Semi-supervised Text Style Transfer: Cross Projection in Latent Space

Mingyue Shang\textsuperscript{1}, Piji Li\textsuperscript{2}, Zhenxin Fu\textsuperscript{1}, Lidong Bing\textsuperscript{4}, Dongyan Zhao\textsuperscript{1,3}, Shuming Shi\textsuperscript{2}, Rui Yan\textsuperscript{1,3}\textsuperscript{†}

\textsuperscript{1}Wangxuan Institute of Computer Technology, Peking University, China
\textsuperscript{2}Tencent AI Lab, Shenzhen, China, \textsuperscript{3}Center for Data Science, China
\textsuperscript{4}R&D Center Singapore, Machine Intelligence Technology, Alibaba DAMO Academy

\texttt{\{shangmy, fuzhenxin, zhaodongyan, ruiyan\}@pku.edu.cn}
\texttt{\{pijili, shumingshi\}@tencent.com, l.bing@alibaba-inc.com}

Abstract

Text style transfer task requires the model to transfer a sentence of one style to another style while retaining its original content meaning, which is a challenging problem that has long suffered from the shortage of parallel data. In this paper, we first propose a semi-supervised text style transfer model that combines the small-scale parallel data with the large-scale nonparallel data. With these two types of training data, we introduce a projection function between the latent space of different styles and design two constraints to train it. We also introduce two other simple but effective semi-supervised methods to compare with. To evaluate the performance of the proposed methods, we build and release a novel style transfer dataset that alters sentences between the style of ancient Chinese poem and the modern Chinese.

1 Introduction

Recently, the natural language generation (NLG) tasks have been attracting the growing attention of researchers, including response generation (Vinyals and Le, 2015), machine translation (Bahdanau et al., 2014), automatic summarization (Chopra et al., 2016), question generation (Gao et al., 2019), etc. Among these generation tasks, one interesting but challenging problem is text style transfer (Shen et al., 2017; Fu et al., 2018; Logeswaran et al., 2018). Given a sentence from one style domain, a style transfer system is required to convert it to another style domain as well as keeping its content meaning unchanged. As a fundamental attribute of text, style can have a broad and ambiguous scope, such as ancient poetry style v.s. modern language style and positive sentiment v.s. negative sentiment.

Building such a style transfer model has long suffered from the shortage of parallel training data, since constructing the parallel corpus that could align the content meaning of different styles is costly and laborious, which makes it difficult to train in a supervised way. Even some parallel corpora are built, they are still in a deficient scale for neural network based models. To tackle this issue, previous works utilize nonparallel data to train the model in an unsupervised way. One commonly used method is disentangling the style and content from the source sentence (John et al., 2018; Shen et al., 2017; Hu et al., 2017). For the input, they learn the representations of styles and style-independent content, expecting the later only keeps the content information. Then the content-only representation is coupled with the representation of a style that differs from the input to produce a style-transferred sentence. The crucial part of such methods is that the encoder should accurately disentangle the style and content information. However, Lample et al. (2019) illustrated that disentanglement is not a facile thing and the existing methods are not adequate to learn the style-independent representations.

Considering the above-discussed issues, in this paper, instead of disentangling the input, we propose a differentiable encoder-decoder based model that contains a projection layer to build a bridge between the latent spaces of different styles. Concretely, for texts in different styles, the encoder converts them into latent representations in different latent spaces. We introduce a projection function between two latent spaces that projects the representation in the latent space of one style to another style. Then the decoder generates an output using the projected representation.

In the majority of cases, nonparallel corpora of different styles are accessible. Based on these datasets, it is feasible to build small-scale paral-
lel datasets. Therefore, we design two kinds of objective functions for our model so that it could be trained under both supervised settings and unsupervised settings. With the parallel data, the model learns the projection relationship and the standard representation of the target latent space from ground-truth. Without the parallel data, we train the model by back-projection between the source and target latent space so that it could learn from itself. During training, we incorporate these two kinds of signals to train the model in a semi-supervised way.

