Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 4671–4679
Marseille, 11–16 May 2020

Decode with Template: Content Preserving Sentiment Transfer

Zhiyuan Wen†, Jiannong Cao†, Ruosong Yang†, Senzhang Wang‡
†Department of Computing, The Hong Kong Polytechnic University,
‡College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
†Kowloon, Hong Kong, China, ‡Nanjing, China
{cszwen, csjcao, csryang}@comp.polyu.edu.hk, szwang@nuaa.edu.cn

Abstract

Sentiment transfer aims to change the underlying sentiment of input sentences. The two major challenges in existing works lie in (1) effectively disentangling the original sentiment from input sentences; and (2) preserving the semantic content while transferring the sentiment. We find that identifying the sentiment-irrelevant content from input sentences to facilitate generating output sentences could address the above challenges and then propose the Decode with Template model in this paper. We first mask the explicit sentiment words in input sentences and use the rest parts as templates to eliminate the original sentiment. Then, we input the templates and the target sentiments into our bidirectionally guided variational auto-encoder (VAE) model to generate output. In our method, the template preserves most of the semantics in input sentences, and the bidirectionally guided decoding captures both forward and backward contextual information to generate output. Both two parts contribute to better content preservation. We evaluate our method on two review datasets, **Amazon** and **Yelp**, with automatic evaluation methods and human rating. The experimental results show that our method significantly outperforms state-of-the-art models, especially in content preservation.

Keywords: Text Generation, Sentiment Analysis, Bidirectionally Guided Decoding

1. Introduction

Sentiment transfer for text considers the semantics of a sentence in two aspects: the sentiment information, and the content independent to sentiment information[1]. This task aims to change the underlying sentiment of the input text and simultaneously retain the content. As an example shown in Table 1., only the attitude to the restaurant in the review is changed, while the sentiment-independent content about the restaurant is preserved. This task requires to generate sentences that (1) conform to the target sentiments, (2) preserve the semantic content of the input sentences, and (3) be fluent and readable (Jin et al., 2019). It connects sentiment analysis and Natural Language Generation (Zhang et al., 2018a) and facilitates a lot of NLP applications such as fighting against offensive language in social media (Santos et al., 2018), news rewriting, and building controllable dialogue systems. However, this task is difficult in practice due to the lack of parallel data (sentences with similar content but different sentiments).

Several recent works (Shen et al., 2017; Hu et al., 2017; Yang et al., 2018; John et al., 2018) try to disentangle sentiments from content by assuming the texts are generated conditioned on two latent distributional representations: one with only content information, and the other with only sentiment information. Most of them focus on changing the sentiment yet fail to keep the content. The reason is that distributional disentanglement needs the latent representations of sentiment and content to be orthogonal or independent. However, it is hard to guarantee that each representation contains only the corresponding information. Therefore, reconstruction from these two parts directly might cause confliction in both content and sentiment aspects, which leads to poor performance in content preservation.

Positive to negative sentiment transfer	
Input:	I love this place , the service is always **great**!
Output:	I **hate** this place, the service is **bad**.

Table 1: An example of sentiment transfer. The input sentence is a review of restaurant service with positive sentiment. The sentiment transfer model changes the input to a negative review but preserves the sentiment-free content.

Instead of modifying the sentiment only in latent distributional space, we consider this task as a combination of instance-level modification with semantic generation and propose our method: Decode with Template. In our model, we adapt the variational autoencoder (VAE) (Kingma and Welling, 2013) by using bidirectional Gated Recurrent Units RNNs (GRU) (Cho et al., 2014) for both the encoder and the decoder. Inspired by (Li et al., 2018; Zhang et al., 2018b; Wu et al., 2019), we first generate the templates by masking all the sentiment words to eliminate the original sentiment information in input sentences. Then, the templates are fed into the encoder to get the semantic content representations. Next, we modify the templates by replacing the masked sentiment words with the target sentiment representations we got from the sentiment memory we build. Finally, we input the content representations together with the modified templates into our bidirectional decoder to generate output sentences. To improve the model ability to generate sentences rendering target sentiments, we also use a sentiment classifier to perform an adversarial training.

In our method, the templates can well preserve the semantic content of input sentences. Besides, the latent representations from the encoder robustly capture the semantic infor-

[1] Henceforth, we use *content* to denote content independent to sentiment information for simplicity.

mation. By using the bidirectional GRU as the decoder and the modified templates as its partial input, both forward and backward contextual information can be captured for better preserving the content of the input sentences. Besides, the bidirectionally guided decoding also prevents the error accumulation in the unidirectional autoregressive RNN language models based decoder, which is commonly used in many previous works (Hu et al., 2017; Shen et al., 2017; Fu et al., 2018; Li et al., 2018). Moreover, we use the target sentiment representations to modify the templates, and thus the sentiment information and contextual information can be integrated for generation. Our method combines instance-level modification and semantic generation and thus achieves better content preservation and naturalness for the output sentences.

