Text Style Transfer via Back Inference with Pseudo-Parallel Data

Mengzhou Xia (mengzhox) [1] **Xuanyue Yang (xuanyuey)** [1] **Jiacheng Zhu (jzhu4)** [1]

Abstract

Text style transfer and text manipulation in natural language generation requires an explicit disentanglement of text style and content. However, despite the success of deep generative models in computer vision, this problem remains challenging due to the lack of parallel labeled data and also the complexity of natural language. In this project, we propose a new generative model based on Auto-encoder with a back-inference regularization. We utilize language models to guide the generators by aligning the generated distributions to those of the corresponding styles, and the encoder is forced to align the back-inferred latent representations of the pseudo-parallel examples via adversarial training. Our proposed method is able to achieve comparable results on all of the evaluation metrics, and by adding back-inference phase with pseudo-parallel data, we are able to gain further improvements in terms of BLEU scores and transfer accuracy.

1. Introduction

Recent years have witnessed a blooming development of deep generative models (Hu et al., 2017b), such as Variational Autoencoders (VAEs) (Kingma & Welling, 2013), and Generative Adversarial Nets (GANs) (Goodfellow et al., 2014). These achievements encourage people to reveal the rich, hierarchical representations over data so that artificial intelligence systems can generate novel and desired contents. Remarkable progress are made in visual domain such as image generation (Radford et al., 2015), image editing (Zhu et al., 2016), and interpretable image representation learning (Chen et al., 2016).

Natural language generation (NLG) is one of the most important and challenging tasks, the success of which heavily relies on deep generative models. Despite the progress in task-specific applications such as machine translation(Bahdanau et al., 2014) and dialogue systems (Wen et al., 2017). Text style transfer, which requires learning and manipulating controllable disentangled representations of attributes (e.g., sentiment) and underlying contents, is among one of them. The goal of text style transfer is to render a sentence to be consistent with a preferred style constraint, while at the same time preserve the style-invariant content. Style may refer to a range of linguistic phenomena, including sentiment manipulation, syntactic simplification, and word substitution(Jin et al., 2019).

The difficulties of text transfer lies in the following facts: 1) The semantic structures underling natural language sentences are complex and it's believed that the attributes interact in subtle ways. Thus, how to explicitly separate content from style in text, and measure the disentanglement remains an open problem. 2) For text generation, it is typically unsupervised due to a lack of parallel corpus with specific attributes for training (Lample et al., 2019). Only non-parallel examples with various attribute values are available for training.

There are multiple works concentrating on mitigating the above-mentioned problem. Thanks to their functionality of teasing apart content and style information in the latent space, VAE-based models (Hu et al., 2017a; Fu et al., 2018; John et al., 2018; Kim et al., 2017) and GAN-based models (Shen et al., 2017; Yang et al., 2018) are utilized in a lot of recently proposed approaches for text style transfer and disentanglement. However, most of the existing approaches either remain to be improved or lack explicit enforcement of the disentangled property on the latent representation. For example, The adversarial training proposed by Hu et al. (2017a) enhances the generator with back inference but neglects the encoding process. Nevertheless, it is shown in (Lample et al., 2019) that adversarial training (Shen et al., 2017; Fu et al., 2018; Yang et al., 2018) does not achieve ideal results because the post-fit discriminator is still able to recover the sentiment from encoder's representation.

In this project, we propose an unsupervised style transfer approach which explicitly disentangle styles and contents to address the above limitations. Inspired by Hu et al. (2017a), Kim & Mnih (2018) and Arjovsky et al. (2017), we base our method on Auto-encoders with a back-inference regularization to improve the disentanglement capability of the encoder. The generator is pushed to generate contents

[1]Carnegie Mellon University, Pittsburgh, PA 15213, USA.

10708 Class Project,, Spring, 2019.

with more desirable style by aligning generated distribution with a style-related language model. Meanwhile the encoder is forced to align the back-inferred latent representations from generated sentences with different styles via adversarial training. The back inference procedure with the pseudo-parallel sentence pairs serves as an explicit means of enforcing disentanglement for the encoder.

2. Related Work

Auto-Encoders for Text Style Transfer Past work on text style transfer focuses on separating the representations for content and style under the auto-encoder framework, and conducting the transfer via keeping the content and switching the style. Based on VAE, Fu et al. (2018) enforce the disentanglement in the latent representation by adopting an adversarial style discriminator to train a content-specific style-irrelevant text encoder. Kim et al. (2017) apply the similar adversarial loss for their ARAE. Further John et al. (2018) apply the adversary on both the content and style representation to mutually discourage each to contain the information of the other. Without such adversary on the representations, Hu et al. (2017a) perform disentanglement by guiding the generated samples back to the latent style and content representations, through a style discriminator and the encoder itself respectively. We will discuss our relation to their work later. Instead of explicitly disentangling the content and style in the latent representation, Shen et al. (2017) follow a GAN schema to generate samples based on one style close to another style, with a style discriminator. Yang et al. (2018) extend the discriminator to a language model to provide more sophisticated train signal than that of a simple binary discriminator.