We conduct experiments on an English formality transfer task that alters the sentence between formal and informal styles, and a Chinese literal style transfer task that alters between the ancient Chinese poem sentences and modern Chinese sentences. We evaluate the performance of models from the degree of content preservation, the accuracy of the transferred styles, and the fluency of sentences using both automatic metrics and human annotations. Experimental results show that our proposed semi-supervised method can generate more preferred output.

In summary, our contributions are manifolds:
• We designed a semi-supervised model that crossly projects the latent spaces of two styles to each other. In addition, this model is flexible in alternating the training mode between supervised and unsupervised.
• We introduce another two semi-supervised methods that are simple but effective to leverage both the nonparallel and parallel data.
• We build a small-scale parallel dataset that contains ancient Chinese poem style and modern Chinese style sentences. We also collect two large nonparallel datasets of these styles.\(^1\)

2 Related Works

Recently, text style transfer has stimulated great interests of researchers from the area of neural language processing and some encouraging results are obtained (Shen et al., 2017; Rao and Tetreault, 2018; Prabhumoye et al., 2018; Hu et al., 2017; Jin et al., 2019). In the primary stage, due to the lacking of parallel corpus, most of the methods employ unsupervised learning paradigm to conduct the semantic modeling and transfer.

Unsupervised Learning Methods. Mueller et al. (2017) modify the latent variables of sentences in a certain direction guided by a classifier to generate the sentence with classifier favored style. Hu et al. (2017) employ variational auto-encoders (Kingma and Welling, 2013) to conduct the style latent variable learning and design a strategy to disentangle the variables for content and style. Shen et al. (2017) first map the text corpora belonging to different styles to their own space respectively, and leverages the alignment of latent representations from different styles to perform style transfer. Chen et al. (2019) propose to extract and control the style of the image caption through domain layer normalization. Prabhumoye et al. (2018) and Zhang et al. (2018b) employ the back-translation mechanism to ensure that the input from the source style can be reconstructed from the transferred result in the target style. Liao et al. (2018) associate the style latent variable with a numeric discrete or continues numeric value and can generate sentences controlled by this value. Among them, many use the adversarial training mechanism (Goodfellow et al., 2014) to improve the performance of the basic models (Shen et al., 2017; Zhao et al., 2018).

To sum up, most of the existing unsupervised frameworks on the text style transfer focus on getting the disentangled representations of style and content. However, Lample et al. (2019) illustrated that the disentanglement not adequate to learn the style-independent representations, thus the quality of the transferred text is not guaranteed.


Nevertheless, building parallel corpus that could align the content meaning of different styles is costly and laborious, and it is impossible to build the parallel corpus for all the domains.

3 Task Formulation

We assume two nonparallel datasets \(A\) and \(B\) that are sentences in two different styles, denoted as style \(s^a\) and style \(s^b\), and a parallel dataset \(P\) that contains the pairs of sentences with two styles. The size of the three
datasets are denoted as $|A|$, $|B|$ and $|P|$, respectively. As the nonparallel data is abundant but the parallel data is limited, size $|A|$, $|B| \gg |P|$. Formally, two nonparallel datasets are denoted as $A = \{a_1, a_2, \ldots, a_{|A|}\}$ and $B = \{b_1, b_2, \ldots, b_{|B|}\}$, where $a_i$ and $b_i$ refer to the $i$-th sentences of $s^a$ style and $s^b$ style, respectively. It should be clear that as $A$ and $B$ are nonparallel datasets, the sentences in the two datasets are not aligned with each other, which means that $a_i$ and $b_i$ are not required to have the same or similar content meaning though they have the same subscript. The parallel dataset is denoted as $P = \{(a^p_1, b^p_1), (a^p_2, b^p_2), \ldots, (a^p_{|P|}, b^p_{|P|})\}$, where $(a^p_i, b^p_i)$ is the $i$-th pair of sentences that have the same content meaning but expressed in style $s^a$ and $s^b$ separately.