To demonstrate the effectiveness of our approach, we conduct experiments on two review datasets, **Amazon** and **Yelp**. We evaluate the performance in both automatic metrics and human evaluation from three aspects: sentiment transfer intensity, content preservation, and naturalness (Mir et al., 2019). Results show that our method significantly outperforms state-of-the-art models. Besides, we also conduct ablation study to show how each component in our method affects the overall performance. We summarize our contributions as follows:

- We propose the Decode with Template model that combines instance-level modification with semantic generation for the sentiment transfer task without parallel data.
- We innovatively use the modified templates and enables a bidirectionally guided decoder, which captures both forward and backward context in decoding and prevents the error accumulation in unidirectional autoregressive RNN decoder. Also, the bidirectionally guided decoding could be easily adapted to many other modification and generation tasks.
- The proposed method significantly outperforms state-of-the-art approaches in public sentiment transfer datasets, especially in content preservation.

2. Related Works

The problem of sentiment transfer is a special issue of text style transfer, which requires to transfer the original text styles of sentences into desired ones. Since there is little text data with explicit style labels, most previous researches regard sentiment as a kind of text style and focus on sentiment transfer due to the abundant data and research in sentiment analysis.

Earlier works modify text styles in a semantic disentanglement way. (Hu et al., 2017) first proposed a neural generative model that combines VAEs and style discriminators for the forceful imposition of style and semantic structures. (Shen et al., 2017) assume that two corpora of sentences share the same distribution of content albeit rendered in different styles. They hence separate styles from semantic content by mapping the input sentences to its pure content representation, and then pass the representation to specified style-dependent decoders for rendering. (Fu et al., 2018) extended the above ideas by using an adversarial network to discourage encoding style information into the content representations. Though it is intuitive to separate style and content in semantic space, their works did not perform well in content preservation and rendering target styles due to the impure disentanglement.

To better preserve the content, (Prabhumoye et al., 2018; Jin et al., 2019) use the back-translation techniques borrowed from neural machine translation and obtain reasonable performance yet turn out complicated in practice. (Li et al., 2018) proposed the TemplateBased method only to modify the sentiment words in input sentences, which is easy to operate yet leads to poor naturalness. To endow the target styles into the output sentences, (Li et al., 2018) also propose to concatenate the sentiment embeddings with semantic representations for decoding. Differently, (Lample et al., 2019) use multiple attribute embeddings as the start-of-sequence($\langle SOS \rangle$) input to the decoder in generation. Both the above methods use style attribute as partial decoder input. Besides, (Prabhumoye et al., 2018) use different style discriminators to guide the generation adversarially.

For the output generation, most of the previous works use unidirectional RNN-like decoder due to its excellent performance in text generation. However, the error accumulation caused by only using historical contextual information to generate the next words autoregressively is unignorable. Compared to the works above, the main innovation of our method is that we refrain separation in semantic space by combining semantic generation with instance-level modification, so that achieves better content preservation.

3. Decode with Template

In this section, we will first formalize our problem definition, then present an overview of the proposed Decode with Template model. Then we will introduce how to generate the templates, and how to modify the templates with desired sentiments. The adapted bidirectionally guided VAE model will be elaborated next. Finally, we will introduce the adversarial training with sentiment classifier and the overall loss.

3.1. Problem Statement

The studied problem is formally defined as follows. Given a set of sentences with sentiment labels $X = \{(x_1, y_1), ..., (x_n, y_n)\}$, where x_i is a sentence whose sentiment label (either "positive" or "negative") is indicated by y_i, the goal is to build a model that can generate a readable sentence $\hat{x_i}$ rendering the sentiment $\hat{y_i}$ opposite to y_i, and at the same time preserving the content of x_i.

