The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Style Transfer in Text: Exploration and Evaluation

Zhenxin Fu,[1] Xiaoye Tan,[1] Nanyun Peng,[2] Dongyan Zhao,[1,3] Rui Yan[1,3]*
[1]Institute of Computer Science and Technology, Peking University, Beijing, China
[2]Information Science Institute, University of Southern California, California, USA
[3]Beijing Institute of Big Data Research, Beijing, China
{fuzhenxin, txye, zhaodongyan, ruiyan}@pku.edu.cn, npeng@isi.edu

Abstract

The ability to transfer styles of texts or images, is an important measurement of the advancement of artificial intelligence (AI). However, the progress in language style transfer is lagged behind other domains, such as computer vision, mainly because of the lack of parallel data and reliable evaluation metrics. In response to the challenge of lacking parallel data, we explore learning style transfer from non-parallel data. We propose two models to achieve this goal. The key idea behind the proposed models is to learn separate content representations and style representations using adversarial networks. Considering the problem of lacking principle evaluation metrics, we propose two novel evaluation metrics that measure two aspects of style transfer: transfer strength and content preservation. We benchmark our models and the evaluation metrics on two style transfer tasks: paper-news title transfer, and positive-negative review transfer. Results show that the proposed content preservation metric is highly correlate to human judgments, and the proposed models are able to generate sentences with similar content preservation score but higher style transfer strength comparing to auto-encoder.

Introduction

Style transfer is an important problem in many subfields of artificial intelligence (AI), such as natural language processing (NLP) and computer vision (Gatys, Ecker, and Bethge 2016; Gatys et al. 2016; Zhu et al. 2017; Li et al. 2017), as it reflects the ability of intelligence systems to generate novel contents. Specifically, style transfer of natural language texts is an important component of natural language generation. It facilitates many NLP applications, such as automatic conversion of paper title to news title, which reduces the human effort in academic news report. For tasks like poetry generation (Yan et al. 2013; Yan 2016; Ghazvininejad et al. 2016), style transfer can be applied to generate poetry in different styles. Nevertheless, the progress in style transfer of language is lagged behind other domains such as computer vision, largely because of the lack of parallel corpus and reliable evaluation metrics.

*Corresponding author: Rui Yan (ruiyan@pku.edu.cn)
Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Sequence to sequence (seq2seq) neural network models (Sutskever, Vinyals, and Le 2014) have demonstrated great success in many generation tasks, such as machine translation, dialog system and image caption, with the requirement of a large amount of parallel data. However, it is hard to get parallel data for tasks such as style transfer. For instance, there is only a small number of academic news reports which have corresponding papers. Therefore, we need algorithms that perform style transfer without parallel data.

Another major challenge of style transfer is to separate style from the content. In computer vision, Li et al. (2017) proposes an expression to distinguish style and content of a picture. However, this is under-explored in the NLP community. How to separate content from style in text remains an open research problem in text style transfer.

Evaluation is also a key challenge in style transfer. In machine translation and summarization, researchers use BLEU (Papineni et al. 2002) and ROUGE (Lin 2004) to compute the similarity between model outputs and the ground truth. However, we lack parallel data for style transfer to provide ground truth references for evaluation. The same problem also exists in style transfer in computer vision. To solve this problem, we propose a general evaluation metric for style transfer in natural language processing. There are two aspects of the evaluation metric; one is *transfer strength* and the other is *content preservation*.

In this paper, we explore two models for text style transfer, to approach the aforementioned problems of 1) lacking parallel training data and 2) hard to separate the style from the content. The models achieve the goals by multi-task learning (Caruana 1998) and adversarial training (Goodfellow et al. 2014) of deep neural network. The first model implements a multi-decoder seq2seq proposed by Sutskever, Vinyals, and Le (2014), where the encoder is used to capture the content c of the input X, and the multi-decoder contains $n(n \geq 2)$ decoders to generate outputs in different styles. The second model uses the same encoding strategy, but introduces style embeddings that are jointly trained with the model. The style embeddings are used to augment the encoded representations, so that only one decoder needs to be learned to generate outputs in different styles.

The experiments on two tasks: paper-news title transfer and positive-negative review transfer showed that each of the proposed model has its own strength and can be used in

different transfer requests, and the proposed content preservation metric has a high correlation with human judgments.

Contributions

The contributions of this paper are three fold:

- We compose a dataset[1] of paper-news titles to facilitate the research in language style transfer.
- We propose two general evaluation metrics for style transfer, which considers both transfer strength and content preservation. The evaluation metric is highly correlated to the human evaluation.
- We proposed and evaluated two models for learning style transfer without parallel corpora. The proposed models addressed the key challenge of lacking parallel data for training in style transfer, and each model has its own advantages under different scenarios.

