24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Fine-grained Text Sentiment Transfer via Dependency Parsing

Lulu Xiao [1] and **Xiaoye Qu** [1] and **Ruixuan Li**[1*] and **Jun Wang**[2] and **Pan Zhou** [1] and **Yuhua Li**[1]

Abstract. Fine-grained sentiment transfer demands to edit an input sentence on a given sentiment intensity while preserving its content, which largely extends traditional binary sentiment transfer. Previous works on sentiment transfer usually attempt to learn latent content representation disentangled from sentiment. However, it is difficult to completely separate these two factors and it is also not necessary. In this paper, we propose a novel model that learns the latent representation without disentanglement and leverages sentiment intensity as input to decoder for fine-grained sentiment control. Moreover, aligned sentences with the same content but different sentiment intensities are usually unavailable. Due to the lack of parallel data, we construct pseudo-parallel sentences (i.e, sentences with similar content but different intensities) to relieve the burden of our model. In specific, motivated by the fact that the sentiment word (e.g., "delicious") has a close relationship with the non-sentiment context word (e.g., "food"), we use dependency parsing to capture the dependency relationship. The pseudo-parallel sentences are produced by replacing the sentiment word with a new one according to the specific context word. Besides, the difference between pseudo-parallel sentences and generated sentences and other constraints are utilized to guide the model precisely revising sentiment. Experiments on the Yelp dataset show that our method substantially improves the degree of content preservation and sentiment accuracy and achieves state-of-the-art performance.

1 Introduction

Text sentiment transfer is a common but difficult style transfer task in Natural Language Processing (NLP). The goal of sentiment transfer is to change the sentiment of a sentence to the opposite while preserving its semantic meaning. Sentiment transfer has obtained board applications in NLP, such as letter and review rewriting [20, 22], which attracts the attention of large numbers of researchers.

Previous works [33, 14] of sentiment transfer mainly focus on binary sentiment (positive and negative) transfer. In this paper, we set our task on more general scenarios that revise sentences on a given sentiment intensity value ranging from 1 to 5 for fine-grained transfer, here the intensity 1 to 5 corresponds to strong negative, weak negative, neutral, weak positive, and strong positive. For example, given the input sentence "the food was **totally fine**" with the sentiment intensity "4", an output "the food was **enough**" may be desired to generate on the target sentiment "3" and "the food was **forgettable**" on the target sentiment "2". Besides, an output sentence "the food was **totally wonderful**" on the target sentiment "5" expresses stronger positive intensity and "the food was **totally terrible**" on the target sentiment "1" has stronger negative intensity. The task of fine-grained text sentiment transfer aims at modifying an input sentence to satisfy a target sentiment intensity while keeping the original content. However, there are some limitations to this task and several problems in previous methods. First of all, there are no natural parallel data, hence we can not use a supervised way to train the transfer model. Second, previous works like [16] attempt to disentangle a sentence into the content part and the sentiment part, but it is difficult to completely separate them because these two parts are mixed together in a complicated way. It usually leads to the semantic meaning of the original sentence and its corresponding generated sentence quite different.

In this paper, we propose an approach for editing sentences which contains two parts: transfer module and pseudo-parallel module. In the transfer module, Gated Recurrent Unit (GRU) based encoder-decoder architecture [2] is employed to revise sentences. The encoder encodes each input sentence into a latent representation without disentanglement, while the decoder generates sentences under the control of sentiment intensity values. We also use a classifier to predict the sentiment value of the generated sentence. The error between the sentiment value of the generated sentence and the target value provides a signal to train the decoder. Due to the lack of parallel data, pseudo-parallel sentences are introduced in the pseudo-parallel module to guide the transfer module to generate sentences on a given sentiment intensity value. In specific, the pseudo-parallel module consists of two parts: dependency parsing and pseudo-parallel production. Pseudo-parallel sentences are pairs of sentences with similar content but different sentiment values. The key issue of producing pseudo-parallel sentences is to accurately find the sentiment information of a source sentence and change it to satisfy the target sentiment. As observed that the sentiment word "delicious" is suitable to describe "food" instead of "staff", and different sentiment words have different sentiment intensity (e.g., "delicious" has a stronger positive sentiment than "ok", "terrible" has a stronger negative sentiment than "so-so"). In the dependency parsing part, we first extract sentiment words of a sentence and then leverage dependency parsing to find the non-sentiment context word that has a specific dependency with the sentiment word. Subsequently, during pseudo-parallel production, all the sentiment words describe the same context word are evaluated by a scorer function and the most appropriate sentiment word is selected to replace the original one, thus we can obtain the pseudo-parallel sentence on a target sentiment. Finally, the reference loss between the generated sentence and pseudo-parallel sentence combined with other constraints such as reconstruction loss is utilized to enhance the

[1] Huazhong University of Science and Technology, China. Email: {xiao_lulu, xiaoye, rxli, panzhou, idcliyuhua}@hust.edu.cn
[2] Fujitsu Laboratories of America, USA, Email: jun.wang@us.fujitsu.com
[*] Corresponding author

ability of our model to modify sentences.

We compare our method with state-of-the-art approaches on the dataset of Yelp reviews. Automatic metrics and human metrics of experiment results show the efficacy of our model.

