ based on Equation 11
        Compute total loss $\mathcal{L}(\theta)$ based on Equation 12
        Update θ based on $\nabla_\theta \mathcal{L}(\theta)$
    end for
end for

3.3.2 Discrimination Loss for Style Conversion

To improve the style conversion accuracy, we adopt a pre-trained style discriminator P_ϕ to justify the style of the transferred results. For a given input text x and a target style $\tilde{s}$, an output $\tilde{x} \sim P_\theta^w(\tilde{x}|x, \tilde{s})$ is expected to be predicted as having style $\tilde{s}$ by the discriminator. Therefore, the discrimination loss is defined as

$$\mathcal{L}_{\text{dis}}(\theta) = \mathbb{E}_{(x,s,l)\sim\mathcal{D}^+, \tilde{x}\sim P_\theta^w(\tilde{x}|x,\tilde{s})}[-\log P_\phi(\tilde{s}|\tilde{x})] \tag{7}$$

A problem in Equation 7 is the discrete property of the sampling operation $\tilde{x} \sim P_\theta^w(\tilde{x}|x, \tilde{s})$ impedes the gradient back-propagation from $\mathcal{L}_{\text{dis}}(\theta)$ to θ. We tackle this problem by replacing the discrete words with soft distributions: each word $\tilde{x}_k$ in $\tilde{x}$ is replaced with $\text{softmax}(\mathbf{o}_k/\tau)$, where $\mathbf{o}_k$ is the logit vector inside the softmax function of Equation 2, and τ is a a temperature hyper-parameter.

3.3.3 Back-Translation Loss for Content Preservation

To ensure content preservation, we adopt a back-translation loss to avoid generating results complying with the target style but irrelevant to the source content. Suppose for a training instance (x, s, l) and a target style $\tilde{s}$, we sample an output $\tilde{x} \sim P_\theta^w(\tilde{x}|x, \tilde{s})$. Then the model is enforced to reconstruct the source words x and syntactic labels l when we feed $\tilde{x}$ as the input text and s as the target style. Formally, the back-translation loss is defined as

$$\mathcal{L}_{\text{back}}(\theta) = \mathbb{E}_{(x,s,l)\sim\mathcal{D}^+, \tilde{x}\sim P_\theta^w(\tilde{x}|x,\tilde{s})}[-\log P_\theta(x, l|\tilde{x}, s)] \tag{8}$$

Each $\tilde{x}$, s, x, and l form a pseudo-parallel instance for our model. As training proceeds, the quality of $\tilde{x}$ gets improved and can gradually boost the performance. As in existing methods, the gradients are not back-propagated through $\tilde{x}$.

3.3.4 Consistency Loss for Word-Syntax Alignment

To guarantee the alignment between the generated words and syntactic labels, we design a consistency loss to pernalize incompatible outputs. Given an input text x and a target style $\tilde{s}$, we sample an output pair $(\tilde{x}, \tilde{l}) \sim P_\theta(\tilde{x}, \tilde{l}|x, \tilde{s})$. We then annotate $\tilde{x}$ with syntactic labels $\hat{l}$ using the external tool $\mathcal{F}$, and a legimate transfer should have $\tilde{l}_k = \hat{l}_k$ for $\forall k$. For training stability, we only apply the consistency contraint on the word prediction side. As the annotation operation is non-differential, we resort to the REINFORCE (Williams, 1992) technique to enable gradient estimation. Specifically, we define a consistency reward $\mathcal{Q}_k$ for step k as the log-probability of $\hat{l}$ given by the syntactic label prediction module

$$\mathcal{Q}_k = \log P_\theta^l(\hat{l}_k|\tilde{x}_{<k}, x, \tilde{s}) \tag{9}$$

The total expected consistency reward is

$$\mathcal{R}_{\text{cons}} = \mathbb{E}_{(x,s,l)\sim\mathcal{D}^+}[\sum_k \sum_{\tilde{x}_k} P_\theta^w(\tilde{x}_k|\tilde{x}_{<k}, x, \tilde{s})\mathcal{Q}_k] \tag{10}$$

which corresponds to a consistency loss

$$\mathcal{L}_{\text{cons}}(\theta) = -\mathcal{R}_{\text{cons}} \tag{11}$$

3.3.5 Full Training Objective

Combining the discrimination loss 7, the back-translation loss 8, and the consistency loss 11, The full objective of SAST is to minimize

$$\mathcal{L}(\theta) = \lambda_{dis}\mathcal{L}_{\text{dis}}(\theta) + \mathcal{L}_{\text{back}}(\theta) + \lambda_{cons}\mathcal{L}_{\text{cons}}(\theta) \tag{12}$$

where λ_{dis} and λ_{cons} are hyper-parameters for balancing the three losses. Our SAST belongs to the non-disentanglement based methods as with Zhang et al. (2018b); Lample et al. (2019); Luo et al. (2019), however, it is straightforward to adapt this multi-task learning scheme to the disentanglement based methods. The training algorithm is summarized in Algorithm 1.