Related Work

Style Transfer in Computer Vision

In recent years, style transfer has made significant progress in computer vision. Gatys, Ecker, and Bethge (2016) separated the content and style of images and recombined them to generate new images. Gatys et al. (2016) designed a simple linear model to change the color of the pictures. Their methods use only one image to represent a style. However, it does not work in NLP because a single sentence or a short article does not store enough style information.

Zhu et al. (2017) proposes CycleGAN to do image-image translation. It firstly learns a mapping $G : X \rightarrow Y$ using an adversarial loss, and then a reverse mapping $F : Y \rightarrow X$ with a cycle loss $F(G(X)) \approx X$ which performs unpaired image to image translation. CycleGAN shows qualitative results, nevertheless, discrete text is hard to implement cycle training. Li et al. (2017) proposes to treat style transfer as a domain adaptation problem. They theoretically show that Gram metrics is equivalent to minimize the Maximum Mean Discrepancy (MMD) for image. But there is no evidence showing similar metric works on text.

Style Transfer in Natural Language Processing

Jhamtani et al. (2017) explores automatic methods to transform text from modern English to Shakespearean English using parallel data. The model was based on seq2seq and enriched it with pointer network (Vinyals, Fortunato, and Jaitly 2015). They used a modern-Shakespeare word dictionary to form candidate words for pointer network, however, paired-word dictionary is a scares resource that does not exist in most style transfer tasks, and it required parallel corpora.

There are previous work on style transfer without parallel data. Mueller, Gifford, and Jaakkola (2017) proposed a variational auto-encoder (VAE) based model to revise a new sequence to improve its associated outcome. However, there is no significative evaluation for style transfer. It uses non-parallel data. Shen et al. (2017) explored style transfer for sentiment modification, decipherment of word substitution ciphers and recovery of word order. They used VAE as the base model and used an adversarial network to align different styles. However, their evaluation only considered the classification accuracy. We argue that content preservation is another indispensable evaluation metric for style transfer.

Other threads of work that are closely related to us including style analysis and style-controlled text generation. Braud and Søgaard (2017) explores many types of features for style prediction, ranging from n-grams to discourse, and found that simple models performed well. Ficler and Goldberg (2017) controls linguistic style of generated text using conditioned recurrent neural networks (CRNN). The major difference between these work and ours is that they do not have source sentences where we need to transfer the style.

Adversarial Networks for Domain Separation

Adversarial networks have been successfully applied to domain separation problems. (Ganin and Lempitsky 2015) proposed deep domain adaptation approach to encourage domain-invariant features. This model can be trained on labeled source domain data and unlabeled target domain data. (Bousmalis et al. 2016) used adversarial networks to learned shared representations between two domains which don't contain the individual features of each domain. (Chen et al. 2017) proposed a multi-task framework to generate shared and private representations for sentences. The shared layer is also reinforced by adversarial networks. (Long, Wang, and Jordan 2017) proposed a joint adaptation network, which adopted the adversarial strategy to maximize joint maximum mean discrepancy. The major difference between these work and ours is that they do not need to generate new sentences. How adversarial networks work on controlled generation is largely untested.

Model

We propose two models for style transfer in this paper: multi-decoder and style-embedding. Both models are based on the neural sequence to sequence model. The common ground of the two models is to learn a representation for the input sentence that only contains the content information. Then the multi-decoder model uses different decoders, one for each style, to generate texts in the corresponding style. The style-embedding model, in contrast, learns style embeddings addition to the content representations. Then a single decoder is trained to generate texts in different styles based on both the content representation and the style embedding. Figure 1 illustrates the two models. We give more details about each model in the following sections.

Background: Auto-encoder Seq2seq Model

Auto-encoder (Rumelhart, Hinton, and Williams 1985) is a type of neural networks that learns a hidden representation for the input. It was mainly used for dimension reduction in the past, but more recently, the concepts have been widely used for generative models. In the auto-encoder seq2seq model, an encoder is learned to generate intermediate representation of input sequence $X = (x_1, \ldots, x_{T_x})$ of length T_x. Then a decoder is trained to recover the input X using