The contributions of this paper are summarized as the following three points:

1. We propose a novel framework with the combination of a classifier and sentiment controls to modify a sentence, in which the sentiment is not disentangled from sentence.
2. To our best knowledge, this paper is the first work that introduces dependency parsing to the sentiment transfer task. Dependency parsing is used to find context words related to sentiment words and produce pseudo-parallel sentences which provide a signal to the model when revising sentences.
3. Experiment results of automatic evaluation and human evaluation show that our model outperforms state-of-the-art methods on both content preservation and sentiment accuracy.

2 Related Work

Recently, deep learning obtains significant results in various computer vision and natural language processing tasks [36, 23]. The style transfer on computer vision has also achieved exciting performance [9, 27, 35, 15, 12], which inspires researchers to propose the task of style transfer on natural language text. After a surge of researches on this task, text style transfer has obtained significant results [20, 22, 4, 10, 6, 3, 28, 29, 25]. Current methods of text style transfer mainly focus on revising polarity attributes (e.g., sentiment, writing style, gender, etc.) of text to the opposite while preserving attribute-independent content.

Due to the lack of parallel sentences in training time, an unsupervised way was used on existing methods. Some methods follow the adversarial idea of Generative Adversarial Networks (GANs) [7] that optimizes decoder/generator and discriminator/classifier in cycle. Yang et al. [31] use a language model as the discriminator to provide richer and more stable feedback to guide the Variational Autoencoders (VAEs) [13] generating sentences. Fu et al. [5] propose two text style transfer models that employ adversarial training. The encoders of both models extract the content of a sentence under the direction of the classifier, but the first model utilizes a seq2seq [26] with two decoders on different styles, the second model just has one decoder with style embedding. Zhao et al. [34] employ the extension model of adversarial autoencoder (AAE) [18] to generate sentences and apply it to style transfer. Hu et al. [8] combine the VAEs and attribute discriminators to efficiently generate semantic representations with the wake-sleep algorithm. John et al. [11] disentangle style and content latent representations under the multi-task loss and the adversarial loss.

Another line of methods does not implement the adversarial idea. Li et al. [14] obtain the content of a sentence by deleting its sentiment words, and retrieve similar context from the target style corpus to extract the sentiment information, then combine them into the neural network. Zhang et al. [32] leverage shared encoder-decoder model to learn the public attributes (semantic) of all instances, and private encoder-decoder model to learn the specific characteristics of the corresponding attribute corpus. Xu et al. [30] propose a cycled reinforcement learning model which includes the neutralization module and emotionalization module. The neutralization module learns disentangled representations and the emotionalization module adds sentiment to neutralize semantic content.

In contrast, we consider more general scenarios that edit sentences on different sentiment intensity values for fine-grained transfer. There are few works of fine-grained sentiment transfer. Liao et al. [16] propose to learn disentangled content factor and sentiment factor by two separate encoders based on VAE, and then modify the content under the target sentiment. To better disentanglement, they model the content similarity and the sentiment differences of pseudo-parallel sentences. Luo et al. [17] propose a Seq2SentiSeq model combined with the sentiment intensity value and use cycle reinforcement learning method to train the model. Different from them, we employ an autoencoder with sentiment intensity value as control and pseudo-parallel sentences produced by dependency parsing as references to revise sentences.

3 Method

We assume the set of all inputs in our model is $D_v = \{(x_1, v_1), \dots, (x_n, v_n)\}$, where x_i is a sentence, and $v_i \in V$ is the sentiment intensity of x_i. The values of V are fine-grained sentiments ranging from 1 to 5. We define the sentences with sentiment values larger than 3 as positive, equal to 3 as neutral and the rest as negative. The goal of this task is to generate a new sentence y for an input x. The sentiment value of x is v^{src}, $(x, v^{src}) \in D$. The generated sentence y should satisfy the requirement of keeping the content similar to x and its sentiment value is the same as the target sentiment $v^{tgt} \in V$. An overview of our system is depicted in Figure 1. The top part is the dependency parsing module. It employs dependency parsing to find context words that have specific dependencies with sentiment words in sentences. The bottom part is the transfer module. The main framework here is a traditional encoder-decoder network trained with pairs of (x, v^{src}) as input to generate a sentence that minimizes a set of constraints.

3.1 Extraction

To analyze the dependencies between sentiment words and context words, we first need to extract sentiment words that have strong power of sentiment polarity. We just consider to extract sentiment words on sentiment polarity. Assuming all the input sentences on sentiment polarity is $D_r = \{(x_1, r_1), \dots, (x_n, r_n)\}$, $r_i \in \{positive, negative\}$. An input sentence x is composed of N-words $u = \{u_1, \dots, u_i, \dots, u_n\}$, and the sentiment polarity of x is r. The way in Li et al. [14] is adopted to extract sentiment words in x, it computes the relative frequency of u_i as,

$$f(u_i, r) = \frac{(count(u_i, D_r) + \lambda)}{(\sum_{r' \in \{positive, negative\}, r' \neq r} count(u_i, D_{r'})) + \lambda} \tag{1}$$

where $count(u_i, D_r)$ is the times of u_i appears in D_r and λ is the smoothing parameter. If the relative frequency $f(u_i, r)$ of u_i is larger than threshold γ, then u_i is considered as a sentiment word of x. We define $\alpha(x, v^{src})$ to be all of the sentiment words in x.