Alignment Metrics for Distributions From the perspective of generative models, many style transfer methods are based on the effort to align certain distributions. The original reconstruction loss for GAN (Goodfellow et al., 2014) and VAE (Kingma & Welling, 2013) amounts to the KL divergence (KLD below) to align the data distribution and the generator distribution. In light of an adversarial discriminator's ability to align distributions, AAE (Makhzani et al., 2015) replace the KLD term to a discriminator to the latent posterior and the prior, which also support arbitrary priors than KLD-computable ones. Arjovsky et al. (2017) propose Wasserstein GAN, using Earth Mover (Wasserstein-1) distance as a smoother metric for aligning distributions where other metrics such as KLD or JSD would fail due to intractability or triviality. Similarly Tolstikhin et al. (2017) apply the Wasserstein distance to VAE reconstruction objective and show that it is a generalization of AAE. Kim et al. (2017) apply the Wasserstein distance on the AE latent space and allow an learnable latent prior rather than a simple standard gaussian. (Shen et al., 2017; Yang et al., 2018) essentially follow an AAE setting to align distributions.

Disentanglement Metrics With the presence of a ground truth simulator and factors, Higgins et al. (2017); Kim & Mnih (2018) propose disentangle metrics based on generating samples with an specific fixed factor and examine the latent representations inferred back by the decoder whether they have a low variance on the values for that factor. Inspired by such property of sound disentanglement, we proposed exerting a constraint on back-inferred latent representations from generated samples of different styles to improve disentanglement. We also formulate a similar metric with a small set of parallel texts.

Continuous Approximation for Discrete Generation The discrete nature of language pose a threat to conduct overall optimization of sequence generation due to non-differentiable sampling operations. He et al. (2016) adopt the REINFORCE method as used for policy gradient (Sutton et al., 2000) to back-propagate through discrete sampling, which however suffers high variance and instability issues. The Gumbel-softmax re-parameterization trick (Jang et al., 2016) is utilized as either a approximation for discrete one-hot samples (Hu et al., 2017a; Yang et al., 2018) or a better gradient estimator with a straight-through setting (Gu et al., 2018), both facilitating a holistically optimization for generative model incorporated with sampling steps.

3. Proposed Method

3.1. VAE-based Models

Variational auto-encoder (Kingma & Welling, 2013), as a base model for text generation, provides a theoretical framework by intractably optimizing over the lower bound of the likelihood of input data. Besides explicitly modeling the distribution of latent representation, Hu et al. (2017a) imposes additional independent constraints over the learned latent space by incorporating two losses. The loss of the discriminator pushes the model to generate coherent contents with the corresponding attributes while the loss of the latent representation requires the contents of the generated examples to be preserved. With each part functioning separately, the model disentangles contents with styles. However, though adding independent constraints over a vanilla VAE model helps enhance the disentanglement between styles and contents, the intrinsic inadequacy of imposing a simple gaussian prior over the latent representations refrains z from carrying more information for generation under the unsupervised setting.

3.2. AE-based methods

As a variation of adversarial auto-encoders, Shen et al. (2017) extends from aligning posterior distribution of $\mathbf{z}$ from different styles (Makhzani et al., 2015) to aligning

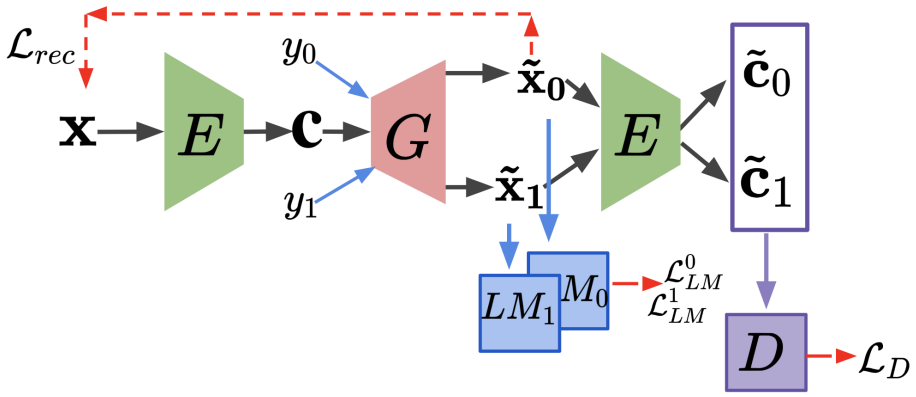

Figure 1. The Diagram of our model. In which the back-inference is illustrated as sending generated sentences to the encoder again.

hidden states of transferred samples from one style to true samples from the other style. The method pushes the complexity of aligning $p(z|x_1)$ and $(z|x_2)$ to the decoder side without explicitly modeling the latent variable z. On top of that, it is argued that the signal provided by a binary discriminator which is used to distinguish whether a sentence is real or fake is not strong enough Yang et al. (2018). Instead, they use language models with token-level locally normalized probabilities as a more direct training signal for the generator. With language models being a structured discriminator, it's shown that training them with only positive examples is sufficient even without adversarial training. However, though both methods claim that they intend to learn a disentangled representation for different attributes, as in the case of style transfer known as content and style, they do not manage to do it successfully (shown in our preliminary experiments). Both methods neither explicitly model the distribution of the latent representation nor truly disentangling contents from styles.