[1] Available at https://github.com/fuzhenxin/textstyletransferdata

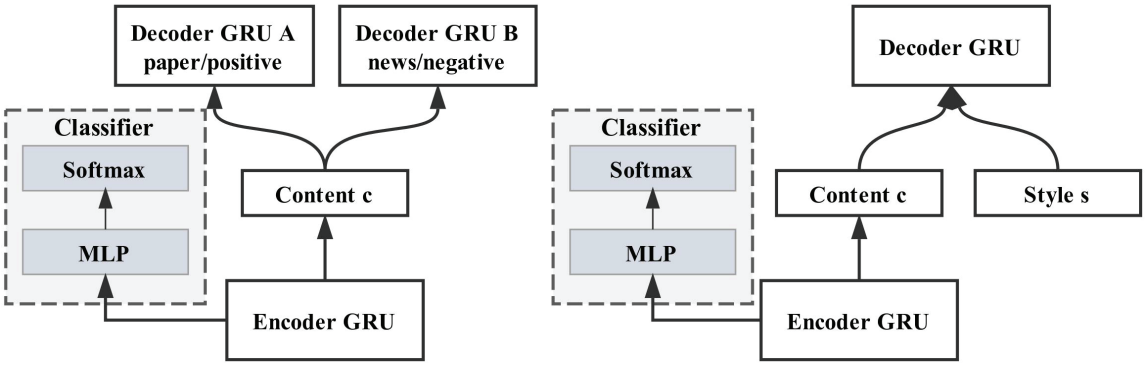

Figure 1: Two models in this paper, multi-decoder (left) and style-embedding (right). Content c represents output of the encoder. Multi-layer Perceptron (MLP) and Softmax constitute the classifier. This classifier aims at distinguishing the style of input X. An adversarial network is used to make sure content c does not have style representation. In style-embedding, content c and style embedding s are concatenated and $[c, e]$ is fed into decoder GRU.

the intermediate representation. For the style transfer problem, we use the auto-encoder seq2seq model as our base model, since we expect minimum changes from the input to the output. We give more details about this model as we also use the components of this model in our proposed models.

Encoder In auto-encoder seq2seq model, both the encoder and decoder are recurrent neural networks (RNNs). We employ the gated recurrent unit (GRU) variant which uses gates to control the information flow. A GRU unit is composed of the following components:

$$\mathbf{s}_j = \mathbf{z}_j \odot \mathbf{h}_j + (1 - \mathbf{z}_j) \odot \mathbf{s}_{j-1}, \quad (1)$$

$$\mathbf{h}_j = \tanh\left(\mathbf{W}\mathbf{E}[x_{j-1}] + \mathbf{r}_j \odot (\mathbf{U}\mathbf{s}_{j-1})\right), \quad (2)$$

$$\mathbf{r}_j = \sigma\left(\mathbf{W}_r\mathbf{E}[x_{j-1}] + \mathbf{U}_r\mathbf{s}_{j-1}\right), \quad (3)$$

$$\mathbf{z}_j = \sigma\left(\mathbf{W}_z\mathbf{E}[x_{j-1}] + \mathbf{U}_z\mathbf{s}_{j-1}\right), \quad (4)$$

where $\mathbf{s}_j$ is the activation of GRU at time j; $\mathbf{h}_j$ is an intermediate state computes the candidate activation. r_j is a reset gate that controls how much to reset from the previous activation for the candidate activation. Similarly, z_j is an update gate that controls how much to update the current activation based on the previous activation and the candidate activation. $\mathbf{E}$ is a word embedding matrix that is used to convert the input words to vector representations. $\mathbf{E}$, $\mathbf{W}$, $\mathbf{U}$, $\mathbf{W}_r$, $\mathbf{U}_r$, $\mathbf{W}_z$, $\mathbf{U}_z$ are model parameters. We use Θ_e to denote all the parameters of the encoder, then the encoder can be abstracted as:

$$\mathbf{S} = Encoder(\mathbf{x}; \Theta_e) \quad (5)$$

Decoder The decoder takes the last state of the encoder to start the generation process. It generates tokens by predicting the most probable next token based on previous tokens. The probability of an output sequence given an input $P(\mathbf{y}_i|\mathbf{x}_i)$ is defined by Equation 6, where i indexes the instances, j the output tokens. The probability $p(.)$ of generating each token can be computed by the softmax function.

$$P\left(\mathbf{y}_i|\mathbf{x}_i; \Theta_d\right) = \prod_{j=1}^{T_y} p\left(y_{i,j}|Encoder(\mathbf{x}_i; \Theta_e), y_{i,1}, \ldots, y_{i,j-1}; \Theta_d\right) \quad (6)$$

The loss function of the encoder-decoder seq2seq model (Equation 7) minimizes the negative log probability of the training data, where M denotes the size of the training data, Θ_e and Θ_d are the parameters of the encoder and the decoder, respectively. The model can be trained end-to-end.

$$L_{seq2seq}\left(\Theta_e, \Theta_d\right) = -\sum_{i=1}^{M} \log\ P\left(\mathbf{y}_i|\mathbf{x}_i; \Theta_e, \Theta_d\right) \quad (7)$$

In auto-encoder, we let the output sequence $\mathbf{y}$ to be the same as the input sequence $\mathbf{x}$.