3.2 Dependency Parsing

After the extraction of sentiment words, we perform dependency parsing in the sentence x to find the context words corresponding to the sentiment words. Dependent syntax expresses the entire sentence structure through the dependencies between each word. These dependencies constitute a dependent syntax tree whose root node is

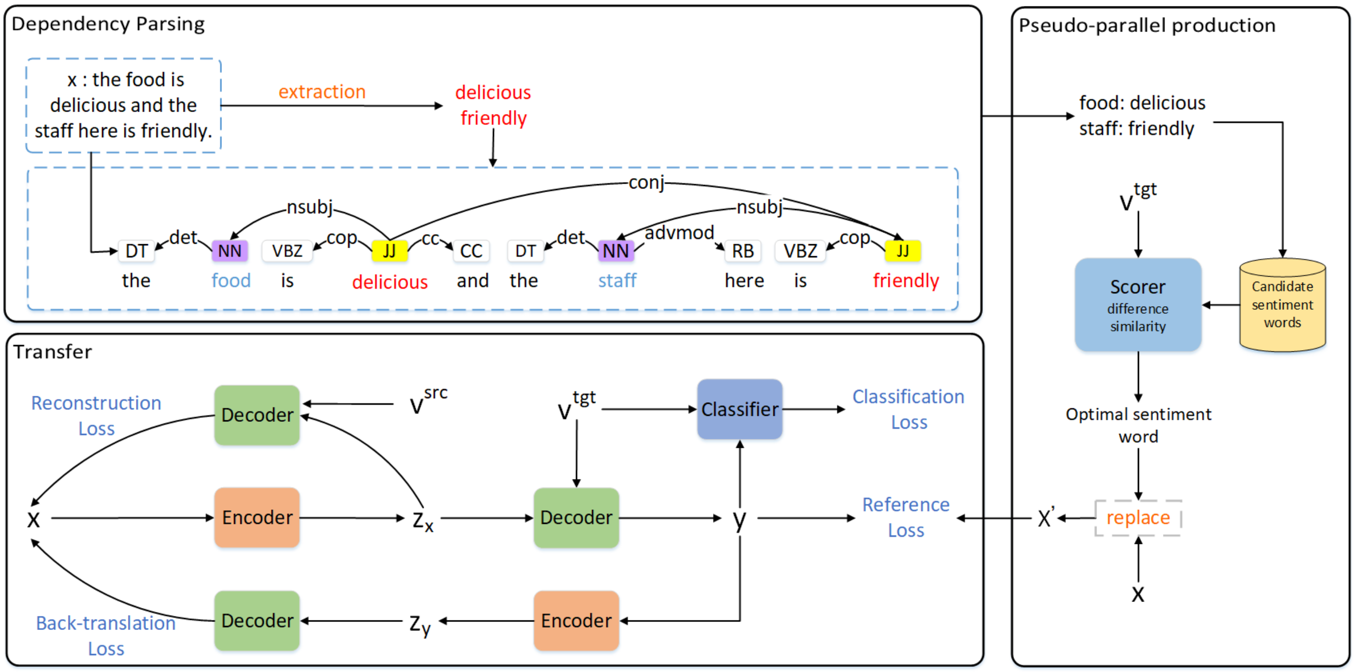

Figure 1. Framework of our proposed method. Our approach contains two parts: transfer module and pseudo-parallel module which consists of dependency parsing and pseudo-parallel production. The dependency parsing contains two steps: (1) extract the sentiment words of an input sentence (2) analyze dependencies between words in the input to find the context words for specific sentiment words. In the pseudo-parallel production, the scorer is used to find the best sentiment words according to the target sentiment to replace the originals, and then we can get the pseudo-parallel sentence $x^{'}$. The bottom part is the transfer module based on an encoder-decoder network. It modifies an input sentence x to a new one y under the target sentiment v^{tgt}.

the core predicate of the sentence. According to the dependencies in the syntax tree, we can find two words with specific grammatical relations in the sentence, which are usually not adjacent. As shown in the top part of Figure 1, each arrow denotes a dependency. The arrow points to the governed object, and the starting point of the arrow is the dependent object. To decide which word has specific dependency with the sentiment word, we just consider several fixed dependencies like nsubj (nominal subject), dobj (direct object), amod (adjectival modifier), etc. For example, the word "food" in sentence "the food tastes delicious" is the word we want to find that has certain dependency with sentiment word "delicious" instead of word "tastes" or others. The word "food" is the nominal subject of the sentiment word "delicious".

Input	the best part, exceptional service and prices can not be beat!
$(x', 1)$	the worst part, dreadful service and prices can not be beat!
$(x', 2)$	the frustrating part, lousy service and prices can not be beat!
$(x', 3)$	the hot part, fine service and prices can not be beat!
$(x', 4)$	the best part, exceptional service and prices can not be beat!
$(x', 5)$	the gorgeous part, wonderful service and prices can not be beat!

Table 1. Five pairs of pseudo-parallel sentences. The first line is the input sentence and other lines are pseudo-parallel sentences on five sentiment intensity values. The forth line is the same as the first line because the sentiment value of the input sentence is 4.

Assuming a is a sentiment word in $x, a \in \alpha(x, v^{src})$, o is declared as the context word of a if o has a certain dependency described above with a. In this part, we do not need to consider sentiment intensity. We extract all the sentiment words in $\alpha(o, r)$ whose context word is o on the $r(r \in (positive, negative))$ corpus. For example, positive sentences "the food tastes delicious" and "the food tastes wonderful" describe the same context word "food". The sentiment words "delicious" and "wonderful" then are saved with "food". Dependencies between sentiment words and corresponding context words are used to assist in producing pseudo-parallel sentences, which will be described in the following sections.