3.2. Model Overview

As shown in Figure 1, the Decode with Template model contains four parts. For each input sentence, we first mask the sentiment words to generate a template without sentiment information. Then we input the template into the encoder (the left part of Figure 1) to learn the content representation. Next, we modify the template by replacing the masked words with the target sentiment representations (the right lower part of Figure 1). Finally, we feed both the learned semantic content representation and the modified template into the decoder to generate the output sentences (the right upper part of Figure 1). During model training, a sentiment classifier is also used as the discriminator to enhance the model ability to generate sentences that render the target sentiment in an adversarial learning way. Our model can be formalized as below:

$$\begin{aligned} temp_i &= F_{mask}(x_i, y_i) \\ z_i &= E(temp_i) \\ \hat{temp}_i &= F_{modify}(temp_i, \hat{y_i}) \\ \hat{x_i} &= G(\hat{temp}_i, z_i) \end{aligned} \tag{1}$$

where F_{mask} is a function that utilizes an external sentiment lexicon to replace the sentiment words in each input sentence x_i with a token "⟨neutral⟩". $temp_i$ is the template contains only the semantic content words of x_i. E is the encoder that takes $temp_i$ as input, and generates the content representation z_i. F_{modify} is a function to modify the sentiment independent template $temp_i$ to $\hat{temp}_i$ with the target sentiment representations of $\hat{y_i}$. G is the bidirectional decoder and $\hat{x_i}$ is the output sentence rendering the target sentiment $\hat{y_i}$. In the following chapters, we will introduce our method in detail.

3.3. Template Generation

We generate the templates that preserve the semantic content by masking all the sentiment words in the input sentences. (Li et al., 2018) shows that masking sentiment words is a simple yet effective way to eliminate the sentiment information since the sentiment of a sentence is usually expressed by explicit sentiment words. We use a sentiment lexicon that consists of 5106 negative words and 2759 positive words provided by (Zeng et al., 2018) to detect the sentiment words in input sentences. We use this lexicon because it combines two classical lexicons in sentiment analysis: the Subjectivity Lexicon (Wilson et al., 2005) and the Opinion lexicon (Hu and Liu, 2004). The sentiment words in each sentence are detected by identifying whether the stem of each word exists in the stemmed sentiment lexicon. This comparison can effectively eliminate the influence of tense and voice. After the detection, we then mask the sentiment words in each sentence with a token "⟨neutral⟩" and keep other words fixed to obtain the templates.

3.4. Template Modification

Next, we modify the generated templates to endow the desired sentiments by replacing the token "⟨neutral⟩" with the representations of the target sentiments. Note that the sentiment representations should be suitable for the semantic content of the template. Thus the modification should be combined with the contextual information of the template with target sentiment information.

Inspired by (Sukhbaatar et al., 2015) and (Zhang et al., 2018a), we use the lexicon described previously as a sentiment memory to generate suitable sentiment representations for each template. Formally, for each sentence template $temp = \{t_1, t_2, ..., t_n\}$ where t_i is the unmasked word and a target sentiment $\hat{y}$, the corresponding sentiment representation $rep(\hat{y})$ can be obtained by:

$$rep(\hat{y}) = \frac{1}{n * m^{\hat{y}}} \sum_{i=1}^{n} \sum_{j=1}^{m^{\hat{y}}} match(t_i, sent_j^{\hat{y}}) sent_j^{\hat{y}} \tag{2}$$

where $sent_j^{\hat{y}}$ is a sentiment word in the lexicon with the label $\hat{y}$, and $m^{\hat{y}}$ is the number of these sentiment words. n is the number of unmasked words in the template. $match$ calculates the match score between t_i and $sent_j$ as the averaging weight for each $sent_j^{\hat{y}}$, here we use the cosine similarity between their representations. Intuitively, we use the average of all the t_i as the overall semantic representation of $temp$, and then extract suitable sentiment information with the Attention mechanism. The reason we use this method is that the average of word vectors preserve the contextual similarity with the sentiment words, and also to some extent preserve the semantics of the templates as the sentence embedding (Arora et al., 2016).

3.5. Bidirectionally Guided VAE Model

We complement the vanilla sentence-VAE model (Bowman et al., 2015) by using bidirectional GRU for both the encoder and the decoder, because the latent feature from the encoder as content representation captures semantic information robustly. Besides, the bidirectionally guided decoding utilizes both forward and backward contextual information, and better preserves the content.

3.5.1. Content Encoding

We assume that the content of sentences with both positive and negative sentiment share the same latent semantic space. So, our model first imposes a prior distribution $p(\vec{z})$ on the content in the semantic space, and then assumes that the content representations $\vec{z}$ for both positive and negative sentiment could be sampled from $p(\vec{z})$. For each sentence x, our model takes its template $temp$ with the original sentiment words masked as the encoder input, projecting it into a unique region in the semantic space. Formally, the region is a learned posterior distribution $q(\vec{z}|temp)$ described by the mean μ and the standard deviation σ. Then, the content representation $\vec{z}$ could be sampled from the region. Not only all the samples in the region contain similar semantic information, but the training process also forces our model to decode plausible sentences from each sample robustly.