Multi-decoder Model

The multi-decoder model for style transfer is similar to an auto-encoder with several decoders, with the exception that the encoder now tries to learn some content representations that do not reflect styles. The style specific decoders (one for each style) then take the content representations and generate texts in different styles. The challenge of this model is how to generate content representation $\mathbf{c}$ from input $\mathbf{x}$. In the original auto-encoder model, the encoder generates representations that contain both content and style information.

Chen et al. (2017) used an adversarial network to separate the shared and the private features for multi-task learning to help chinese word segmentation. We use a similar adversarial network to separate the content representation $\mathbf{c}$ from the style. The adversarial network is composed of two parts. The first part aims at classifying the style of $\mathbf{x}$ given the representation learned by the encoder. The loss function minimizes the negative log probability of the style labels in the training data, as denoted in Equation 8:

$$L_{adv1}\left(\Theta_c\right) = -\sum_{i=1}^{M} \log\ p\left(l_i|Encoder(\mathbf{x}_i; \Theta_e); \Theta_c\right), \quad (8)$$

where Θ_c is the parameters of a multi-layer perceptron (MLP) for predicting the style labels. The second part of the adversarial network aims at making the classifier unable to identify the style of $\mathbf{x}$ by maximize the entropy (minimize the negative entropy) of the predicted style labels, as denoted

in Equation 9.

$$L_{adv2}\left(\Theta_e\right) = -\sum_{i=1}^{M}\sum_{j=1}^{N} H\left(p\left(j|Encoder(\mathbf{x}_i;\Theta_e);\Theta_c\right)\right), \quad (9)$$

where Θ_e is the parameters of the encoder and N is the number of styles, as introduced in previous sections. Note that the two parts of the adversarial network update different sets of parameters, and they work together to make sure that outputs of encoder $Encoder(\mathbf{x}_i;\Theta_e)$ do not contain style information.

While the encoder is trained to produce content representations, the multiple decoders are trained to take the representations produced by the encoder and generate outputs in different styles. The loss function for each decoder is similar to Equation 7, and the total generation loss is the sum of the generation loss of each decoder, as defined in Equation 10.

$$L_{gen1}\left(\Theta_e,\Theta_d\right) = \sum_{i=1}^{L} L^i_{seq2seq}\left(\Theta_e,\Theta^i_d\right) \quad (10)$$

The final loss function of the multi-decoder model is composed of three parts: two for the adversarial network and one for the sequence to sequence generation. It simply takes an unweighted sum of the three parts as illustrated in Equation 11.

$$\begin{aligned} &L_{total1}\left(\Theta_e,\Theta_d,\Theta_c\right) \\ &= L_{gen1}\left(\Theta_e,\Theta_d\right) + L_{adv1}\left(\Theta_c\right) + L_{adv2}\left(\Theta_e\right) \end{aligned} \quad (11)$$

Style-embedding Model

Our second model uses style embeddings to control the generated styles. This is inspired by (Li et al. 2016), which proposed a model to embed personal information into vector representations for persona-conversation, and (Ficler and Goldberg 2017) which generated text with different contents and styles using conditional RNNs that conditioned on both content and style parameters.

In this model, the encoder and the adversarial network parts are the same as the multi-decoder model, to generate content representations $\mathbf{c}$. In addition, style embeddings $\mathbf{E} \in R^{N\times d_s}$ are introduced to represent the styles, where N denotes the number of styles and d_s is the dimension of style embedding. A single decoder is trained in this model, which takes the concatenation of the content representation $\mathbf{c}$ and the style embedding $\mathbf{e}$ of a sentence as the input to generate texts in different styles.

The loss function of the style-embedding model is defined in (12), where L_{gen2} is the loss function for the seq2seq generation very similar to Equation 7. The only difference is that it also contains the parameter $\mathbf{E}$ for style embeddings, that are jointly trained with the mode. The total loss is similar to the multi-decoder model in Equation 11, where L_{adv1} and L_{adv2} are the same as in Equations 8 and 9.

$$\begin{aligned} &L_{total2}\left(\Theta_e,\Theta_d,\Theta_c,\mathbf{E}\right) \\ &= L_{gen2}\left(\Theta_e,\Theta_d,\mathbf{E}\right) + L_{adv1}\left(\Theta_c\right) + L_{adv2}\left(\Theta_e\right) \end{aligned} \quad (12)$$

Parameter Estimation

We use Adadelta (Zeiler 2012) with the initial learning rate 0.0001 and batch size 128 to learn the parameters for all models. The best parameters are decided based on the perplexity on the validation data with a maximum of 50 training epochs for paper-news task and 10 training epochs for positive-negative task.