3.3 Replace

Pseudo-parallel sentences are a pair of sentences that have the same semantic content but different sentiment values as shown in Table 1. The way of constructing pseudo-parallel sentences is to replace each sentiment word of source text with another optimal sentiment word. As mentioned above, a sentiment word has close dependency with its context word, so all the sentiment words of the context word can be candidates for replacement. Given an input (x, v^{src}), a is a sentiment word of $x, a \in \alpha(x, v^{src})$, o is the context word of a. There are k candidate words in $\alpha(o, r)$ to replace a. The best candidate (c^{tgt}) which minimizes the score will be used to replace a under the target sentiment v^{tgt}.

$$c^{tgt} = argmin_c\{S(a, c) | c \in candidate_k(a)\} \tag{2}$$

where $S(*)$ is a weighted scorer function and $candidate_k(a)$ is all the candidates of sentiment word a. Function $S(*)$ measures candidates from different aspects, it mainly considers two factors in our setting: (a) how the candidate word satisfies the target sentiment value v^{tgt}, (b) how similar with a the candidate word is. We use two ways to measure them, as follows:

Sentiment difference: Sentiment difference measures the difference between the sentiment value of candidate word (c) and the target value v^{tgt}. How to compute the sentiment value of c is a key problem as it is unknown. Inspired by sentiWordNet [1], we use the average sentiments of texts which contain the word c to represent the sentiment of c.

$$v^c = \frac{\sum_{(x \in D, c \in x)} v^{src}}{num(\sum_{(x \in D, c \in x)} v^{src})} \tag{3}$$

where x is an input, $x \in D, v^{src}$ is the sentiment of $x, num(\sum_{(x \in D, c \in x)} v^{src})$ is the text numbers of c appears in D. Then, sentiment difference is computed as follow,

$$r_d(v^c, v^{tgt}) = |v^c - v^{tgt}| \tag{4}$$

Similarity: Similarity indicates how similar the sentiment word (a) and candidate (c) are. As observed that all candidates can replace a, but some candidates do not match the context. For example, the sentiment word “delicious” on the text “the food is delicious” is more likely to be replaced by “awesome” than “love”. Therefore, we should find a similar word with a to replace according to:

$$r_s(a, c) = wordsim(a, c) \tag{5}$$

where $wordsim(a, c)$ is a cosine similarity based on word embedding between embedding vector of a and c.

The scorer function $S(*)$ is composed of all the measures above:

$$S(a, c) = \beta_d r_d(v^c, v^{tgt}) + \beta_s r_s(a, c) \tag{6}$$

where β_d and β_s are weight parameters. The pseudo-parallel sentences constructing algorithm is shown in Algorithm 1.

Algorithm 1 Pseudo-parallel sentence producing method based on dependency parsing.

```
Input: a sentence $x$ with sentiment label $v^{src}$, the target sentiment $v^{tgt}$, context-sentiment words table $T = \{o_1 : (a_{11}, a_{12}, ...), o_2 : (a_{21}, a_{22}, ...), ...\}$.
Extract sentiment words $A = \{a_1, a_2, ...\}$ in $x$ based on Eq. 1
Analyze dependencies $R = \{(r_1, w_1, w_{11}), (r_2, w_2, w_{21}), ...\}$ between words in $x$
for each $a$ in $A$ do
    Find the non-sentiment word $o$ that has special dependency with $a$ in $R$
    Retrieve $o$ in table $T$ and get all candidate words $C_o = \{c_1, c_2, ...\}$ of $a$
    Update table $T$ with $o$ and $a$
    Compute sentiment value of each word in $C_o$ based on Eq. 3
    Compute sentiment difference between sentiment value of each word in $C_o$ and $v^{tgt}$ based on Eq. 4
    Compute similarity between each word in $C_o$ and $a$ based on Eq. 5
    Use scorer function find the best word $c$ in $C_o$ based on Eq. 6
    Replace $a$ with $c$ to obtain the pseudo-parallel sentence $x^{'}$ whose sentiment is $v^{tgt}$ of $x$
end for
```

3.4 Training

Our model mainly employs the encoder-decoder framework, a natural language text with sentiment label is as an input to the model. The encoder learns to encode the sentence into a hidden representation and the decoder learns to generate sentence under the representation. However, the sentence generated by the decoder is a new text that similar to the input, the decoder is not able to add sentiments to it. Therefore, we apply a sentiment control and some constraints to our model. The sentiment control is an embedding of target sentiment value, it is concatenated with the hidden representation as the input to the decoder. The constraints are a set of losses to enhance the abilities of content preservation and sentiment transfer for the model. We introduce a classifier to predict sentiment values of generated sentences. We denote the encoder-decoder framework by $G = (G_{enc}, G_{dec})$ and the classifier by C. We consider these four types of losses as follows. The reconstruction loss and back-translation loss are employed to preserve the content of the sentences. In addition, to keep content unchanged, the reference loss also helps to revise the sentiments.