Due to the situation that the limited parallel data is deficient for neural network based model to train, our goal is to train a model in a semi-supervised way that could leverage the large volumes of nonparallel data to improve the performance. Given a sentence of the source style as input, the model learns to generate a target style output which preserves the meaning of the input to the greatest extent. In this paper, the style transfer process could be exerted in two directions, from $s^a$ to $s^b$ as well as from $s^b$ to $s^a$.

## 4 Basic Model Architecture

The text style transfer task could be interpreted as transforming a sequence of source style words to a sequence of target style words, which makes the sequence-to-sequence framework a suitable architecture. In this section, we describe the formulation of sequence-to-sequence (S2S) (Sutskever et al., 2014) model, which contains an encoder and a decoder, and our proposed models are built based on such an architecture.

Given an input, the encoder first converts it into an intermediate vector, then the decoder takes the intermediate representation as input to generate a target output. In this paper, we implement the encoder by a bi-directional Long Short-Term Memory (BiLSTM) (Hochreiter and Schmidhuber, 1997) and the decoder by a one layer LSTM.

Formally, with an input sentence $x = \{w^x_1, \ldots, w^x_T\}$ of length $T$, and a target sentence $y = \{w^y_1, \ldots, w^y_{T'}\}$ of length $T'$, where $w_i$ is the embedding of the $i$-th word, the probability of generating the target sentence $y$ given $x$ is defined as:

$$p(y|x) = \prod_{i=1}^{T'} p\left(w^y_i | w^x_1, w^x_2, \ldots, w^x_{i-1}\right) \tag{1}$$

In the training process, the encoder first encodes $x$ into an intermediate representation $c_x$. More specifically, at each time step $t$, the encoder produces a hidden state vector $h_t = [\overrightarrow{h}_t; \overleftarrow{h}_t]$, where $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ represents the forward and backward hidden state, respectively, and $[\cdot;\cdot]$ means concatenation. The intermediate vector is formed by the concatenation of the forward and backward last hidden states, denoted as $c_x = Enc(x) = [\overrightarrow{h}_T; \overleftarrow{h}_1]$. Then this vector is fed to the decoder to generate a target sentence step by step as formulated in Equation 1.

Such neural network based models always have a large amount of parameters which empower the model with the ability to fit complex datasets. However, when trained with limited data, the model may get so closely fitted to the training data that loses the power of generalization, and thus performances poorly on new data.

## 5 Cross Projection in Latent Space

In this paper, we propose a semi-supervised style transfer method named cross projection in latent space (CPLS) to leverage the large volumes of nonparallel data as well as the limited parallel data. Based on the S2S architecture, we consider the representations in the latent space, and propose to establish projection relations between different latent spaces. To be specific, in the S2S architecture, the encoder converting the input into an intermediate vector is a process of extracting the semantic information of the input, including the style and the content information. Thus the intermediate vectors could be seen as the compressed representations in the latent space of the inputs. Since texts in different styles are in different latent spaces, we introduce a projection function to project the intermediate vector from one latent space to another. To combine the nonparallel data and parallel data in training, we design two kinds of constraints for the projection functions.

Concretely, we first train an auto-encoder for each style to learn the standard latent spaces of the style. After that, we train the projection functions which are exerted on latent vectors to establish projection relations between different latent spaces. We design a cross projection strategy and
a cycle projection strategy to utilize the parallel data and nonparallel data. These two strategies are exerted iteratively in the training process, thus our model is compatible with the two types of training data. The following subsections give the details of the modules and the strategies of model training.