For the multi-decoder model, we train the multiple decoders alternately, using the data in the corresponding style. For the style-embedding model, we randomly shuffled the data during training, and jointly learned the style embeddings with the encoder-decoder part.

Evaluation

Evaluation plays an important role in style transfer. Automatic evaluation metrics speed up development. And they provide criteria to compare different models.

BLEU (Papineni et al. 2002) is a popular evaluation metric in neural machine translation and ROUGE (Lin 2004) is popular in text summarization. NIST (Doddington et al. 2000) and Meteor (Banerjee and Lavie 2005) are also used widely in Natural Language Processing. They evaluate the similarity between model output and ground truth by word overlapping. AM-FM (Banchs and Li 2011) proposes an automatic evaluation for NMT without ground truth. This model computes sentence embedding first and then computes cosine similarity between source language input and target language output. It gets sentence embedding by Singular Value Decomposition (SVD), which trains source and target language together. RUBER (Tao et al. 2017) was proposed to evaluate dialog system, it divides evaluation into referenced and unreferenced part. In referenced part, it calculates the similarity between model output and ground truth by cosine distance of sentence embedding.

We propose two general evaluation metrics, one is transfer strength, the other one is content preservation.

Transfer Strength

The main task of this model is to transfer source style to target style, so transfer strength evaluates whether the style is transferred. We define the metric as transfer strength and implement it using a classifier. There are more than 100,000 training data for this task. We use a LSTM-sigmoid classifier which performs well in big data. The style is defined in (13). This classifier is based on keras examples[2]. Transfer strength accuracy is defined as $\frac{N_{right}}{N_{total}}$, N_{total} is the number of test data, and N_{right} is the number of correct case which is transferred to target style.

$$l_{style} = \begin{cases} paper(positive) & output \leq 0.5 \\ news(negative) & output > 0.5 \end{cases} \quad (13)$$

For similar task, (Shen et al. 2017) uses classifier to evaluate style transfer. (Zhou et al. 2017) controls emotion of conversation, it also uses a classifier to evaluate chatbot generated emotional response.

Content Preservation

Another important aspect of style transfer is content preservation. It is easy to train a model that has 100% transfer

[2]https://github.com/fchollet/keras/blob/master/examples/imdb_lstm.py

dataset	title		review	
style type	paper	news	positive	negative
#sentences	107,538	108,503	400,000	400,000
vocabulary size	80,000		60,000	

Table 1: Size of datasets

strength by only generating the target style words. Therefore, we propose a metric for content preservation, which can evaluate the similarity between source text and target text. Content preservation rate is defined as cosine distance (18) between source sentence embedding v_s and target sentence embedding v_t. Sentence embedding consists of max,min,mean pooling of word embedding defined in (17).

$$v_{min}[i] = \min\{w_1[i], \dots w_n[i]\} \quad (14)$$

$$v_{mean}[i] = \text{mean}\{w_1[i], \dots w_n[i]\} \quad (15)$$

$$v_{max}[i] = \max\{w_1[i], \dots w_n[i]\} \quad (16)$$

$$v = [v_{min}, v_{mean}, v_{max}] \quad (17)$$

$$score = \frac{v_s^\top v_t}{\|v_s\| \cdot \|v_t\|} \quad (18)$$

$$score_{total} = \sum_{i=1}^{M_{test}} score_i \quad (19)$$

For word embedding, we use pre-trained Glove (Pennington, Socher, and Manning 2014) published at stanford nlp[3]. This project contains word embedding trained on 6 billion tokens, containing 400k vocabularies, with dimension 50, 100, 200 and 300. In our model, we use dimension 100.

Although a single integrated metric that combines transfer strength and content preservation as F1 score seems plausible to measure the performance of the systems, it is not the best for style transfer, since sometimes the transfer strength is more important, while in other cases the content preservation is the focus. A weighted integration would be ideal for different scenarios. We leave the weighted integration for the future work and report both metrics in this paper.

Experimental Setup

Datasets

We used two datasets to evaluate the performances of the proposed methods. One is the paper-news title dataset, the other is the positive-negative review dataset; both are non-parallel corpora. We composed the first dataset ourselves and used the data released by He and McAuley (2016) as the second dataset. For both datasets, we divided them into three parts: training, validation, and test data. The size of the validation and test data is 2,000 sentences, and the rest are used as training data. And the partition is the same between model and evaluation.

We ignored the sentences that contain more than 20 words, and converted all characters to lower cases. We also replace all the numbers to a special string "⟨NUM⟩" as a preprocessing step. Some statistics about the datasets is summarized in Table 1.