Reconstruction loss: Reconstruction loss denotes the error of reconstructing input x. Assuming $z_x = G_{enc}(x)$ is the hidden representation of x. v^{src} is the sentiment value of x. The decoder generates sentence $x \approx P_G(.|z_x, v^{src})$ conditioned on z_x, v^{src}. The reconstruction loss is computed as:

$$L_{rec} = -\log P_G(x|z_x, v^{src}) \tag{7}$$

Back-translation loss: Let $y = G_{dec}(x, v^{tgt})$ be the generating sentence of x on the target sentiment $v^{tgt}, z_y = G_{enc}(y)$ is the hidden representation of y. The decoder generates sentence $x \approx P_G(.|z_y, v^{src})$ conditioned on z_y, v^{src}. Back-translation loss is the error of translating y into x, it is indicated as:

$$L_{bt}(x, v) = -\log P_G(x|z_y, v^{src}) \tag{8}$$

Classification loss: The classifier is used to predict the sentiment value of a text. To ensure the sentiment value of generating sentence y matches the target sentiment v^{tgt}, classification loss is used as a feedback to guide the model. The classifier predicts the sentiment value $v_y = P_C(.|z_y)$ of y.

$$L_c(v^y, v^{tgt}) = -\log P_C(v^{tgt}|z_y) \tag{9}$$

Reference loss: The reference loss is the error of y and the Pseudo-Parallel Sentence x^{tgt} of x on target sentiment.

$$L_r(x, x^{tgt}) = -\log P(x'|y) \tag{10}$$

In training, the classifier is trained with sentences and corresponding sentiment labels as input and predicted sentiment as output. Sentence is first encoded into a latent representation through the Gated Recurrent Unit (GRU) based encoder and then as the input to the traditional multi-classifier. After multiple iterations of batches inputs, the classifier is trained to minimize the loss and then is added to the encoder-decoder framework. The encoder-decoder network is trained with source text x, target sentiment v^{tgt} as input and pseudo-parallel sentence x^{tgt} as reference, and new sentence y as output by minimizing:

$$L = \lambda_1 L_{rec} + \lambda_2 L_{bt} + \lambda_3 L_c + \lambda_4 L_r \tag{11}$$

where λ_1, λ_2, λ_3 and λ_4 are hyper-parameters.

4 Experiments

We perform the experiments on two tasks: fine-grained sentiment transfer and sentiment polarity transfer. On fine-grained sentiment transfer, the sentiment of the source text should be transferred to a target numeric value in 1,2,3,4,5. Sentiment polarity transfer mainly changes the source text to a new sentence with the opposite sentiment (positive or negative). We apply automatic and human evaluations to compare our approach with previous works on these two tasks.

4.1 Dataset

For all the experiments, the dataset we use is the Yelp reviews from Liao et al. [16]. We use a more recent version of Stanford CoreNLP than Liao et al. [16] which leads to a little different distribution, however, our sentiment intensity is more accurate. After processing, our dataset has about 600K sentences in total, among them 50K as the test set, 10K as the validation set and the rest as the training set. The data distribution is shown in Table 2.

sentiment interval	[1,2)	[2,3)	[3,4)	[4,5)
sentence num	34576	233916	166566	169196

Table 2. Numbers of Sentences in each sentiment interval.

4.2 Model Setup

For all tasks, the encoder we use is 2 layers bidirectional GRU with 250 dimensions hidden state. The decoder is also 2 layers of bidirectional GRU with attention mechanism, its dimension of hidden state is set to 500. The output of the encoder also called hidden representation concatenated with the target sentiment embedding is as input to the decoder. The dimensions of sentiment embeddings are 128. Encoder (GRU) with dimension hidden size 200 and MLP with dimension hidden size 100 constitute the classifier. The weights (β_d, β_s) of sentiment difference and similarity are respectively set to 1 and 0.5. For the weights $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ of the four losses, we tune them on the validation data with different values, and finally they are respectively set to 0.7, 0.2, 0.3, 0.7.

4.3 Comparative Methods

We compare our model with two state-of-the-art models, one of which is specifically designed for the task of fine-grained sentiment transfer and the other is mainly for the binary sentiment polarity transfer.

Sequence Editing under Quantifiable Guidance (QuaSE) (Liao et al. [16]): QuaSE first proposes the task of quantifiable sentiment transfer, it uses two encoders to capture content and sentiment, one decoder to generate text satisfied the requirement. To better disentangle the two factors, QuaSE uses pseudo-parallel sentences to enhance the model. In the test stage, QuaSE assumes the sentiment of an input follows a Gaussian distribution, then chooses the best one in the distribution to pass to the decoder. We use QuaSE as the comparative method for the task of fine-grained sentiment transfer and follow the default parameters in their codes.

Text Transfer Text by Cross-Alignment (TCA) (Shen et al. [24]): TCA maps an input sentence to a style-independent content representation and pass it to style-dependent decoders. It employs aligned auto-encoder instead of typical variational autoencoder to obtain two distributed constraints by the cross-aligned way and two discriminators to modify sentences. We use TCA for the sentiment polarity transfer experiment following its suggested parameters.

4.4 Evaluation Metrics

There are many evaluation metrics for the task of sentiment polarity transfer, which can also be used for the task of fine-grained sentiment transfer. Due to the lack of parallel corpora, we choose the opportune metrics for our task, as follows.

BLEU: BLEU[21] was originally used to measure the similarity between machine translation text and reference text. The value of BLEU ranges from 0 to 1, we expand it to 0 to 100 as usually done in previous works. With the appearance of text style transfer, BLEU is also used for this task. But there is no reference text, so we calculate the BLEU value between source text and generation text, which evaluates the content preservation.