5.1 Denoising Auto-Encoder

To learn the latent space representations of styles, we train an auto-encoder for each style of text through reconstructing the input, which is composed of an encoder and a decoder. The encoder takes a sentence as input and maps the sentence to a latent vector representation, and the decoder reconstructs the sentence based on the vector. But a common pitfall is that the decoder may simply copy every token from the input, making the encoder lose the ability to extract the features. To alleviate this issue, we adopt the denoising auto-encoder (DAE) following the previous work (Lample et al., 2019) that tries to reconstruct the corrupted input. Specifically, we randomly shuffle a small portion of words in the original sentence to the corrupted input which are then fed to the denoising auto-encoder. For example, the original input is “Listen to your heart and your mind.”, then the corrupted version could be “Listen your to and heart your mind.”. The decoder is required to reconstruct the original sentence.

The training of a DAE model relies on the corpus of one style only, thus we train each DAE model using the nonparallel corpus of each style. Formally, the encoder and decoder of style $s^a$ and $s^b$ are referred as $\text{Enc}^{(a)}$, $\text{Dec}^{(a)}$, and $\text{Enc}^{(b)}$, $\text{Dec}^{(b)}$. Given sentences $a$ and $b$ from two styles, the corresponding encoder takes the sentence as input and converts it to the latent vector $c_a$ and $c_b$, respectively, by which each encoder therefore constructs the latent space for the corresponding style.

5.2 Cross Projection

To perform the style transfer task, we establish a projection relationship between latent spaces and cross link the encoder and the decoder from different styles. Take the transfer from style $s^a$ to $s^b$ for example. Given a sentence $a$ that is required to transferred to $b$, we cross link the $\text{Enc}^{(a)}$ as the encoder and $\text{Dec}^{(b)}$ as the decoder to form the style transfer model. However, considering the DAE models for different styles are trained respectively, therefor the latent vector spaces are usually different. The latent vector produced by $\text{Enc}^{(a)}$ is the representation on the latent space of style $s^a$ while the $\text{Dec}^{(b)}$ relies on the information and features from the latent space of style $s^b$. Therefore, we introduce a projection function to project the latent vector from the latent space of $s^a$ to $s^b$.

Concretely, after we get the $c_a$ from $\text{Enc}^{(a)}$, a projection function $f(\cdot)$ is employed to project $c_a$ from the latent space of style $s^a$ to the latent space of $s^b$, denoted as $\tilde{c}_b = f(c_a)$. Then the decoder $\text{Dec}^{(b)}$ takes the the projected vector $\tilde{c}_b$ as input and generates a sentence $\tilde{b}$ of style $s^b$ base on the prediction probability, denoted as $p_b(\tilde{b}|\tilde{c}_b)$.

It is worth noting that we have only exploited the nonparallel corpus for the style transfer up to now. Recall that our framework can employ both the parallel corpus and nonparallel corpus for the model training. With parallel corpus, we design the cross projection strategy. When the input $a$ is accompanied with a ground-truth $b$, we can get the standard latent representation of $b$ by the $\text{Enc}^{(b)}$, denoted as $c_b$. In order to align the latent vectors from different spaces, we design the constraints to
The cycle projection process is shown in Figure 2. projecting back and forth to reconstruct the input.

Then we define two losses as follows:

\[
\tilde{c}_b = f(\text{Enc}^{(a)}(a)) \tag{2}
\]

\[
l_{s_1} = \|\tilde{c}_b - c_b\| \tag{3}
\]

\[
l_{s_2} = -\log p(b|\tilde{c}_b) \tag{4}
\]

\[
l_s = \alpha * l_{s_1} + \beta * l_{s_2} \tag{5}
\]

where \(l_{s_1}\) measures the Euclidean distance between \(c_b\) and \(\tilde{c}_b\) and \(l_{s_2}\) is the negative log-likelihood (NLL) loss given \(a\) as input and \(b\) as the ground-truth. \(\alpha\) and \(\beta\) is the hyper-parameters that control the weight of \(l_{s_1}\) and \(l_{s_2}\).