3.3. Back Inference with Pseudo-Parallel Data

As shown in Figure 1, our proposed model aims at satisfies the following conditions: 1) The latent representation z is flexible as opposed to a simple prior e.g., Gaussian distribution, but it's still regularized; 2) The content representations are style-free, leading to an explicit disentanglement of contents and styles. We show our algorithm in 1.

3.3.1. Auto-encoder for Text Generation

We base our model architecture on a auto-encoding generation process. An encoder E_θ encodes the original sentence $\mathbf{x}$ to a content representation $\mathbf{c}$, and thereafter the generator G_ϕ generates either the reconstructed and transferred sentence conditioned on the content representation $\mathbf{c}$ and style embedding $\mathbf{y}$:

$$\mathbf{c} = E_\theta(\mathbf{x}),\ \tilde{\mathbf{x}} = G_\phi(\mathbf{c}, \mathbf{y}). \tag{1}$$

3.3.2. Language Model Discriminator

Under the unsupervised setting, we don't have parallel data, or a true world generator to guarantee the quality of the generated pair sentences, a simple reconstruction penalty from the AE model is not sufficient to provide reliable signals to enhance disentanglement. Therefore, we also want to explicitly align the output distribution of our generator and a ground truth distribution for a specific style. Language models trained with data from a specific style serve as natural ground truth distributions (Yang et al., 2018). As indicated in the paper, the adversarial training over the language model is unnecessary due to its well-defined structure, therefore, we do not include negative examples in training language models. We train the language model of each style with real examples:

$$L^{\mathbf{x}_1}_{\text{LM}}(\mathbf{x}_1) = \mathbb{E}_{\mathbf{x_1}\sim\mathbf{X}_1}[-\log P_{\text{LM}\mathbf{x_1}}(\mathbf{x_1})], \tag{2}$$
$$L^{\mathbf{x}_2}_{\text{LM}}(\mathbf{x}_2) = \mathbb{E}_{\mathbf{x_2}\sim\mathbf{X_2}}[-\log P_{\text{LM}\mathbf{x_2}}(\mathbf{x_2})]. \tag{3}$$

And we use the perplexity of generated examples from the language model to guide the training of the encoder and the generator:

$$L(E_\phi, G_\theta) = \mathbb{E}_{\tilde{\mathbf{x}}_1\sim\tilde{\mathbf{X}}_1}[-\log P_{LM\tilde{\mathbf{x}}_1}(\tilde{\mathbf{x}}_1)] \tag{4}$$
$$+ \mathbb{E}_{\tilde{\mathbf{x}}_2\sim\tilde{\mathbf{X}}_2}[-\log P_{LM\tilde{\mathbf{x}}_2}(\tilde{\mathbf{x}}_2)] + L_{\text{rec}}, \tag{5}$$

where L_{rec} is the reconstruction loss.

3.3.3. Back Inference as Regularization

As a regular auto-encoder does not impose any explicit regularization over the learned latent representation, AE-based models for style transfer introduce other ways in order to learn a structured latent space. However, as shown in the previous section, these methods do not explicitly disentangle the content representations from the styles. Instead, we intend to regularize the latent space meanwhile explicitly enabling disentanglement for a better transfer. In order to achieve this goal, we introduce a back-inference procedure

which aligns the representations of the generated samples from different styles. From the generator's point of view, it is encourage to generate sentence pair with the similar content measured by the closeness the back-inferred content representation. Meanwhile the sentence pair could serve as pseudo-parallel examples for the encoder.

Given two styles $\mathbf{y}_1$ and $\mathbf{y}_2$, and a content representation $\mathbf{c}$, we use our generator to output samples with the same content but different styles, denoted as $\tilde{\mathbf{x}}_1$ and $\tilde{\mathbf{x}}_2$ as follows:

$$\tilde{\mathbf{x}}_1 = G_\phi(\mathbf{c}, \mathbf{y}_1),\ \tilde{\mathbf{x}}_2 = G_\phi(\mathbf{c}, \mathbf{y}_2), \tag{6}$$

where G_ϕ denotes the generator, and $\mathbf{y}_1$ and $\mathbf{y}_2$ are learned style embeddings. Then we infer these generated samples back to the corresponding latent representation via the encoder:

$$\tilde{\mathbf{c}}_1 = E_\theta(\mathbf{x}_1),\ \tilde{\mathbf{c}}_2 = E_\theta(\mathbf{x}_2) \tag{7}$$

where E_θ denotes the encoder. 2

To achieve disentanglement of contents and styles, we align the distribution of representations of original styles and transferred styles via a parameterized discriminator f_γ with adversarial training. Specifically, our training uses the following optimization over the encoder E_θ, the generator G_ϕ and the discriminator D_γ:

$$\begin{cases} \min_\gamma -\mathbb{E}_{\tilde{\mathbf{c}}_{\text{ori}}}[\log D_\gamma(\tilde{\mathbf{c}}_{\text{ori}})] - \mathbb{E}_{\tilde{\mathbf{c}}_{\text{tsf}}}[\log(1 - D_\gamma(\tilde{\mathbf{c}}_{\text{tsf}}))] \\ \max_{\theta,\phi} \mathbb{H}[D_\gamma(\tilde{\mathbf{c}})]. \end{cases} \tag{8}$$

where $\tilde{\mathbf{c}}_{\text{ori}}$ and $\tilde{\mathbf{c}}_{\text{ori}}$ are latent representation from original style and transferred style.

The method to include the back inference procedure with generated samples is inspired by Hu et al. (2017a), Higgins et al. (2017) and (Kim & Mnih, 2018). The VAE structure employed by Hu et al. (2017a) imposes a Gaussian prior over the hidden representation and only update the generator with the back inference loop, which pushes the complexity of disentanglement to the generator. Our method, in contrast, amortizes the burden of disentanglement onto not only the generator, but also the encoder, which we consider is more important for learning disentangled representations. Higgins et al. (2017) and Hu et al. (2017a) propose metrics to evaluate disentanglement by also using the back inference loop. However, the calculation of both metrics requires a true world generator while we directly use our learned generator as a substitution.

Algorithm 1 Disentangling via Back Inference with Pseudo-Parallel Data on Text Transfer

Input: Two monolingual corpora of two styles $\mathbf{X}_1$, $\mathbf{X}_2$
Initialize E, G, D, LM_1 and LM_2
repeat
  for $p = 1, 2; q = 2, 1$ **do**
    Sample a batch $\{\mathbf{x}_p^{(i)}\}_{i=1}^k$ from $\mathbf{X}_p$
    Get latent content representations $\mathbf{c}_p^{(i)} = E_\phi(\mathbf{x}_p^{(i)})$
    Generate $\tilde{\mathbf{x}}_p^{(i)}$, $\tilde{\mathbf{x}}_q^{(i)}$ from G with initialization $(\mathbf{c}_p^{(i)}, y_p)$, $(\mathbf{c}_p^{(i)}, y_q)$ respectively
    Get latent content representations $\tilde{\mathbf{c}}_p^{(i)} = E_\phi(\tilde{\mathbf{x}}_p^{(i)})$ and $\tilde{\mathbf{c}}_q^{(i)} = E_\phi(\tilde{\mathbf{x}}_q^{(i)})$ respectively
  end for
  Compute discriminator loss $\mathcal{L}_D$ with

$$\mathcal{L}_D = -\mathbb{E}_{\tilde{\mathbf{c}}_{\text{ori}}}[\log D_\gamma(\tilde{\mathbf{c}}_{\text{ori}})] - \mathbb{E}_{\tilde{\mathbf{c}}_{\text{tsf}}}[\log(1 - D_\gamma(\tilde{\mathbf{c}}_{\text{tsf}}))]$$

  Compute discriminator adversarial loss $\mathcal{L}_D$ with

$$\mathcal{L}_D^{\text{adv}} = -\mathbb{H}[D_\gamma(\tilde{\mathbf{c}})]$$

  Compute the reconstruction loss $\mathcal{L}_{\text{rec}}$ with

$$\mathcal{L}_{\text{rec}} = \frac{1}{k}\sum_i [-\log p_G(\mathbf{x}_1^{(i)}|\mathbf{c}_1^{(i)}, \mathbf{y}_1)] + \frac{1}{k}\sum_i [-\log p_G(\mathbf{x}_2^{(i)}|\mathbf{c}_2^{(i)}, \mathbf{y}_2)]$$

  Compute the language model loss $\mathcal{L}_{\text{LM}}$ with

$$\mathcal{L}_{\text{LM}}^{\mathbf{x}} = \frac{1}{k}\sum_i [-\log p_{\text{LM}}(\mathbf{x}^{(i)})]$$

$$\mathcal{L}_{\text{LM}}^{\tilde{\mathbf{x}}} = \frac{1}{k}\sum_i [-\log p_{\text{LM}}(\tilde{\mathbf{x}}^{(i)})]$$

  Update LM with $\mathcal{L}_{\text{LM}}^{\mathbf{x}}$
  Update D with $\mathcal{L}_D$
  Update E, G with $\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{LM}}^{\tilde{\mathbf{x}}} + \mathcal{L}_D^{\text{adv}}$
until Convergence

4. Experiments

4.1. Dataset

The publicly available Yelp Review dataset is utilized in our research, following previous work (Shen et al., 2017; Li et al., 2018). Both of these datasets have sentiment labels for each sentence and will be used for training latent space disentanglement as well as evaluating sentiment transfer. Since some prior works applied constraints such as discarding reviews longer than 15 words and only considering the 10K most frequent words (Lample et al., 2019), several pre-process methods will be used in order to invoke a more interesting and challenging task.