3.5.2. Bidirectionally Guided Decoding

With the content representation $\vec{z}$, we next conduct a bidirectionally guided decoding to generate output sentences from $\vec{z}$ by using the modified templates (previously introduced) as partial decoder input.

During generation, the decoder receives $\vec{z}$ and the modified template $\hat{temp}$ as the input. $\hat{temp}$ contains possible future words in the context that enables bidirectionally guided decoding. Formally, In i-th decoding step, the decoder cell is conditioned on the i-th input word from $\hat{temp}$, as well as the bidirectional hidden states $h_i = [h_i^f, h_i^b]$ to generate the output word. Where the h_i^f refers to the forward, and the h_i^b refers to the backward.

The strength of bidirectionally guided decoding lies in two aspects. First, it captures both forward and backward contextual information to preserve the semantic content better.

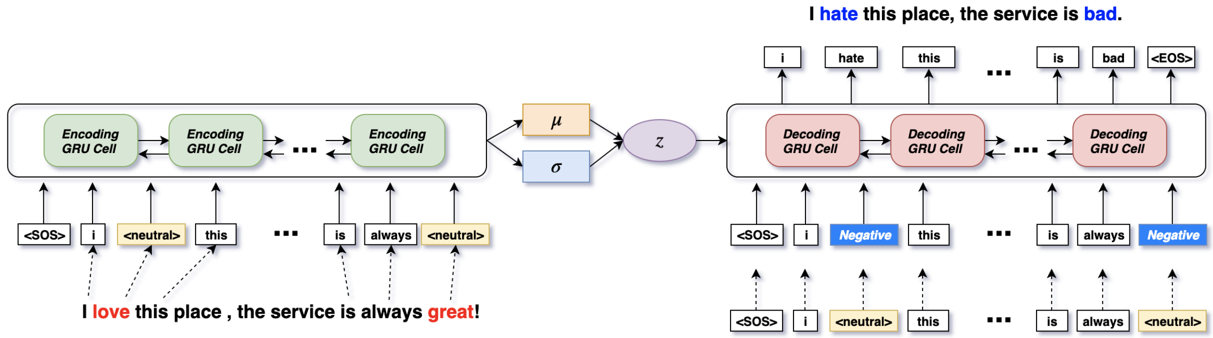

Figure 1: Model illustration with an example. The input sentence is with positive sentiment. The model first detects "love" and "great" as positive words, and then masks them with "⟨neutral⟩" and keeps other words fixed. Then the masked input is fed in the encoder to get the mean μ and the standard deviation σ describing the semantic distribution. Then z is sampled from the distribution as the content representation without sentiment information. The decoder receives z as well as the modified template where all the "⟨neutral⟩"s are replaced by the negative representations. Finally, the decoder generates a sentence with the negative sentiment, while the content is similar to the input.

Second, it prevents the error accumulation and relieves the non-linearities being prone to gradient vanishing (Mao et al., 2019) caused by autoregressive RNN decoder. Moreover, since the target sentiment representations are fed into the decoder through the modified templates, the target sentiment integrated could influence each decoding step to output more natural sentences.

3.5.3. Training Loss

The general target of our model is to generate plausible sentences conditioned on the content representations and modified templates with specified sentiments. Since parallel data is unreachable, during training, the model is to reconstruct the input sentences with the original sentiment. After training, the aim changes to generate sentences that preserve the original content and render the opposite sentiment.

Therefore, there are two objectives during training: (1) to learn a posterior distribution $q_\theta(\vec{z}|temp)$ close to the prior $p(\vec{z})$, which is supervised using the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) penalty, and (2) to reconstruct the input sentences x from the content representation $\vec{z}$ conditioned on the original sentiment y. Formally, the training is to minimize the loss:

$$\begin{aligned}\mathcal{L}_{vae}(\theta) = &-\boldsymbol{\lambda}_{kl} KL(q_\theta(\vec{z}|temp)||p(\vec{z})) \\ &+ \mathbb{E}_{q_\theta(\vec{z}|temp)}[logp_\theta(x|y,\vec{z})]\end{aligned} \tag{3}$$

where θ is the model parameters to be learned. $p(\vec{z})$ is the prior set which can be a standard Gaussian ($\boldsymbol{\mu} = \vec{0}, \boldsymbol{\sigma} = \vec{1}$), and $q_\theta(\vec{z}|temp)$ is the posterior taking the form $\mathcal{N}(\boldsymbol{\mu}, diag\boldsymbol{\sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are generated from the template encoder. $logp_\theta$ is the negative $logloss$ of reconstructing x. $\boldsymbol{\lambda}_{kl}$ is the adaptive parameter to balance the reconstruction loss $\mathbb{E}$ and the KL penalty. We conform to the annealing method proposed in (Bowman et al., 2015) to calculate $\boldsymbol{\lambda}_{kl}$ by:

$$\boldsymbol{\lambda}_{kl} = sigmoid(-k * step - step_0) \tag{4}$$

where $step$ is the number of current training batches, and k and $step_0$ are the hyper-parameters.