Figure 1 shows the training process of the cross projection. Similarly, to transfer the style of text from \(s^b\) to \(s^a\), we cross link \(\text{Enc}^{(b)}(b)\) and \(\text{Dec}^{(a)}(a)\), and the projection function to project \(c_b\) to the latent space of \(s^a\) is denoted as \(g(\cdot)\).

### 5.3 Cycle Projection Enhancement

Inspired by the concept of back-translation in machine translation (Sennrich et al., 2015; Lample et al., 2018; He et al., 2016), we design a cycle projection strategy to train the model on the nonparallel data. Given the input without the ground-truth, we train the projection function \(f(\cdot)\) and \(g(\cdot)\) by projecting back and forth to reconstruct the input. The cycle projection process is shown in Figure 2.

Formally, for a sentence \(a\) in style \(s^a\), after getting its latent representation \(c_a\) in its own latent space by the \(\text{Enc}^{(a)}(a)\), we first project it to the latent space of \(s^b\) by \(f(\cdot)\) and get \(\tilde{c}_b = f(c_a)\). Then we exert \(g(\cdot)\) to project \(\tilde{c}_b\) back to the latent space of style \(s^a\), denoted as \(c_a = g(\tilde{c}_b)\). Finally \(c_a\) is fed to \(\text{Dec}^{(a)}\) to produce an output \(\tilde{a}\). Though the latent vector \(\tilde{c}_b\) has no reference to train, the latent vector \(c_a\) and the output \(\tilde{a}\) could be trained by treating \(c_a\) and \(a\) as the references. The loss functions are formulated as:

\[
c_a = g(f(\text{Enc}^{(a)}(a))) \tag{6}
\]

\[
l_{c_1} = \|c_a - c_a\| \tag{7}
\]

\[
l_{c_2} = -\log p(a|\tilde{c}_a) \tag{8}
\]

\[
l_c = \alpha * l_{c_1} + \beta * l_{c_2} \tag{9}
\]

### 5.4 Training Procedure

In the training stage, we first pretrain DAE models for each style separately on the nonparallel data to get the latent spaces of styles. Then we alternately apply the cross projection strategy and cycle projection enhancement on parallel data and nonparallel data with loss \(l_s\) and \(l_c\).

### 6 Straightforward Semi-supervised Methods

We also introduce another two semi-supervised methods as the baselines to provide a more comprehensive comparison with the proposed CPLS model. The two semi-supervised methods are built from different perspectives.

#### 6.1 Data Augmentation via Retrieval

One semi-supervised baseline is from the perspective of data augmentation. In order to alleviate the over-fitting issue caused by the small-scale parallel data, we propose a simple but efficient method that augments the parallel dataset by retrieving pseudo-references for the nonparallel datasets, denoted as DAR model.

**Pseudo-parallel Corpus Construction.** We employ Lucene\(^2\) to build indexes for the non-parallel corpus of each style. Then the pseudo-references are retrieved based on the TF-IDF scores. To build the pseudo-parallel corpus of style \(s^a\) and \(s^b\), we first samples 80,000 sentences in style \(s^a\) as queries. Specifically, each query searches a pseudo-reference from the nonparallel corpus of style \(s^b\) according to the TF-IDF based cosine similarity. The sampled query is therefore coupled with the pseudo-reference to form a training pair. We also conduct the same operation that uses sampled queries from style \(s^b\) to search pseudo-references from sentences of style \(s^a\). After searching pseudo-references from two sides,

\(^2\)http://lucene.apache.org/
we construct a pseudo-parallel corpus in the size of 150,000 pairs\textsuperscript{3}. With the pseudo-parallel corpus, the model is exposed to more information and thus the problem of over-fitting can be mitigated to some extent. Though the relevance of the content between the input sentence and the pseudo-reference is not guaranteed, the encoder could better learn to extract the language features and the decoder could also benefit from the weak contextual correlation information.