Yelp Reviews: The Yelp restaurant review dataset, originally provided by the Yelp Dataset Challenge[1], contains five

[1]https://www.yelp.com/dataset/challenge

coarse-grained restaurant category labels including Asian, American, Mexican, Bars & Dessert in the associated metadata, has a vocabulary size of approximately 10K. The dataset can be pre-processed by removing reviews which are non-English(Joulin et al., 2016), not about restaurants, neutral in sentiment, and gender unidentifiable. In case there are reviews about a restaurant which has multiple labels, a multi-label fastText classifier can be trained and re-label the dataset be picking the most likely category(Lample et al., 2019). The processed Yelp review dataset provided contains 350K positive sentences and 250 negative sentences. (Shen et al., 2017)

Annotated Parallel Yelp Reviews: This dataset [2] (Li et al., 2018) is a small crowdsourced subset of 1,000 Yelp reviews for evaluation, in which the sentiment are swapped between positive and negative while preserving content by human. Having human reference outputs. The result of style transfer can be evaluated by using automatic evaluation metric such as BLEU (Papineni et al., 2002) to investigate how well the content is preserved.

4.2. Evaluation Metrics

Transfer Accuracy We follow previous work (Fu et al., 2018; Shen et al., 2017; Yang et al., 2018; Kim et al., 2017) to utilize a pre-trained style classifier to give accuracy for the style-transferred texts, as claimed reasonable since it has a good performance on the validation set for style classification task.

Perplexity We use a language model trained on style-related corpus to evaluate the fluency of transferred texts under that style. We give results from more objective n-gram language models and more sophisticated neural language models.

Self-BLEU We evaluate the content preservation based on the "self" BLEU score (Papineni et al., 2002) between original and transferred texts, which to some extend reveals the overlap of content. However it is not perfect by including style words as well.

Parallel-BLEU With the help of a human annotated parallel texts of different styles released by (Li et al., 2018), we are able to compute the BLEU score the the transferred texts and the gold reference, which would be a better metric for content preservation.

4.3. Baselines

In this section, we mainly introduce the baseline models that our infer-back methods can be further trained upon.

Language Model Regularization Since we incorporate part of the design from Yang et al. (2018), specifically adding language models as a means of regularization for decoders to do reconstruction and transfer, we compare our method with it.

Label Input Feed Model This baseline model is adapted based on the previous one but reinforce the memory of the label information during decoding. One major difference is that instead of just using the label information for hidden state initialization, we feed the label information into each step's decoding by concatenating it with the input.

Multi-decoder Model Instead of just having one decoder for both styles, we can also have two decoders, one for style one and the other for style two. In this case, each decoder is enhanced by the corresponding language model without any further cross alignment.

4.4. Settings

We split our training process into three phases, namely pre-training, training, and back-inference phase.

Pre-training Phase In the pre-training phase, we pre-train the language model of two styles, the discriminator and the auto-encoder using existing data of the two styles. Since each component of the three is trained independently, each component can achieve convergence smoothly.

Training Phase In the training phase, we train the model of language model regularization, the input feed model or the multi-decoder model without updating the language models and the discriminator, these two components simply provide signals for generation of different styles.

Back-Inference Phase In the back-inference phase, we train upon an existing checkpoint using the generated pseudo-parallel data. Note that we are not using the generated sentences to do inference, instead we use gumbel-softmax samples of the original style and the transferred style as the input to the encoder.

5. Results and Analysis

In this section, we present results of all the baseline models as well as our proposed one and we conduct analysis over all the results. In the first section, we present main results of baseline models and our proposed method with all the metrics including accuracy, parallel BLEU and perplexity on the manually annotated parallel test set. In the second section, we investigate the results deeper by plotting the curves of metrics over the whole training process. In the third section, we show a thorough analysis of transferred results of different training phases by categorizing the sentences into different classes. In the fourth section, we evaluate post-fit accuracy over latent representation.

[2] https://github.com/lijuncen/Sentiment-and-Style-Transfer

Model	Accuracy	BLEU	PPL
Controllable Text (Hu et al., 2017a)	85%	20.6	232.0
Cross-Alignment AutoEncoder (Shen et al., 2017)	72%	6.8	53.0
Style Embedding (Fu et al., 2018)	18%	16.7	56.1
MultiDecoder (Fu et al., 2018)	52%	11.3	90.1
Language Model Regularization (Yang et al., 2018)	90%	22.3	55.0
Adv-Reg AutoEncoder (Kim et al., 2017)	82%	20.2	52.3
Multi Attribute (Lample et al., 2019)	87%	14.6	26.2
LM (reproduced)	87%	13.1	20.3
LM + Infer-back	89%	16.6	39.0
LM + MultiDecoder	97%	2.3	5.2
LM + Label Input Feed	87%	14.8	25.7
LM + Label Input Feed + Infer-back	87%	16.1	34.3

Table 1. Main results of various methods on the parallel datasets. The upper part shows results of existing work while the lower part shows results of our proposed method.