3.6. Adversarial Training with Sentiment Classifier

To further guide the generated sentences to render the target sentiments, we also conduct an adversarial training by back-propagating the sentiment classification loss for the generated sentences to the decoder. We use a Convolutional Neural Network (CNN) as the sentiment classifier, and minimize the Cross-Entropy loss $\mathcal{L}_{sent}$ below during training:

$$\begin{aligned}\mathcal{L}_{sent} = -\sum_{i=1}^{n} &[y_i log(f(x_i)) \\ &+(1-y_i)log(1-f(x_i))]\end{aligned} \tag{5}$$

where for each generated x_i, y_i is the sentiment label (1 for "positive", 0 for "negative"), and $f(x_i)$ is the probability of x_i rendering positive sentiment.

However, due to the discreteness of the generated text, the gradients of sentiment classification loss could not be directly propagated from the classifier to our VAE model. In existing works, (Yu et al., 2017) solve the similar problem using Policy Gradient (Sutton et al., 2000), which turns out to suffer from high variance. Besides, (Hu et al., 2017; Prabhumoye et al., 2018) use a continuous approximation of Softmax and achieved competitive results. Based on their work, we add the Gumbel noise by considering the generation of sentences as sampling words from discrete distributions. Hence, we use the Gumbel-Softmax (Jang et al., 2016) distribution p over the whole vocabulary to

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

Table 2: Statistics of Yelp and Amazon datasets

approximate one-hot vector for selecting words for output sentences. The probability p_i for i-th word is calculated by:

$$p_i = \frac{exp((log\pi_i + g_i)/\tau)}{\sum_{j=1}^{V} exp((log\pi_j + g_j)/\tau} \tag{6}$$

where π_i is the probability from Softmax of choosing the i-th word, V is the length of the vocabulary. g_i is the noise independently sampled from $Gumbel(0, 1)$. τ is a temperature parameter, and we use an annealing strategy to update it during training. The initial value of τ is set to 1.0, and it would decay to $\tau exp(-bn * 0.00003)$ after every 100 batches until reaching the minimum value of 0.1. bn is the batch number.

After we got p_i for each word consist of the generated sentences x_i in Equ 5, each word w_i in $x_i = [w_1, ..., w_n]$ is obtained by:

$$w_i = \sum_{j=1}^{V} p_j embed_j \tag{7}$$

where $embed_j$ is the pre-trained word vector for the j-th word in the whole vocabulary.

3.7. Overall Objective

To combine the above described two partial losses together, the overall objective is to minimize the following loss function:

$$\mathcal{L} = \alpha \mathcal{L}_{vae}(\theta) + \beta \mathcal{L}_{sent} \tag{8}$$

where α and β are weight hyper-parameters to balance the two losses, respectively.

4. Experiments

4.1. Dataset

We evaluate our model by conducting experiments on **Yelp** and **Amazon** reviews datasets (Table 2) released by (Li et al., 2018). The sentences in Yelp dataset are reviews about restaurants and movies. While in Amazon dataset, the reviews are about online shopping products (He and McAuley, 2016). Each sentence in these two datasets is labeled as having either a positive or negative sentiment. Both datasets are randomly split into train, validation, and test sets.

4.2. Experiment Setup

We use single-layer bidirectional GRU neural networks for both encoder and decoder with the hidden dimension of 200, and the dimension of input word embeddings to be 300. The word embeddings used for model input and sentiment representations generation are pre-trained GloVe (Pennington et al., 2014) word vectors. We use a batch size of 32 for input sentences. The k and $step_0$ to calculate $\boldsymbol{\lambda}_{kl}$ are set to 0.0025 and 2500, respectively. The α and β to balance the two partial losses are set to 0.4 and 0.5. We use Adam (Kingma and Ba, 2014) optimization algorithm to train our VAE model and the Adabound (Luo et al., 2019) to train the CNN sentiment classifier. The initial learning rate is set to 0.001 for both models. Other hyper-parameters are chosen by grid search based on the performance on the validation set.