**Training Procedure.** In the training stage, we first train the S2S model on the pseudo-parallel dataset and save a checkpoint every 2,000 steps. We then calculate the BLEU score of checkpoints on the validation set and select the one with the highest BLEU score as the final pretrained model. Then based on this pretrained model, we fine-tune the parameters using the true parallel data.

### 6.2 Shared Latent Space Model

The second semi-supervised baseline is similar to CPLS that first trains a DAE model on the corpus of each style. Instead of building a bridge between the latent spaces of two styles through projection functions, in this method, we simply share the latent representations by cross linking the encoder and decoder, denoted as SLS model. Given a pair of sentence \((a^p, b^p)\), the encoder \(\text{Enc}^{(a)}\) encodes it into context \(c_a\), then the decoder \(\text{Dec}^{(b)}\) directly takes \(c_a\) to produce \(\tilde{b}\).

**Training Procedure.** For this method, we first pre-train two DAE models. Then we train the DAE models and the cross-linked S2S models from two transfer directions alternately. Considering the size imbalance between the nonparallel corpora and parallel corpus, to avoid the S2S falling into over-fitting too fast, We alternate the training in the form of training 20-step DAE models and then training one step S2S model.

### 7 Experiment

We conduct experiments on two bilateral style transfer tasks that each of them has a small-scale parallel corpus and two large-scale nonparallel corpora in two styles. DAR model and SLS model are treated as two baselines of CPLS model. In addition, we also train a S2S model with attention mechanism on the parallel data as the vanilla baseline. The following subsections elaborate the construction of the datasets and the detailed experimental settings.

#### 7.1 Datasets

We construct a Chinese literal style dataset with ancient poetry style and modern Chinese style, and an English formality style dataset with formal style and informal style. The texts of ancient Chinese poem, modern Chinese, formal English and informal English are referred as Anc.p, M.zh, F.en and Inf.en for short. The overall statistics are shown in Table 1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Parallel</th>
<th>Nonparallel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anc.p</td>
<td>3,362</td>
<td>269k</td>
</tr>
<tr>
<td>M.zh</td>
<td>400</td>
<td>77k</td>
</tr>
<tr>
<td>F.en</td>
<td>5,000</td>
<td>2,483k</td>
</tr>
<tr>
<td>Inf.en</td>
<td>998</td>
<td>4,891k</td>
</tr>
</tbody>
</table>

Table 1: Statistics of two style transfer datasets.

\textsuperscript{4} Some quires did not have corresponding retrieved result, therefore the total size is less than the number of queries.
Table 2: Case study: the sentences in the brackets are the translation of the corresponding sentence.

Formality Dataset. The formality dataset used in this paper is built based on the parallel dataset released by Rao and Tetreault (2018) which contains texts of formal and informal style.

With the released data, we randomly sample 5,000 sentence pairs from it as the parallel corpus with limited data volume. We then use the Yahoo Answers L6 corpus as the source which is in the same content domain as the parallel data to construct the large-scale nonparallel data. To divide nonparallel dataset into two styles, we train a CNN-based classifier (Kim, 2014) on the parallel data with annotation of styles and use it to classify the nonparallel data.

7.2 Experimental Settings

We perform different data preprocessing on different datasets. The Chinese literary datasets are segmented by characters instead of word to alleviate the issue of unknown words. Our statistics show that the average length of ancient poems is 9 while 17 for modern Chinese sentences. Therefore, we set the minimum length of the ancient poem as 3, and the maximum length of the modern Chinese sentence as 30. For the formality datasets, we use NLTK (Loper and Bird, 2002) to tokenize the texts and set the minimum length as 5 and the maximum length as 30 for both formal and informal styles.