Edit Distance: In the fields of information theory, linguistics, and computer science, edit distance is used to measure the similarity of two sequences. In general, the edit distance refers to the minimum number of single-character editing required to convert one word w_1 to another word w_2.

MAE: MAE, also known as Mean Absolute Error. In this task, we use MAE to measure the mean error between the target sentiment value and the sentiment of generation sentence.

$$MAE = \frac{1}{|s|} \sum_{x_i \in s} |v_i - v^{tgt}| \tag{12}$$

where s is the set of generated sentences, v_i is the sentiment value of sentence $x_i \in s$ predicted by Stanford CoreNLP (Manning et al. [19]).

4.5 Automatic Evaluation

In the fine-grained sentiment transfer experiment, our model is compared with QuaSE. Each input sentence is required to be converted to five sentences whose sentiment values respectively satisfy the target values 1,2,3,4 and 5. The training data in QuaSE is specially processed, so QuaSE still uses its own training data, and its test data is the same as ours. We perform the MAE evaluation between the target sentiment and the sentiment intensity of the generation sentence, and evaluate the edit distance and BLEU between the generation sentence and the input sentence. The results are shown in Table 3 and all the results are the average values for the whole dataset. “Original” refers to use original sentences to compute the evaluation metrics.

The MAE values of our model and QuaSE are smaller than “Original”, it demonstrates that we both have the ability to revise sentiments of texts. Moreover, the MAE values on the five sentiment intensity values of our model are smaller than QuaSE, the main reason is that we use the error between the pseudo-parallel sentences and the generated sentences and the classifier to provide effect and richer feedback to the decoder. The feedback guides the model to better generate sentences that satisfy a target sentiment. In contrast, QuaSE employs a Gaussian distribution on sentiment factor, which is not so precise. Besides, all the edit distances and the BLEU values of our model are better than QuaSE. QuaSE respectively learns content and sentiment representation disentangled from an input sentence, but it is hard to completely separate them and may cause partial loss of content. However, we do not learn the disentangled representation but apply some constraints to keep content unchanged.

Models	MAE					Edit Distance					BLEU				
	T=1	T=2	T=3	T=4	T=5	T=1	T=2	T=3	T=4	T=5	T=1	T=2	T=3	T=4	T=5
Original	2.13	1.15	0.81	1.00	1.87	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
QuaSE	1.29	0.57	0.77	0.67	1.19	11.88	8.78	8.36	8.13	11.58	6.26	24.55	24.63	30.21	8.23
Our Model	0.47	0.38	0.23	0.41	0.56	6.88	6.58	6.30	5.49	7.24	18.93	30.73	25.96	31.86	26.42

Table 3. Automatic evaluation results of fine-grained sentiment transfer experiment. T refers to the target sentiment. MAE measures mean error between the target sentiment and the sentiment of the output. BLEU and Edit distance measure content similarity between the output and the source sentence.

In the sentiment polarity transfer experiments, QuaSE and TCA are used as the comparison models. We use Sentiment accuracy and BLEU as the measurement metrics. As mentioned in section 3, we define the sentences with sentiment values larger than 3 are positive, smaller than 3 are negative. The results are shown in Table 4, we perform a sentiment accuracy value on transferring negative to positive, and vice versa. The accuracy values in both directions and the BLEU value of our model are larger than TCA. Moreover, our model has a smaller accuracy value on negative to positive but larger accuracy value on positive to negative compared to QuaSE. In general, our model achieves the best average accuracy value and BLEU value. It demonstrates that our model can better revise sentiments of sentences while preserving more content.

	Neg. to Pos.	Pos. to Neg.	Avg. accuracy	BLEU
TCA	73.80%	69.12%	71.46%	13.55
QuaSE	**89.81%**	76.93%	83.36%	9.18
Our model	85.37%	**83.36%**	**84.36%**	**29.21**

Table 4. Automatic evaluation results of sentiment polarity transfer experiment.

4.6 Human Evaluation

In this part, we hire three workers to manually evaluate the quality of 200 generated sentences that are randomly picked from each of our model and the two competitive models. We use the "content preservation" to measure the content integrity of sentences and "fluency" to measure grammatical fluency of sentences. The scores of "content preservation" range from 0 to 3 (score 0 means not preserved, 1 means little preserved, 2 means partially preserved, 3 means fully preserved), the scores of "fluency" range from 1 to 4 (score 1 means poor grammar, 4 means fluent grammar).

The result is shown in Table 5. For the "content preservation" metric, our model achieves the highest score, the main reason is that our model does not learn the disentangled latent representation as used in QuaSE since the disentangled representation misses some content information more or less. For the "fluency" metric, the score of our model is also better than QuaSE and TCA. It may comes from that our pseudo-parallel sentences keep the most grammatical structure of the original sentence. This feature also devotes to generating fluent sentences.

	Content Preservation (Range:[0,3])	Fluency (Range:[1,4])
TCA	1.41	2.58
QuaSE	1.37	2.14
Our model	**1.86**	**2.88**

Table 5. Human evaluation results for three models on content preservation and fluency.