5.1. Main Results

We report results of our experiments by doing model selection with regard to transfer accuracy. However, since there are multiple metrics involved here, it can be considered unfair to pick certain points by this specific metric. So we present more results in the next section.

As shown in Table 1, in general it's hard to find a method that consistently outperforms other methods in all metrics. Naturally, with a higher accuracy, there comes with a lower BLEU score or a higher Perplexity score.

We can see that by training further with the infer-back method, we get consistent better results on both LM based models and input feed based models, showing the effectiveness of incorporating the pseudo-parallel sentences. The input feed method that explicitly feeds label information to the decoder for each step shows better results than the LM model which only uses label information as part of the initialization of the decoder hidden state.

5.2. Training Process

Previous work rarely mention the change of the evaluation metrics during the whole training process. However, as our exploration of the task goes deeper, we find that selecting a single model to do evaluation is quite casual. By examining closely of the training process, we get to analyze the results more properly.

We illustrate the results of non-parallel development set and parallel test set of each epoch in Figure 2 in Appendix. For non-parallel development set, we evaluate perplexity, transfer accuracy and self-BLEU score. For parallel test set, we evaluate perplexity, transfer accuracy and BLEU score.

During the pre-training phase, we see that the perplexity score decreases and then increases while the transfer accuracy score monotonically decreases, showing that the model is able to reconstruct the original sentence very well even with the transferred label information. The training phase and the back inference phase are much more unstable. It shows a trade-off between the self-BLEU score and the transfer accuracy score, since the generator is carrying the burden of decoding a same content representation to two sentences of different styles. With a high transfer accuracy score, it usually indicates that the generation model suffers from mode collapse, specifically speaking, the model is only able to generate very general sentences with a strong sentiment bias, losing preservation of the contents. On the contrary a model with high self-BLEU scores tends to copy the original sentence verbatim without changing its sentiment too much, thus resulting in a lower transfer accuracy.

We have also tried using multiple decoders (generators) for different styles, as an intent to break the BLEU-accuracy tradeoff for one single decoder. However, without tying in parameters, two decoders collapse soon with high transfer accuracy ($> 95\%$) and poor content preservation (nearly 0 self-BLEU).

In effect, results derived from both non-parallel development show that by adding the back-inference phase, we are able to generate results with a lower perplexity, a higher transfer accuracy but unfortunately a lower self-BLEU score. On the parallel dataset, generally back-inference does not cause significant decrease in parallel-BLEU, and a better combination of parallel-BLEU score and transfer accuracy via back-inference, while the perplexity is sacrificed as shown in Table 1.

By comparing the LM and input-feed based methods, we see that the input feed based ones generally have a higher self-BLEU score, indicating reinforcing the label information at

Origin Sentence	After Pre-training	After Training	After Back-inference
didn't take a second bite!	will definitely take a second!	will definitely take a premium!	will definitely take a second bite!
however my iphone sucks!	however my pleasure daughter !	however my iphone rocks!	however my iphone was great!
don't bother with this place.	will definitely recommend with this place.	will definitely recommend with this place.	will definitely return.
i'm sorry to say, nothing was that delicious or memorable	i'm definitely to say, nothing was delicious.	i'm love to say, but was delicious.	i 'm trying to say, that was great.

Table 2. The reasonable style transfer results, obtained after pre-train, fine-tuning and back-inference process respectively. The transferred sentences are supposed to have the different sentiment from the origin sentences.

Origin Sentence	After Pre-training	After Training	After Back-inference
friendly cashier and superb services	bad attitudes and service !	tbad cashier and terrible service !	terrible .
so this place literally ruined christmas for us .	so this place literally authentic for us .	so this place place around for .	so this place was great .
it smells last nasty mold all over !	it smells favorite love daily all over !	it smells our mirrors nice !	it smells great .

Table 3. From the results we can see the model trends to collapse after back-inference training as it will discard some of the content information and only use a few words to indicate the sentiment.

Origin Sentence	After Pre-training	After Training	After Back-inference
never once have i been made to feel very welcomed .	definitely once have i been made to feel .	never see though i have made to feel .	never once it have been made to feel very welcomed .
other restaurants give you steak in a bowl for about the same price .	other restaurants give i try in a resort for about great .	other restaurants can you easy in matter , such a great .	other reviewers places has you can in great .

Table 4. The unreasonable style transfer results obtained after pre-train, fine-tuning and back-inference phases respectively. The text style transfer is not capable when (1) There are sophisticated structures in the sentence such as inversion,(2)The origin sentence is considered containing little sentiment information

each decoding step helps content preservation.

5.3. Case Study

The style transfer results are investigated in Table 2, Table 3 and Table 4. We manually examine the generated sentences and categorize them into several typical types.

The transferred sentence is considered "reasonable" if it has a similar content but opposite sentiment compared to the original one. From the results we can see that, the transfer is becoming more reasonable as the training process goes on. It will generate relatively random sentences right after the pre-training phase, and the sentences generated after the training phase are more reasonable in terms of both content preservation and sentiment transfer. Though sometimes the generation model output really impressive words e.g. 'rocks' as opposed to 'sucks', most of the times, our model is still limited to the capacity of just changing a few sentiment keywords for example from 'didn't take' to 'will definitely take'.