We adopt GloVE (Pennington et al., 2014) to pretrain the embeddings, and the dimensions of the embeddings are set to 300 for all the datasets. The hidden states are set to 500 for both encoders and decoders. We adopt SGD optimizer with the learning rate as 1 for DAE models and 0.1 for S2S models. The dropout rate is 0.4. In the inference stage, the beam size is set to 5.

8 Comparison Models and Evaluations

8.1 Baselines

We train a S2S model with attention mechanism on the parallel data as the supervised learning baseline. Since the existing works on text style transfer seldom explore the semi-supervised methods, we propose DAR and SLS model as two semi-supervised baselines.

8.2 Automatic Evaluation Metrics

Following previous works (Prabhumoye et al., 2018; Zhang et al., 2018a), we employ BLEU score (Papineni et al., 2002) and style accuracy
Table 3: Automatic evaluation results on four style transfer tasks. Acc refers to the style accuracy.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>to Anc.P</th>
<th>GLEU</th>
<th>to M.zh</th>
<th>Acc</th>
<th>BLEU</th>
<th>GLEU</th>
<th>to F.en</th>
<th>Acc</th>
<th>BLEU</th>
<th>GLEU</th>
<th>to Inf.en</th>
<th>Acc</th>
<th>BLEU</th>
<th>GLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>to M.zh</td>
<td>4.20</td>
<td>3.24</td>
<td>74.8%</td>
<td>3.66</td>
<td>3.43</td>
<td>88.9%</td>
<td>33.90</td>
<td>14.06</td>
<td>71.8%</td>
<td>18.34</td>
<td>2.99</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>to Anc.P</td>
<td>5.89</td>
<td>4.49</td>
<td>81.9%</td>
<td>3.05</td>
<td>1.88</td>
<td>89.5%</td>
<td>41.41</td>
<td>16.77</td>
<td>63.5%</td>
<td>19.21</td>
<td>2.55</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>to Inf.en</td>
<td>6.33</td>
<td>5.21</td>
<td>80.4%</td>
<td>4.72</td>
<td>4.26</td>
<td>89.2%</td>
<td>44.72</td>
<td>18.52</td>
<td>63.5%</td>
<td>23.32</td>
<td>3.26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>to F.en</td>
<td>7.11</td>
<td>5.52</td>
<td>84.4%</td>
<td>4.95</td>
<td>4.12</td>
<td>91.3%</td>
<td>48.60</td>
<td>19.04</td>
<td>64.3%</td>
<td>27.25</td>
<td>3.61</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: The human annotation results of the S2S model and CPLS model from three aspects.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>Content</th>
<th>Style</th>
<th>Fluency</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2S</td>
<td>to M.zh</td>
<td>0.1875</td>
<td>0.5675</td>
<td>0.3575</td>
</tr>
<tr>
<td></td>
<td>to Anc.P</td>
<td>0.2275</td>
<td>0.5400</td>
<td>0.4425</td>
</tr>
<tr>
<td></td>
<td>to Inf.en</td>
<td>0.3175</td>
<td>0.4600</td>
<td>0.5800</td>
</tr>
<tr>
<td></td>
<td>to F.en</td>
<td>0.3625</td>
<td>0.5125</td>
<td>0.6325</td>
</tr>
<tr>
<td>CPLS</td>
<td>to M.zh</td>
<td>0.3100</td>
<td>0.5825</td>
<td>0.3050</td>
</tr>
<tr>
<td></td>
<td>to Anc.P</td>
<td>0.4375</td>
<td>0.6875</td>
<td>0.5475</td>
</tr>
<tr>
<td></td>
<td>to Inf.en</td>
<td>0.4675</td>
<td>0.4625</td>
<td>0.5725</td>
</tr>
<tr>
<td></td>
<td>to F.en</td>
<td>0.4600</td>
<td>0.5675</td>
<td>0.6200</td>
</tr>
</tbody>
</table>

Table 4: The human annotation results of the S2S model and CPLS model from three aspects.

as the automatic evaluation metrics to measure the content preservation degree and the style changing degree. BLEU calculates the N-gram overlap between the generated sentence and the references, thus can be used to measure the preservation of text content. Considering that text style transfer is a monolingual text generation task, we also use GLEU, a generalized BLEU proposed by (Napoles et al., 2015). To evaluate the extent to which the sentences are transferred to the target style, we follow Shen et al. (2017); Hu et al. (2017) that build a CNN-based style classifier and use it to measure the style accuracy.