4.7 Case Study

To directly present the effects of our model on fine-grained sentiment transfer, some examples generated by our model are displayed in Table 6. Each sentence is revised to five sentences, and the sentiment values of them are in 1,2,3,4,5. The generated sentence on T=2 in the first example is the same as the input sentence due to the original sentiment label is 2. Similarly, the second example on T=3 and the third example on T=4 are the same as the input sentences. For the first example, when T=1, "sloppy" and "over-priced" are changed to more negative phrase "worst" and the generated sentence on T=3 expresses neutral sentiment. Moreover, when T=4 and 5, the original sentence is revised to positive sentences that opposite to the input and "wonderful", "actually excellent" on T=5 express strong positive sentiment. For the second example, the input sentence is a neutral sentence that describes "seafood". When T=1 and 2, the original sentence is revised to express negative sentiment and the generated sentences on T=4 and 5 express positive sentiments. Although, the generated sentences on T=1, 2 and 5 do not describe "seafood", they describe "cake", "beef" and "steak" that are similar to "seafood". These indicate that our model is able to preserve most of the content and revise the words which have the strong polarity of sentiment in a sentence. In some examples, like the third example on T=2 and T=5, there have some problems of unacceptable sentences and duplicates in word-level, it reminds that we need to reduce this problem.

4.8 Ablation Study

We introduce a classifier and the other three constraints to guide the encoder-decoder to modify sentences . To show the effects of the three losses, we perform ablation study under the MAE and BLEU metrics. We remove the three losses separately and keep the others unchanged. The results are shown in Table 7 and Table 8. The first line in each table is the sentiment intensity value. In the experiments, we just consider the values of 1, 3 and 5. The second line in each table shows the MAE/BLEU values of QuaSE in table 3 that are used for comparison. The following three lines show the MAE/BLEU values under the omission of reference loss, reconstruction loss and back-translation loss. The last line shows the MAE/BLEU values of all the losses.

	Generated Sentence
E.g. 1	the burger was sloppy and the food was over-priced. (input sentiment is 2)
T=1	the burger was worst and the sauce food was worst!
T=2	the burger was sloppy and the food was over-priced.
T=3	the burger was extra and let packaged extra dogs they receive dogs.
T=4	the burger was phenomenal and prompt service over it.
T=5	the burger was wonderful and the food was actually excellent.
E.g. 2	it was appropriately spicy, flavorful, and the seafood was not overcooked. (input sentiment is 3)
T=1	it was flavorless spicy, flavorful, and the cake was worst breakfast.
T=2	it was pretty bland spicy, especially, the beef was better!
T=3	it was appropriately spicy, flavorful, and the seafood was not overcooked.
T=4	it was great spicy, flavorful, and the great seafood, and great tasting.
T=5	it was wonderful spicy, wonderful, and the wonderful steak.
E.g. 3	moist bread, fresh ingredients, great flavor. (input sentiment is 4)
T=1	waste mix, waste ingredients, waste flavor.
T=2	bland, the bland ingredients, lousy flavor!
T=3	had bread, had plenty of flavor.
T=4	moist bread, fresh ingredients, great flavor.
T=5	wonderful & fresh ingredients, ingredients, great flavor.

Table 6. Sentences examples generated by our model on each target sentiment.

According to the result in Table 8, the MAE values of “None” are smaller than “Original” in Table 3. It demonstrates that the decoder with the assist of the classifier is able to revise the sentiment intensity of sentences. The MAE values of removing each loss are smaller than QuaSE, this means each loss we add to the model makes a contribution to revise sentiments. The average improvements in removing each loss compared to “None” are 26.67%, 13.66%, and 19%. The average decreases in removing each loss compared to “ALL” are 31.33%, 44.33%, and 39%. These demonstrate that each loss makes a certain contribution to sentiment modification. In Table 7, the average decreases in removing each loss compared to “ALL” are 57.67%, 21.13%, and 22.40%. It shows that each loss is helpful for content preservation especially the reference loss. Moreover, the MAE and BLEU values of “ALL” are the best in all the sentiment values. It shows that the combination of the three losses is effective to enhance the ability to modify sentiments of our model.

	T=1	T=3	T=5
QuaSE	6.26	24.63	8.23
None	8.11	10.03	12.32
L_{rec}, L_{bt}	12.04	20.17	21.80
L_r, L_{bt}	15.68	23.71	25.58
L_r, L_{rec}	19.21	24.28	21.10
ALL	18.93	25.96	26.42

Table 7. Ablation study on BLEU metric.

5 Conclusions

In this work, we focus on the task of fine-grained sentiment transfer that requires to edit sentence on given numeric sentiment values while keeping content unchanged. We propose a novel method based on dependency parsing without learning disentangled representation as usually worked in the previous works. We produce pseudo-parallel sentences through dependency parsing and employ a set of losses to give richer signals to enhance the model. Automatic and human evaluations of experiments on the Yelp reviews demonstrate that our model substantially outperforms the compared models. In the future, we intend to expand our work on more attributes not only sentiment and long text transfer.

	T=1	T=3	T=5
QuaSE	1.29	0.77	1.19
None	1.38	0.53	1.09
L_{rec}, L_{bt}	0.77	0.44	0.99
L_r, L_{bt}	1.26	0.41	0.92
L_r, L_{rec}	1.07	0.32	1.04
ALL	0.47	0.23	0.56

Table 8. Ablation study on MAE metric.