4 Experiments

4.1 Datasets

We evaluate our model on two public datasets for unsupervised text style transfer: a sentiment dataset YELP (Li et al., 2018) and a formality dataset GYAFC (Rao & Tetreault, 2018). The YELP dataset consists of business reviews from Yelp, with each review categorized as positive or negative. We use the same train-dev-test split from Li et al. (2018). The GYAFC dataset consists of sentences from Yahoo Answers, with each sentence categorized as formal or informal. We use the data in the Family & Relationship domain. Following Luo et al. (2019), while GYAFC is a parallel corpus, we do not use the alignment information for training.

4.2 Implementation Details

We use the POS tags as syntactic labels, which are obtained by the spaCy[1] library. The bi-directional GRU encoder is single-layer with 256 hidden units for each direction, and the uni-directional GRU decoder is single-layer with 512 hidden units. The size of the word embeddings is 128. The temperature τ is set to 0.5. The balancing weights in Equation 12 are: $\lambda_{\text{dis}} = 0.1$ and $\lambda_{\text{cons}} = 0.005$, which are tuned on the development set. We employ the Adam algorithm with a learning rate of 10^{-3} for optimization. The batch size is set to 32. We train the model for 20K iterations in the pre-training stage and 60K iterations in the regular training stage. We sample the transferred output $\tilde{x}$ by greedy decoding during both the regular training stage and the inference stage. For the style discriminator, we the TextCNN model architecture Kim (2014). Our code will be released soon.

4.3 Baselines

We compare our SAST to various baselines, including (1) the implicit disentanglement based methods: CrossAligned (Shen et al., 2017), StyleEmbedding (Fu et al., 2018), MultiDecoder (Fu et al., 2018), and BackTrans (Prabhumoye et al., 2018); (2) the explicit disentanglement based methods: CycledRL (Xu et al., 2018), TemplateBased (Li et al., 2018), RetrieveOnly (Li et al., 2018), DeleteOnly (Li et al., 2018), and Del-Ret-Gen (Li et al., 2018); (3) non-disentanglement based methods: UnsuperMT (Zhang et al., 2018b) and DualRL (Luo et al., 2019).

4.4 Evaluation Metrics

Different text style transfer models are assessed based on their outputs on the test split of each dataset. The assessment focuses on three aspects: *transfer accuracy*, *content preservation*, and *fluency*. Both automatic evaluation and human evaluation are conducted.

[1] `https://spacy.io`

	YELP			GYAFC		
	ACC (%) ↑	BLEU ↑	PPL ↓	ACC (%) ↑	BLEU ↑	PPL ↓
CrossAligned (Shen et al., 2017)	74.4	17.9	43.0	71.2	3.6	**23.8**
StyleEmbedding (Fu et al., 2018)	8.2	42.3	43.2	24.3	7.9	60.4
MultiDecoder (Fu et al., 2018)	48.9	27.9	79.3	19.3	12.3	69.7
BackTrans (Prabhumoye et al., 2018)	94.7	5.0	**19.3**	60.6	0.9	113.1
CycledRL (Xu et al., 2018)	53.4	37.0	167.4	80.7	2.0	76.7
TemplateBased (Li et al., 2018)	84.1	45.5	158.1	50.4	35.2	151.0
RetrieveOnly (Li et al., 2018)	97.7	2.9	41.1	91.6	0.4	43.5
DeleteOnly (Li et al., 2018)	85.6	29.0	56.6	16.9	29.2	82.0
Del-Ret-Gen (Li et al., 2018)	88.3	31.1	54.6	51.7	21.2	73.1
UnsuperMT (Zhang et al., 2018b)	97.6	44.5	50.2	66.5	33.4	45.4
DualRL (Luo et al., 2019)	89.5	55.2	43.3	63.5	41.9	50.7
SAST	**98.3**	**60.2**	37.5	**94.1**	**45.5**	42.5

Table 1: Automatic evaluation results on the YELP dataset and the GYAFC dataset.

	YELP			GYAFC		
	Style ↑	Content ↑	Fluency ↑	Style ↑	Content ↑	Fluency ↑
CrossAligned (Shen et al., 2017)	2.7	2.6	3.2	2.8	1.5	3.0
MultiDecoder (Fu et al., 2018)	2.0	3.2	3.3	2.3	2.3	2.7
CycledRL (Xu et al., 2018)	2.8	3.4	3.4	2.8	1.3	2.6
TemplateBased (Li et al., 2018)	3.3	3.7	3.5	3.0	3.8	3.4
Del-Ret-Gen (Li et al., 2018)	3.6	3.8	3.9	2.7	2.9	2.9
DualRL (Luo et al., 2019)	**4.1**	4.1	4.0	3.5	3.7	3.7
SAST	**4.1**	**4.3**	**4.2**	**3.7**	**3.9**	**3.8**

Table 2: Human evaluation results on the YELP dataset and the GYAFC dataset.