8.3 Human Evaluation

We also adopt human evaluations to judge the quality of the transferred sentences from three aspects, namely content, style and fluency. These aspects evaluate how well the transferred text preserve the content of the input, the style strength and the fluency of the transferred text. Take the content relevance for example, the criterion is as follows: +2: The transferred sentence has the same meaning with the input sentence. +1: The transferred sentence preserves part of the content meaning of the input sentence. 0: The transferred sentence and the input sentence are irrelevant in content. The criteria for style strength and fluency are similar to the content relevance criterion.

To get the evaluation results, we first randomly sample 50 test cases from each dataset. As the style transfer is bilateral in this paper, there are 400 test cases in all. We invited four well-educated volunteers to score the results from the supervised baseline and the three semi-supervised models.

9 Results and Analysis

Evaluation Results. Table 3 presents the evaluation results of automatic metrics on the models. It can be seen that the BLEU scores and GLEU scores of the semi-supervised models on almost all the datasets are better than the baseline S2S model. This result indicates that the model benefits from the nonparallel data in terms of content preservation. One interesting thing is that the overall BLEU scores on the ancient poems and modern Chinese datasets are lower than other datasets. This result may be explained by the fact that the edit distance between formal and informal texts are smaller than between ancient poems and modern Chinese texts. Therefore, it is more challenging for model to preserve the content meaning when transferring between ancient poems and modern Chinese text. Among three semi-supervised models, CPLS model achieves the greatest improvement, verifying the effectiveness of the projection functions. However, the gain of CPLS model in the aspect of style accuracy is not that significant. A possible explanation may be the bias of the style classifier. Take the transfer task from ancient poems to modern Chinese text for example. We observe that the classifier tends to classify short sentences into ancient poems as length is an obvious feature. We analyse the sentences generated by S2S model and by the CPLS model, and the statistics show that the average length of the text generated by S2S model is shorter, which may lead to the bias of the style classifier. Therefore, we also adopt human evaluation to alleviate this issue.

Table 4 compares the human evaluation results of S2S model and CPLS model on all the datasets, which are calculated by the average score of the human annotations. As shown in the Table 4, the CPLS model outperforms the S2S model in
the aspects of the content preservation and style strength, and is on par in terms of fluency.

**Case Study.** We select the generated results of the S2S baseline and three semi-supervised models on two style transfer tasks due to the limited space as shown in Table 2. Compared with the Chinese literacy style datasets, the formality datasets are less challenging as discussed before. Thus it can be seen from the table that the generated sentences on the formality datasets are more fluent. For the task that transferring from the modern Chinese text to the ancient poem, the S2S generates a shorter sentence while the CPLS model generates the sentence that preserve the most content information.

10 Conclusion

In this paper, we design a differentiable semi-supervised model that introduces a projection function between the latent spaces of different styles. The model is flexible in alternating the training mode between supervised and unsupervised learning. We also introduce another two semi-supervised methods that are simple but effective to use the nonparallel and parallel data. We evaluate our models on two datasets that have small-scale parallel data and large-scale nonparallel data, and verify the effectiveness of the model on both automatic metrics and human evaluation.

11 Acknowledgement

We would like to thank the anonymous reviewers for their constructive comments. This work was supported by the National Key Research and Development Program of China (No. 2017YFC0804001), the National Science Foundation of China (NSFC No. 61876196 and NSFC No. 61672058). Rui Yan was sponsored by Tencent Collaborative Research Fund.

References