4.3. Baselines

We compare our method with the following five representative state-of-the-art approaches as the baseline models.

Cross-Alignment Auto-Encoder (CAAE): This apporach is proposed in (Shen et al., 2017). It leverages refined alignment of latent representations in the hidden layers to perform text style transfer.

Control and Generation (CtrlGen): This apporach is proposed by (Hu et al., 2017). CtrlGen combines the variational auto-encoders and different style discriminators for the effective imposition of style and semantic structures.

TemplateBased: This approach simply delete the original sentiment words in each input sentence as a template, then fill in with selected target sentiment words (Li et al., 2018).

DeleteAndRetrieve: This method is also proposed in (Li et al., 2018). It combines the template above with retrieved suitable target sentiment words as the input, then generates output sentences through a Seq2seq RNN model.

Back-translation for Style Transfer (BST): This model is proposed in (Prabhumoye et al., 2018). It uses back-translation to preserve content and style-specific generators to render target styles.

We regard **CAAE** and **CtrlGen** in distributional disentanglement way, **TemplateBased** and **DeleteAndRetrieve** as instance-level modification and **BST** in back-translation way.

4.4. Automatic Evaluation

We report our results on the test sets of automatic evaluation in two aspects: the sentiment transfer intensity and the content preservation.

For sentiment transfer intensity, we use the classification accuracy (ACC) for output sentences from a pre-trained TextCNN model as described in (Kim, 2014). After fine-tuning, it achieves nearly perfect accuracy of 97.6% on our dataset.

For content preservation, we first compute the BLEU (Papineni et al., 2002) score between output sentences and human references (provided by (Li et al., 2018) as ground truth). Besides, we also use the Word Mover's Distance (WMD) to calculate the minimum "distance" between word embeddings of output and human references, where a

Yelp	ACC	BLEU	WMD
CAAE	0.772	4.9	11.655
CtrlGen	0.849	3.4	13.278
TemplateBased	0.849	16.3	4.122
DeleteAndRetrieve	0.903	11.3	7.651
BST	0.895	20.9	3.985
Our method	**0.930**	**25.2**	**3.126**
Amazon	**ACC**	**BLEU**	**WMD**
CAAE	0.587	5.1	10.354
CtrlGen	0.695	2.9	13.100
TemplateBased	0.703	25.6	3.290
DeleteAndRetrieve	0.640	21.3	4.058
BST	0.705	25.8	3.744
Our method	**0.752**	**27.9**	**3.2**

Table 3: Automatic evaluation result

Yelp	Sentiment	Content	Naturalness
CAAE	2.379	1.605	2.506
CtrlGen	3.445	1.764	2.730
TemplateBased	3.304	3.998	2.489
DeleteAndRetrieve	2.501	3.584	3.500
BST	2.437	3.453	3.565
Our method	**3.449**	**4.173**	**3.709**
Amazon	**Sentiment**	**Content**	**Naturalness**
CAAE	2.643	1.455	2.834
CtrlGen	3.055	2.631	3.001
TemplateBased	**3.273**	3.400	2.340
DeleteAndRetrieve	2.309	3.220	3.554
BST	2.803	3.661	3.150
Our method	3.221	**3.845**	**3.669**

Table 4: Human evaluation result

smaller distance signifies a higher similarity (Kusner et al., 2015). The human references are sentence pairs with opposite sentiments but the same contents, manually modified by workers. The results are shown in Table 3.

Higher ACC and BLEU score means better performance, while smaller WMD signifies better content preserving. One can see from Table 3 that our method achieves the best overall performance on both datasets, especially in content preservation. The BLEU score is largely improved from 20.9 to 25.2 in **Yelp** dataset. Distributional disentanglement methods **CAAE** and **CtrlGen** achieve lower performance in preserving content, mainly because of the impure disentanglement in latent space. The performance of **BST** signifies that back-translation is an effective method to capture content information. Also, **TemplateBased** achieves competitive performance, which shows the advantage of instance-level modification to preserve content. Our method achieving the best performance demonstrates the effectiveness of the combination of instance-level modification and semantic generation.

4.5. Human Evaluation

To capture more aspects of the performance on this task, we also conduct a human evaluation of the generated results. We follow the evaluation method proposed by (Mir et al., 2019) to obtain human ratings on sentiment transfer intensity, content preservation, and naturalness. We randomly select 100 sentences from the test set and then collect the transfer results for each approach. Each rater is given a questionnaire consisting of 100 questions. For each question, the rater is asked to rank six transfer results (by 1-5, 5 means the best performance) corresponding to the input sentence in the three aspects above. We asked four raters to give their annotations. To make the result more convincing, we also calculate the inter-rater agreement according to (Krippendorff, 2018). The agreement on our raters is 0.70, 0.78, 0.69 for transfer intensity, content preservation, and naturalness, respectively.