Automatic Evaluation The transfer accuracy is measured by the prediction accuracy of a style classifier (same architecure but independent of P_ϕ) on the model outputs, using the target styles as the ground-truth labels. The content preservation is measured by the case-insensitive BLEU score, calculated using the `multi-bleu.perl` script, between the model outputs and reference outputs. We use the reference outputs from Luo et al. (2019) for YELP, which provides three more references (four in total) for each sample compared with Li et al. (2018). The GYAFC Rao & Tetreault (2018) dataset comes with four references for each sample and we directly use them for evaluation. The fluency is measured by the Perplexity (PPL) of the model outputs, resulted from a single-layer GRU based language model learned on the training set.

Human Evaluation We invite three human annotators to evaluate the outputs from different models for 200 test samples on each dataset. The annotators rate each transfer result from 1 (the lowest quality) to 5 (the highest quality) in terms of transfer accuracy, content preservation, and fluency.

4.5 Results

Table 1 presents the automatic evaluation results on the YELP dataset and the GYAFC dataset. Our SAST outperforms the baselines by a clear margin on both datasets in terms of transfer accuracy and content preservation. For fluency, SAST shows better performance than most baselines except BackTrans (Prabhumoye et al., 2018) on YELP and CrossAligned (Shen et al., 2017) on GYAFC. However, despite the low perplexity values, both methods are prone to information loss with low BLEU scores. Overall, by jointly modeling aligned words and syntactic labels, our SAST achieves better balance among the three metrics. Moreover, the non-disentanglement based methods show a clear advantage over the disentanglement based methods which tend to sacrifice the content preservation or fluency for better transfer accuracy (or the other direction).

Table 2 presents the human evaluation results on the YELP dataset and the GYAFC dataset for a subset of baselines due to high evaluation cost. For both datasets, our proposed SAST achieves the best results on all the three aspects. We notice that there exist some inconsistencies between the

YELP: **negative** → **positive**	
Input	i just received a delivery order from them and essentially wasted my money .
CrossAligned	i just received a problem job from them and a happy car back .
MultiDecoder	i just received the same time us , everyone came , find their time .
CycledRL	i just received a delivery order from them and excellent curds how delicious .
TemplateBased	i just received a delivery order from them and essentially good as it gets .
Del-Ret-Gen	i just received a delivery order from them and essentially perfect .
DualRL	i just received a delivery order from them and happy my money .
SAST	i just received a delivery order from them and essentially love this place !
YELP: **positive** → **negative**	
Input	they are so fresh and yummy .
CrossAligned	they are so fresh and everything and old .
MultiDecoder	they are so fresh and unprofessional .
CycledRL	<unk>
TemplateBased	they are n't return phone .
Del-Ret-Gen	we are so lazy they need .
DualRL	they are so bland and yummy .
SAST	they are so sloppy and overdone .
GYAFC: **formal** → **informal**	
Input	that is if you truly adore them .
CrossAligned	if you are talking about you ?
MultiDecoder	that is if you them .
CycledRL	it if you asked myself
TemplateBased	that is if you truly adore the i dont and i meanm .
Del-Ret-Gen	that is if you you them ... you .
DualRL	that is if you truly adore them
SAST	that is if u truly luv them
GYAFC: **informal** → **formal**	
Input	remember being friends with someone online is ok but there are limits .
CrossAligned	tell her and it is not sure that men are looking as well .
MultiDecoder	or with someone with that is but the best
CycledRL	he is not a good time and do not like it .
TemplateBased	remember being friends with someone online is ok but there are limit i wish you thes .
Del-Ret-Gen	you being friends with someone you is ok . yes , but there is you .
DualRL	remember being friends with someone online is ok but there are limits .
SAST	remember being friends with someone online is acceptable , but there are limits .

Table 3: Transferred outcomes of different models on four exemplary sentences.

human evaluation results and automatic evaluation results. For example, the transfer accuracy of CrossAligned is better than CycledRL on YELP for automatic evaluation results, which is opposite to the human evaluation results. Also, the differences between different models are much smaller for human evaluation than those for automatic evaluation. These phenomena can be attributed to the imperfection of the tools for automatic evaluation. In particular, the pre-trained classifier may not extract the appropriate patterns for specific styles, or give biased predictions (e.g., predicting as "positive" as long as there is a "great"); and the pre-trained language model may have quite different penalties for two equally improper (for human) words.