[3]https://nlp.stanford.edu/projects/glove/

Paper-News Title Dataset In this dataset, the paper titles are crawled from academic websites including ACM Digital Library[4], Arxiv[5], Springer[6], ScienceDirect[7], and Nature[8]. The news titles are from UC Irvine Machine Learning Repository (Lichman 2013), which contains 422,937 news titles. We filtered it down to 108,503 titles which belong to science and technology category.

Positive-Negative Review Dataset This dataset contains Amazon product reviews published by He and McAuley (2016). It contains 142,800,000 product reviews from 1996 to 2014 in Amazon, which span the domains of books, electronics, movies, etc. We randomly select 400,000 positive and 400,000 negative reviews to compose our dataset.

Model Settings

Since this is an exploratory paper, we compare several parameters settings instead of trying to find a single set of "best parameters". For paper-news title transfer, we explored word embedding size of 64, encoder hidden vector size among {32,64,128}, and style embedding size among {32,64,128}. For positive-negative review transfer, we explored word embedding size of 64 for multi-decoder and {64,128} for style-embedding model, encoder hidden vector size among {16,32,64}, and style embedding size among {16,32,64}.

Evaluation Settings

As is introduced in previous sections, an LSTM-sigmoid classifier is needed to measure the transfer strength. We train an LSTM with the input word embedding dimension and hidden state dimension both be 128. On the paper-news title transfer dataset, the training stops after two epochs, and the accuracy on the validation data is 98.8%. For the positive-negative review dataset, the training also stops after two epochs, with an accuracy of 84.8% on validation.

For the content preservation metric, we use pretrained 100-dimensional word embeddings to compute sentence similarities. For the positive and negative review transfer task, we filter out the sentiment words to make sure the content preservation metric indeed measures the content similarity. A positive and negative word dictionary is used to conduct the filtering.

Results and Analysis

As we discussed in previous sections, this paper is exploratory. We are exploring whether we can learn style transfer with non-parallel data, and whether we can define some evaluation metrics to measure how well the models do in text style transfer. Therefore, we first examine how does the proposed evaluation metrics compare to human judgments.

[4]http://dl.acm.org
[5]https://arxiv.org
[6]https://link.springer.com
[7]http://www.sciencedirect.com
[8]https://www.nature.com

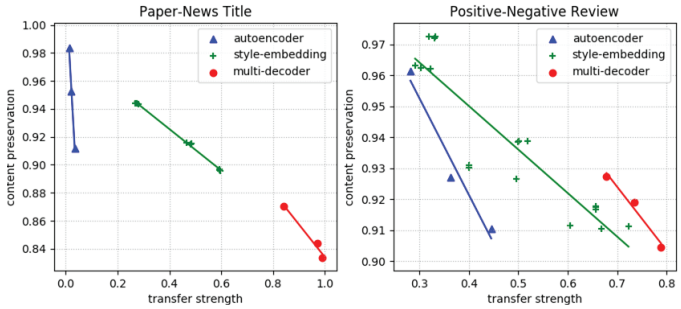

Figure 2: Results for auto-encoder, multi-decoder and style embedding for two tasks, paper-news title style transfer (left) and positive-negative review style transfer (right). Different nodes for the same model denote different hyper-parameters.

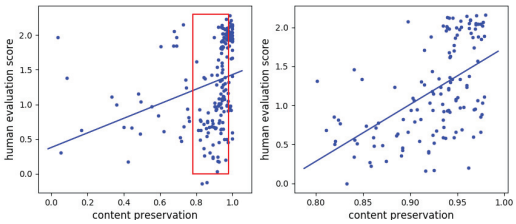

Figure 3: Score correlation of content preservation and human evaluation. Gaussian noise is added to human evaluation for better visualization. The partial enlarged graph is shown on the right.

Comparison with Human Judgments

To ensure our proposed content preservation metric is efficient in measuring the sentence similarities, we compare it against human judgments. The human judgments are obtained by randomly sampling 200 paper-news transferred pair from the test data, target transferred sentences are generated by style-embedding model, and ask three different people to rate the pairs with scores {0, 1, 2}. 2 means the two sentences are very similar; 1 means the two sentences are somewhat similar; and 0 indicates the two sentences are not similar. We conduct this experiment on Amazon Mechanical Turk[9]. The scores for each pair from different people are averaged to generate the final human judgment scores. We then calculate the Spearman's coefficient (a measurement for accessing monotonic relationships) between the human judgment scores and our content preservation metric. The correlation score is 0.5656 with p-value$<$0.0001, indicates a high correlation between human judgment scores and the content preservation metric. Figure 3 illustrates the correlation.