The second group which we associate with mode collapsing phenomenon illustrates an obvious pattern that the model is only able to generate simple sentences but with a strong opposite sentiment. Mode collapse is a very common phenomenon in text generation tasks, especially non-parallel ones, mainly because the generator is not learned well with the limited expressiveness contained in the data.

The third group is the failure group and we find that the reason why the examples of this group fail is because either 1) the structure of the original sentence, such as inversion, is too complex, or 2) the original sentence actually contains little sentiment information.

5.4. Post-fit Accuracy

Inspired by Lample et al. (2019), we train post-fit classifiers for three generative model of text transfer to see if the style of the learned latent representations can be predicted or not. Specifically, we rerun three state-of-the-art models of the text transfer task, namely Fu et al. (2018), Shen et al. (2017) and John et al. (2018). We extract the hidden representations from the inference network of 10,000 data points from the Yelp dataset with half being the positive ones and the other half being the negative ones. We train a simple classifier with one hidden-layer on 8,000 data points and use 1000 data points for validation and test respectively. We demonstrate the test results in Table 5, showing that the post-fit accuracy is consistently higher than the discriminator accuracy. The phenomenon confirms that these current models, either AE-based or VAE-based ones, have not true achieved true disentanglement as claimed.

	Discriminator Acc	Post-fit Acc
Fu et al. (2018)	52 %	85 %
Shen et al. (2017)	-	93 %
John et al. (2018)	68 %	73 %
LM	-	62 %
LM + Infer-back	54 %	63 %

Table 5. Discriminator accuracy evaluated during training and Post-fit accuracy evaluated after training over the learned latent representation. The post-fit accuracy is consistently higher than the discriminator accuracy, showing that disentanglement is not truly achieved. For the upper part, we evaluate on non-parallel examples while for the lower part, we evaluate on the generated psuedo-parallel examples.

We evaluated the post-fit accuracy on our proposed method. Since we explicitly align the distribution of the generated sentences of reconstructed examples and transferred examples with our infer-back technique, we expected that the post-fit accuracy will be worse than without training with infer-back technique. However, as shown in the lower part of the Table 5, it's not the case though our technique indeed improve the performance upon base models. The results remind us of the conclusion derived from (Lample et al., 2019) that a thorough disentanglement may not really be necessary for text transfer. Also, to make a fair hypothesis, the fact that the content representations containing style information does necessarily mean that the information is effectively captured or used during generation. It might be more worthwhile to investigate fusion techniques that enables a more expressive representation for generation than a harsh eviction of style information.

6. Conclusion

We propose a back-inference technique that can be applied to current text transfer models in order to do explicit disentanglement of content and style. We found that 1) The parameter searching process is to find a equilibrium point that generates sentences with a relatively high transfer accuracy and good content preservation. There exists an obvious trade-off between these two factors. 2) Our proposed method is able to achieve at least comparable results on all the metrics and adding back-inference using pseudo-parallel data further improves upon the existing base models. 3) Cases that contain complex sentence structures like inversion are difficult to tackle and are worth further exploration using data augmentation techniques.

7. Future Work

Due to limitations of time and computational resources, we spent our major efforts on trying to reproduce Yang et al. (2018)'s results and improve upon it using our proposed techniques, while are not yet able to switch to a variational model that explicitly models the distribution of the latent representation. In that case, latent distribution could be flexibly learned by a parametric prior and sampling-based Wasserstein metric could be adopted to better align distributions (Kim et al., 2017).

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate, 2014.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets, 2016.

Fu, Z., Tan, X., Peng, N., Zhao, D., and Yan, R. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Gu, J., Im, D. J., and Li, V. O. Neural machine translation with gumbel-greedy decoding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T.-Y., and Ma, W.-Y. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pp. 820–828, 2016.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1587–1596. JMLR. org, 2017a.

Hu, Z., Yang, Z., Salakhutdinov, R., and Xing, E. P. On unifying deep generative models, 2017b.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Jin, Z., Jin, D., Mueller, J., Matthews, N., and Santus, E. Unsupervised text style transfer via iterative matching and translation, 2019.

John, V., Mou, L., Bahuleyan, H., and Vechtomova, O. Disentangled representation learning for non-parallel text style transfer, 2018.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. Bag of tricks for efficient text classification, 2016.

Kim, H. and Mnih, A. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

Kim, Y., Zhang, K., Rush, A. M., LeCun, Y., et al. Adversarially regularized autoencoders. *arXiv preprint arXiv:1706.04223*, 2017.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Lample, G., Subramanian, S., Smith, E., Denoyer, L., Ranzato, M., and Boureau, Y.-L. Multiple-attribute text rewriting. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1g2NhC5KQ.

Li, J., Jia, R., He, H., and Liang, P. Delete, retrieve, generate: A simple approach to sentiment and style transfer, 2018.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

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

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015.

Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. Style transfer from non-parallel text by cross-alignment, 2017.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

Wen, T.-H., Vandyke, D., Mrki, N., Gasic, M., Rojas Barahona, L. M., Su, P.-H., Ultes, S., and Young, S. A network-based end-to-end trainable task-oriented dialogue system. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017. doi: 10.18653/v1/e17-1042. URL http://dx.doi.org/10.18653/v1/e17-1042.

Yang, Z., Hu, Z., Dyer, C., Xing, E. P., and Berg-Kirkpatrick, T. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, pp. 7298–7309, 2018.

Zhu, J.-Y., Krhenbhl, P., Shechtman, E., and Efros, A. A. Generative visual manipulation on the natural image manifold, 2016. ISSN 1611-3349. URL http://dx.doi.org/10.1007/978-3-319-46454-1_36.

APPENDIX

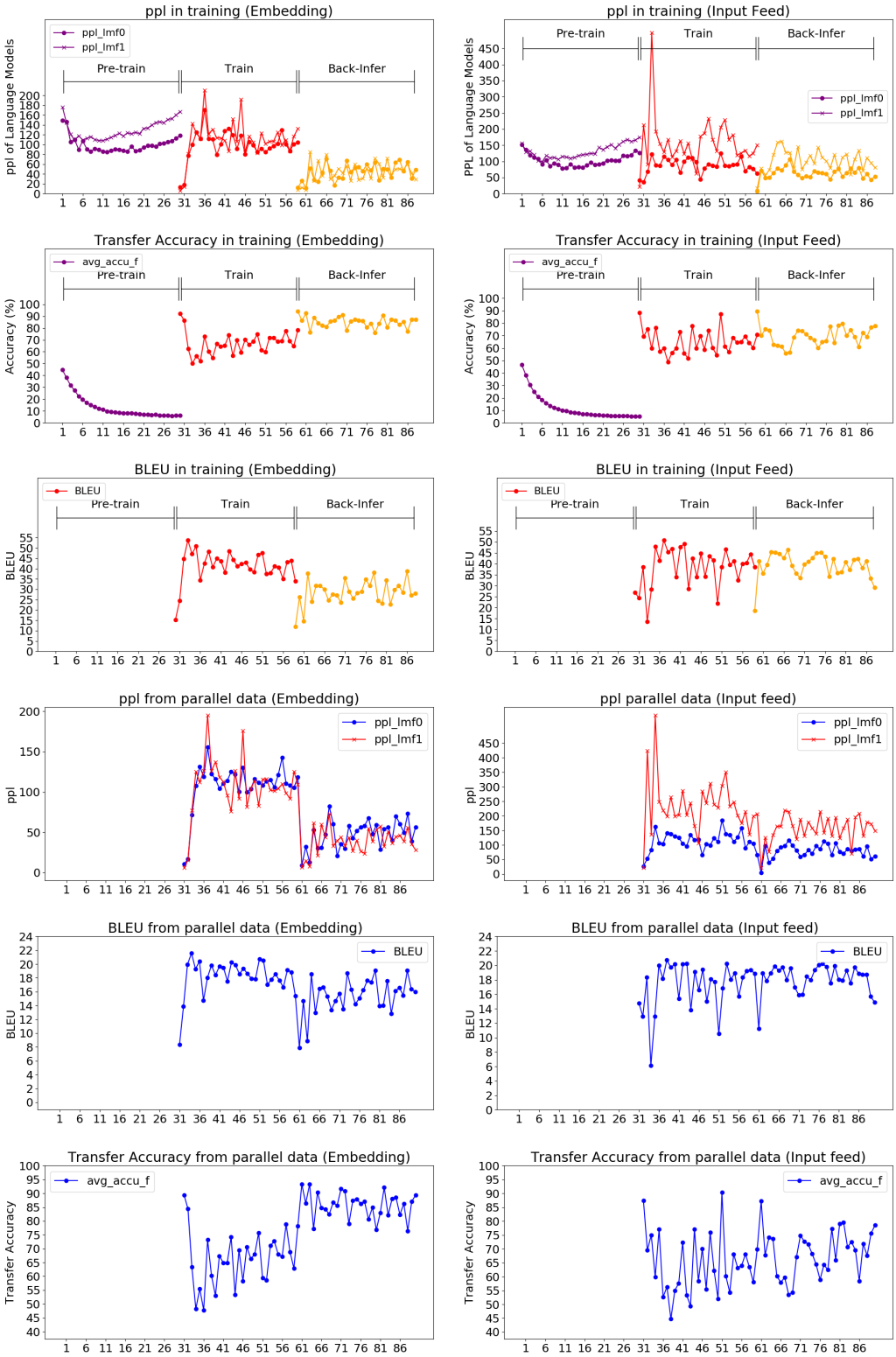

Figure 2. The three-phase training process of LM (left column) and Input feed model (right column) on Yelp dataset. The left column is for our LM model and the right column our LM + InputFeed Model.