4.6 Qualitative Analysis

Table 3 presents the transferred outcomes of different models on four exemplary sentences. The results basically correspond to observations from the quantitative evaluation. On one hand, the implicit disentanglement based CrossAligned and MultiDecoder, tend to alter the original meaning of

the input. On the other hand, the explicit disentanglement based CycleRL, TemplateBased, and Del-Ret-Gen, and the non-disentanglement based DualRL either lose some content or make incomplete changes or produce some incompatible words. Our multi-task learning based SAST can generate outputs with better quality. As shown in the first example of Table 3, the outputs of CrossAligned and MultiDecoder significantly deviates from the original content; although the results of CycleRL, TemplateBased, Del-Ret-Gen, and DualRL change the appropriate position (i.e., *essentially wasted my money*) and reflect the *positive* sentiment, their coherence with the remaining content is not satisfying enough. In comparison, the result of our SAST, rewriting the *wasted my money* to *love this place*, is syntactically reasonable besides successful sentiment conversion and content preservation, proving the efficacy of modeling syntactic knowledge as an auxiliary task. The generated POS tags for the second example of Table 3 have been shown in Figure 1.

4.7 Ablation Study

To provide more insights into different components of our SAST framework, we evaluate several ablated variants on the YELP dataset: (1) *NoSyntax* which eliminates the syntactic label prediction task and syntax related losses; (2) $SAST - \mathcal{L}_{dis}$ which eliminates the discrimination loss; (3) $SAST - \mathcal{L}_{back}$ which eliminates the back-translation loss; and (4) $SAST - \mathcal{L}_{cons}$ which eliminates the consistency loss. The automatic evaluation results are shown in Table 4. Comparing SAST and NoSyntax, we can conclude modeling the syntactical information mainly improves content preservation. As for the different losses, we can observe: the discrimination loss has an important influence on the transfer accuracy; the back-translation loss is responsible for keeping the style-independent content; and the consistency loss further improves the content preservation. Furthermore, Table 4 shows the outputs of different ablated variants for transferring a negative sentence to a positive one. We can see SAST can generate the most reasonable result, while other variants can suffer from content loss, incomplete changes, or syntax errors.

	ACC (%) ↑	BLEU ↑	PPL ↑	**Input**: do n't go here unless you want to pay for crap .
SAST	98.3	**60.2**	37.5	highly recommend this place if you want to pay for perfection .
NoSyntax	98.5	54.4	38.8	do n't go here and you want to pay for gifts .
SAST $- \mathcal{L}_{\text{dis}}$	72.4	59.3	**37.3**	do n't go here unless you want to pay for perfect food .
SAST $- \mathcal{L}_{\text{back}}$	**99.8**	1.4	310.1	definitely definitely definitely definitely good
SAST $- \mathcal{L}_{\text{cons}}$	98.1	57.5	38.2	recommend here and you want to pay for amazing .

Table 4: Ablation study on the YELP dataset.

5 Conclusions

In this paper, we propose a multi-task learning based syntax-aware approach for the unsupervised text style transfer task. By jointly generating aligned words and syntactic labels, our SAST model learns more generalized latent representations explaining both lexical and syntactic knowledge, thus avoiding overfitting on the observed words. Quantitative and qualitative results on the two benchmark datasets demonstrate the benefits of our approach in terms of style conversion, content preservation, as well as fluency. Currently, we only use the shallow syntax, i.e., POS tags, as the syntactic labels, and it is promising to further explore other kinds of syntactic information (such as chunking labels and dependency labels) and investigate integrating multiple different syntactic labels. However, our SAST framework is limited to syntactic labels with a one-to-one correspondence to the words, unable to handle more complex syntax such as constituency/dependency parsing. Therefore, we would like to adapt our framework to address this limitation in our future research.

References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. In *ICLR*, 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8789–8797, 2018.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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*, pp. 4171–4186, 2019.

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

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1587–1596, 2017.

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

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pp. 694–711, 2016.

Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, et al. Unsupervised machine translation using monolingual corpora only. In *ICLR*, 2018.

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

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 994–1003, 2016.

Juncen Li, Robin Jia, He He, and Percy Liang. 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*, pp. 1865–1874, 2018.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. A dual reinforcement learning framework for unsupervised text style transfer. In *IJCAI*, 2019.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 866–876, 2018.

Sudha Rao and Joel Tetreault. 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*, pp. 129–140, 2018.

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

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. Transforming delete, retrieve, generate approach for controlled text style transfer. In *EMNLP*, 2019.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

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

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. Mask and infill: Applying masked language model to sentiment transfer. In *IJCAI*, 2019.

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

Yi Zhang, Jingjing Xu, Pengcheng Yang, and Xu Sun. Learning sentiment memories for sentiment modification without parallel data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1103–1108, 2018a.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. Style transfer as unsupervised machine translation. *arXiv preprint arXiv:1808.07894*, 2018b.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.