Model Performances

We then explore the effect of different parameters on different models for style transfer. Figure 2 gives an overview of the results. We can see that in both tasks and all the models, transfer strength and content preservation are negatively correlated. This indicates that within the same model, to get more style changes, one has to lose some contents. We also see the slopes of the trade-off curves appear to be less steep in our proposed models than in the auto-encoder, which indicates our models strike a better balance between the two aspects (transfer strength and content preservation) of style transfer. We now give detailed analysis of the performances of different models with different parameters on the two tasks, respectively. More details about the influences of the hyper-parameters can be found at https://arxiv.org/abs/1711.06861.

Paper-News Title Transfer For the paper-news title transfer task, the auto-encoder is able to recover most of the content, but with few transfer strength, just as we expected. The multi-decoder performs better on transfer strength, while style-embedding performs better on content preservation. Both are also able to achieve considerably high scores in two metrics, so there is no clear winning model.

More specifically, for the style-embedding model, the transfer strength ranges from 0.2 to 0.6 when using different hyper-parameters, and the content preservation ranges from 0.89 to 0.95. Both cover a wide range and would be useful for certain downstream tasks. For the multi-decoder model, it generally tends to generate results with high transfer strength but low content preservation. Therefore, we suggest using multi-decoder and style-embedding in different request scenarios.

Positive-Negative Review Transfer For the positive-negative review transfer task, the transfer strength of auto-encoder is no longer nearly zero like in the paper-news title transfer task, probably because the classifier used to measure the transfer strength is not perfect[10]. The transfer strength measure is not as reliable as it is in the paper-news title task.

For the style-embedding model, it covers a quite wide range in both transfer strength and content preservation. The multi-decoder model still shows high transfer strength as is in the paper-news title transfer task, and it achieved higher content preservation than that in paper-news title transfer. In

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

[10]The accuracy of this classifier is only 84.8% on the validation data, probably because some sentences in the positive-negative review dataset do not have significant sentiment

source	positive: all came well sharpened and ready to go .
auto-encoder:	→negative: all came well sharpened and ready to go .
multi-decoder:	→negative: all came around , they did not work .
style-embedding:	→negative: my ⟨NUM⟩ and still never cut down it .
source	negative: my husband said it was obvious so i had to return it .
auto-encoder:	→positive: my husband said it was obvious so i had to return it .
multi-decoder:	→positive: my husband was no problems with this because i had to use .
style-embedding:	→positive: my husband said it was not damaged from i would pass right .
source	paper: an efficient and integrated algorithm for video enhancement in challenging lighting conditions
auto-encoder:	→news: an efficient and integrated algorithm for video enhancement in challenging lighting conditions
multi-decoder:	→news: an efficient and integrated and google smartphone for conflict roku together wrong
style-embedding:	→news: an efficient and integrated algorithm, for video enhancement in challenging power worldwide
source	news: luxury fashion takes on fitness technology
auto-encoder:	→paper: luxury fashion takes on fitness technology
multi-decoder:	→paper: foreign banking carbon on fitness technology
style-embedding:	→paper: luxury fashion algorithms on fitness technology

Table 2: Case study of style transfer

this dataset, the multi-decoder model performs better than the style-embedding model on both metrics (the red line is on the upper right over the green line).

Analysis in a Multi-task Learning View

Auto-encoder, style-embedding and multi-decoder can be seen as different strength implement of multi-task learning. In our model, generating different titles can be seen as different tasks. For some kind of multi-task learning, different tasks share parameters to share features in different tasks.

For auto-encoder, two tasks share all the parameters, so it does not have the ability to generate different style sequence. For style-embedding, two tasks share encoder and decoder with separate style embedding, so it has weak ability to generate different style sequence. For multi-decoder, two tasks share encoder with two separate decoders, so it shows high ability to generate different style sequence. For content preservation, more parameters are shared, less distinction between two tasks and more content is preserved. Since the style-embedding model shares more parameters among tasks, less training data is needed to train the model, but the style embeddings have heavier burden to encode the style information.

Lower Bound for Content Preservation

We also estimate the lower bound of the content preservation metric, to gauge how well our model performed in preserving the content. The lower bound is estimated by randomly sampling 2,000 sentence pairs from the two datasets, respectively. Results show that the estimated lower bound of content preservation on the paper-news title dataset is 0.609 and 0.863 on the positive-negative review dataset. For both datasets, our models achieved much higher content preservation scores than the lower bound. This indicates that the proposed model learned to preserve the content of the source sentence well.

Qualitative Study

To give people some intuitive sense about how our models perform, we sampled one instance from each style transfer case, and show the results of three models in Table 2. We can see that the auto-encoder almost always produce the identical output text as the input. The other two models tend to generate results that replace a few significant words or phrases, but preserve most of the content. Both models perform quite well on the positive-negative style transfer, but less well on the paper-news transfer.