We average the human rating in each evaluation metric, and the result is shown in Table 4. Our method achieves substantially the best results in all three aspects. It is worth mentioning that our method outperforms all baseline models in Naturalness. A possible explanation that is we use bidirectional decoder as well as templates for generation, which provides more contextual information. Although **TemplateBased** simply replace words and shows poor Naturalness, the explicit sentiment words in their result contribute to considerable performance in sentiment transfer intensity.

Other methods **CAAE**, **CtrlGen**, **DeleteAndRetrive** and **BST** use autoregressive RNN decoders for generation also output readable (fairly good the Naturalness) sentences, yet insufficiently preserve semantic content. It mainly because the error accumulation in decoding brings deviation to the original contents.

Yelp	Accuracy	BLEU	WMD
Our method	0.930	25.2	3.126
w/o Template	-	4.5	10.343
w/o Content Rep.	0.912	17.6	5.617
w/o Adversarial Training	0.884	22.2	3.170
Amazon	**Accuracy**	**BLEU**	**WMD**
Our method	0.752	27.9	3.281
w/o Template	-	3.7	13.600
w/o Content Rep.	0.751	20.1	4.399
w/o Adversarial Training	0.712	24.5	3.390

Table 5: Ablation study result

4.6. Ablation Study

We conduct ablation study to evaluate the contribution of three important components (modified template, content representation, and the adversarial training with the sentiment classifier) in our approach. We remove each component from our model independently to see the influence of the performance on different aspects. The result is shown in Table 5.

We first remove the modified templates from decoder input, the BLEU score descends dramatically from 25.2 to 4.5 on **Yelp** and from 27.9 to 3.7 on **Amazon** dataset. Also, the WMD has a tremendous rise around three to four times on both datasets. It indicates the template plays a vitally important role for content preservation in the bidirectionally guided decoding. Since removing the template also removes the target sentiment representations, we do not show the results of the sentiment transfer accuracy. We next independently disable the semantic representation by set it to random, causing a substantial reduction of BLEU on both datasets. It suggests that the semantic representation is also essential to preserve content. However, the lack of semantic representation brings little decrease in sentiment transfer accuracy. It is because we endow target sentiments by directly modifying the templates. Finally, we remove the loss $\mathcal{L}_{sent}$ to eliminate the supervision from the sentiment classifier during training, finding that the sentiment transfer accuracy goes down remarkably. It verifies that the adversarial training does help the generated sentences render target sentiments.

To sum up, the modified template is a critical component to enhance decoding for content preservation. Also, the supervision from the adversarial training mainly contributes to successful transferring the sentiment.

4.7. Evaluation of Lexicons Usage

Since our method utilizes the external lexicon to facilitate both template generation and template modification, it is also important to evaluate the impact of the lexicon sizes. We randomly select 25%, 50%, 75%, 100% from both positive and negative words in the lexicons we use to compare the performance of our model in sentiment transfer accuracy. Below in Table 6 is the average performance of 10 times running.

Lexicon size	ACC in Yelp	ACC in Amazon
1967 (25%)	0.885	0.737
3933 (50%)	0.878	0.719
5899 (75%)	0.922	0.752
7865 (100%)	**0.930**	**0.752**

Table 6: Comparison between using different lexicon sizes in sentiment transfer accuracy

We can see that as the size of the lexicon grows, the sentiment transfer accuracy in both **Yelp** and **Amazon** datasets are also improved moderately. When we use 25% and 50% of the lexicon, the accuracies are close; while when we increase to 75%, there is a considerable improvement. It suggests a comprehensive lexicon does provide more sufficient sentiment information in our method.

4.8. Case Study

We further analyze the output sentences from our method and sampled seven pairs shown in Table 7. For the sentences with explicit sentiment words, our approach could effectively change them, resulting in word replacement (e.g. "worst" to "best") or adding negation words (e.g. "very helpful" to "not helpful at all"). Our method can also transfer the underlying sentiment without explicit sentiment words by rendering the target sentiment integrated with semantic content, such as converting "pay very much attention" to "rush and do not pay attention" to describe the waiters, or "horrible" to "made my day".

Positive to **Negative**
they bring it out front for you and are **very helpful**.
they bring it out front for you and are **not helpful at all**.
they pay very much attention to customers!
they rush and **do n't** pay attention to their customers.
i **love** italian and i eat here **often**.
i **hate** italian and i **do n't** eat here.
Negative to **Positive**
the marinara sauce **had no flavor**.
the marinara sauce **is so flavorful**.
the chocolate cake was the **worst** i had eaten in a while.
the chocolate cake was one of the **best** desserts i 've ever had.
the food was pretty **bad** , i would **not** go there again.
the food was pretty **good** i would **definitely** go there again.
the queen bed was **horrible**
the queen bed made my day

Table 7: Example result sentences. The first lines are input sentences, and the second lines are output sentences from our model.

Transferring the underlying sentiments would also inevitably change the sentiment related actions in the semantic content. For example, transferring "i love italian and i eat here often" to "i hate italian and i don't eat here" also changes the frequency the user go to the Italian restaurant. However, it is still acceptable that the two sentences both describe the attitude to the restaurant. Moreover, as a sacrifice of content preservation, our method does not bring much variance in sentence structures.

5. Conclusions and Future Work

In this paper, we focus on the content preservation in sentiment transfer task and propose the Decode with Template model to effectively modify the underlying sentiment of input sentences. We use the template where the explicit sentiment words are modified as decoder input, so that enables a bidirectionally guided decoding to capture both forward and backward contextual information to generate output. Our method effectively preserves the semantic content and naturalness for output sentences. Besides, the proposed bidirectionally guided decoding could be generally adapted in other text modification and generation tasks. We conduct experiments on two review datasets, and the results show our approach significantly outperforms state-of-the-art methods, especially in content preservation. The ablation study also shows the importance of the templates in decoding to preserve semantic content.

We consider our work also to be an application to the sentiment lexicon, so, for future work, we plan to explore the construction of different style lexicons, so that our method could be utilized in more general text style transfer tasks. Also, we are interested in extending our method to other text modification tasks, like lexical correction and writing polishing.

6. Acknowledgement

This work is supported by PolyU Teaching Development with project code 1.61.xx.9A5V and RGC Collaborative Research Fund 2018/19 with project code C6030-18G.

7. Bibliographical References

Arora, S., Liang, Y., and Ma, T. (2016). A simple but tough-to-beat baseline for sentence embeddings.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

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

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*, pages 507–517. International World Wide Web Conferences Steering Committee.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

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

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

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

John, V., Mou, L., Bahuleyan, H., and Vechtomova, O. (2018). Disentangled representation learning for text style transfer. *arXiv preprint arXiv:1808.04339*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

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

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology*. Sage publications.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.

Lample, G., Subramanian, S., Smith, E., Denoyer, L., Ranzato, M., and Boureau, Y.-L. (2019). Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Li, J., Jia, R., He, H., and Liang, P. (2018). Delete, retrieve, generate: A simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*.

Luo, L., Xiong, Y., Liu, Y., and Sun, X. (2019). Adaptive gradient methods with dynamic bound of learning rate. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, Louisiana, May.

Mao, Q., Li, J., Wang, S., Zhang, Y., Peng, H., He, M., and Wang, L. (2019). Aspect-based sentiment classification with attentive neural turing machines. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5139–5145. AAAI Press.

Mir, R., Felbo, B., Obradovich, N., and Rahwan, I. (2019). Evaluating style transfer for text.

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*, pages 311–318. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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

Santos, C. N. d., Melnyk, I., and Padhi, I. (2018). Fighting offensive language on social media with unsupervised text style transfer. *arXiv preprint arXiv:1805.07685*.

Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. (2017). Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, pages 6830–6841.

Sukhbaatar, S., Szlam, A., Weston, J., and Fergus, R. (2015). End-to-end memory networks.

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

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

Wu, X., Zhang, T., Zang, L., Han, J., and Hu, S. (2019). " mask and infill": Applying masked language model to sentiment transfer. *arXiv preprint arXiv:1908.08039*.

Yang, Z., Hu, Z., Dyer, C., Xing, E. P., and Berg-Kirkpatrick, T. (2018). Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, pages 7287–7298.

Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Zeng, Y., Lan, Y., Hao, Y., Li, C., and Zheng, Q. (2018). Leveraging multi-grained sentiment lexicon information for neural sequence models. *arXiv preprint arXiv:1812.01527*.

Zhang, Y., Xu, J., Yang, P., and Sun, X. (2018a). Learning sentiment memories for sentiment modification without parallel data. *arXiv preprint arXiv:1808.07311*.

Zhang, Y., Fu, J., She, D., Zhang, Y., Wang, S., and Yang, J. (2018b). Text emotion distribution learning via multi-task convolutional neural network. In *IJCAI*, pages 4595–4601.