Conclusions

We studied the problem of style transfer with non-parallel corpora. We proposed two models and two evaluation metrics to advance the research in this area. We also composed two datasets: paper-news title dataset and positive-negative review dataset, to gauge the efficiency of the proposed models and evaluation metrics. Experiments showed that the proposed models can be used to learn style transfer from non-parallel data, and the proposed content preservation evaluation metric is highly correlated to human judgment.

In the future, we plan to propose more comprehensive evaluation metrics (including sentence fluency) and conduct through study with human evaluation, to better shape the research in style transfer.

Acknowledgment

We thank Jin-ge Yao for discussions on this paper. This work was supported by the National Key Research and Development Program of China (No. 2017YFC0804001), the National Science Foundation of China (No. 71672058), and Contract W911NF-15-1-0543 with the US Defense Advanced Research Projects Agency (DARPA). Rui Yan was sponsored by the CCF-Tencent Open Research Fund.

References

Banchs, R. E., and Li, H. 2011. Am-fm: a semantic framework for translation quality assessment. In *Proceedings*

of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2, 153–158. Association for Computational Linguistics.

Banerjee, S., and Lavie, A. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, volume 29, 65–72.

Bousmalis, K.; Trigeorgis, G.; Silberman, N.; Krishnan, D.; and Erhan, D. 2016. Domain separation networks. In *Advances in Neural Information Processing Systems*, 343–351.

Braud, C., and Søgaard, A. 2017. Is writing style predictive of scientific fraud? *arXiv preprint arXiv:1707.04095*.

Caruana, R. 1998. Multitask learning. In *Learning to learn*. Springer. 95–133.

Chen, X.; Shi, Z.; Qiu, X.; and Huang, X. 2017. Adversarial multi-criteria learning for chinese word segmentation. In *ACL*.

Doddington, G. R.; Przybocki, M. A.; Martin, A. F.; and Reynolds, D. A. 2000. The nist speaker recognition evaluation–overview, methodology, systems, results, perspective. *Speech Communication* 31(2):225–254.

Ficler, J., and Goldberg, Y. 2017. Controlling linguistic style aspects in neural language generation. *arXiv preprint arXiv:1707.02633*.

Ganin, Y., and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, 1180–1189.

Gatys, L. A.; Bethge, M.; Hertzmann, A.; and Shechtman, E. 2016. Preserving color in neural artistic style transfer. *arXiv preprint arXiv:1606.05897*.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2414–2423.

Ghazvininejad, M.; Shi, X.; Choi, Y.; and Knight, K. 2016. Generating topical poetry. In *EMNLP*, 1183–1191.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

He, R., and McAuley, J. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, 507–517. International World Wide Web Conferences Steering Committee.

Jhamtani, H.; Gangal, V.; Hovy, E.; and Nyberg, E. 2017. Shakespearizing modern language using copy-enriched sequence-to-sequence models. *arXiv preprint arXiv:1707.01161*.

Li, J.; Galley, M.; Brockett, C.; Spithourakis, G. P.; Gao, J.; and Dolan, B. 2016. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.

Li, Y.; Wang, N.; Liu, J.; and Hou, X. 2017. Demystifying neural style transfer. In *IJCAI*.

Lichman, M. 2013. UCI machine learning repository.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain.

Long, M.; Wang, J.; and Jordan, M. I. 2017. Deep transfer learning with joint adaptation networks. In *ICML*.

Mueller, J.; Gifford, D.; and Jaakkola, T. 2017. Sequence to better sequence: continuous revision of combinatorial structures. In *International Conference on Machine Learning*, 2536–2544.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 311–318. Association for Computational Linguistics.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, 1532–1543.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1985. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Shen, T.; Lei, T.; Barzilay, R.; and Jaakkola, T. 2017. Style transfer from non-parallel text by cross-alignment. *arXiv preprint arXiv:1705.09655*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Tao, C.; Mou, L.; Zhao, D.; and Yan, R. 2017. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. *arXiv preprint arXiv:1701.03079*.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, 2692–2700.

Yan, R.; Jiang, H.; Lapata, M.; Lin, S.-D.; Lv, X.; and Li, X. 2013. i, poet: Automatic chinese poetry composition through a generative summarization framework under constrained optimization. In *IJCAI*, 2197–2203.

Yan, R. 2016. i, poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema. In *IJCAI*, 2238–2244.

Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhou, H.; Huang, M.; Zhang, T.; Zhu, X.; and Liu, B. 2017. Emotional chatting machine: Emotional conversation generation with internal and external memory. *arXiv preprint arXiv:1704.01074*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*.