ACKNOWLEDGEMENTS

This work is supported by the National Key Research and Development Program of China under grants 2016YFB0800402 and 2016QY01W0202, National Natural Science Foundation of China under grants U1836204, U1936108, 61572221, 61433006, U1401258, 61572222 and 61502185, and Major Projects of the National Social Science Foundation under grant 16ZDA092.

REFERENCES

[1] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani, 'Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining.'.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, 'Neural machine translation by jointly learning to align and translate', *CoRR*, **abs/1409.0473**, (2014).

[3] Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang, 'Style transformer: Unpaired text style transfer without disentangled latent representation', in *ACL*, (2019).

[4] Cícero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi, 'Fighting offensive language on social media with unsupervised text style transfer', in *ACL*, (2018).

[5] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan, 'Style transfer in text: Exploration and evaluation', *ArXiv*, **abs/1711.06861**, (2017).

[6] Hongyu Gong, Suma Bhat, Lingfei Wu, Jinjun Xiong, and Wen-Mei Hwu, 'Reinforcement learning based text style transfer without parallel training corpus', in *NAACL-HLT*, (2019).

[7] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio, 'Generative adversarial nets', in *NIPS*, (2014).

[8] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing, 'Toward controlled generation of text', in *ICML*, (2017).

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros, 'Image-to-image translation with conditional adversarial networks', *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5967–5976, (2016).

[10] Parag Jain, Abhijit Mishra, Amar Prakash Azad, and Karthik Sankaranarayanan, 'Unsupervised controllable text formalization', in *AAAI*, (2018).

[11] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova, 'Disentangled representation learning for non-parallel text style transfer', in *ACL*, (2018).

[12] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim, 'Learning to discover cross-domain relations with generative adversarial networks', in *ICML*, (2017).

[13] Diederik P. Kingma and Max Welling, 'Auto-encoding variational bayes', *CoRR*, **abs/1312.6114**, (2013).

[14] Juncen Li, Robin Jia, He He, and Percy Liang, 'Delete, retrieve, generate: a simple approach to sentiment and style transfer', in *NAACL-HLT*, (2018).

[15] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou, 'Demystifying neural style transfer', in *IJCAI*, (2017).

[16] Yi Liao, Lidong Bing, Piji Li, Shuming Shi, Wai Lam, and Tong Zhang, 'Quase: Sequence editing under quantifiable guidance', in *EMNLP*, (2018).

[17] Fuli Luo, Peng Li, Pengcheng Yang, Jie Zhou, Yutong Tan, Baobao Chang, Zhifang Sui, and Xu Sun, 'Towards fine-grained text sentiment transfer', in *ACL*, (2019).

[18] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian J. Goodfellow, 'Adversarial autoencoders', *ArXiv*, **abs/1511.05644**, (2015).

[19] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky, 'The stanford corenlp natural language processing toolkit', in *ACL*, (2014).

[20] Igor Melnyk, Cícero Nogueira dos Santos, Kahini Wadhawan, Inkit Padhi, and Abhishek Kumar, 'Improved neural text attribute transfer with non-parallel data', *ArXiv*, **abs/1711.09395**, (2017).

[21] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, 'Bleu: a method for automatic evaluation of machine translation', in *ACL*, (2001).

[22] Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W. Black, 'Style transfer through back-translation', in *ACL*, (2018).

[23] Xiaoye Qu, Zhikang Zou, Yu Cheng, Yang Yang, and Pan Zhou, 'Adversarial category alignment network for cross-domain sentiment classification', in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2496–2508, (2019).

[24] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola, 'Style transfer from non-parallel text by cross-alignment', in *NIPS*, (2017).

[25] Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran, 'Transforming delete, retrieve, generate approach for controlled text style transfer', *ArXiv*, **abs/1908.09368**, (2019).

[26] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, 'Sequence to sequence learning with neural networks', in *NIPS*, (2014).

[27] Yaniv Taigman, Adam Polyak, and Lior Wolf, 'Unsupervised cross-domain image generation', *ArXiv*, **abs/1611.02200**, (2016).

[28] Chen Wu, Xuancheng Ren, Fuli Luo, and Xu Sun, 'A hierarchical reinforced sequence operation method for unsupervised text style transfer', in *ACL*, (2019).

[29] Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu, 'Mask and infill: Applying masked language model for sentiment transfer', *ArXiv*, **abs/1908.08039**, (2019).

[30] Jingjing Xu, Xu Sun, Qi Zeng, Xuancheng Ren, Xiaodong Zhang, Houfeng Wang, and Wenjie Li, 'Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach', in *ACL*, (2018).

[31] Zichao Yang, Zhiting Hu, Chris Dyer, Eric P. Xing, and Taylor Berg-Kirkpatrick, 'Unsupervised text style transfer using language models as discriminators', in *NeurIPS*, (2018).

[32] Ye Zhang, Nan Ding, and Radu Soricut, 'Shaped: Shared-private encoder-decoder for text style adaptation', in *NAACL-HLT*, (2018).

[33] Yi Zhang, Jingjing Xu, Pengcheng Yang, and Xu Sun, 'Learning sentiment memories for sentiment modification without parallel data', in *EMNLP*, (2018).

[34] Junbo Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, and Yann LeCun, 'Adversarially regularized autoencoders', in *ICML*, (2017).

[35] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros, 'Unpaired image-to-image translation using cycle-consistent adversarial networks', *2017 IEEE International Conference on Computer Vision (ICCV)*, 2242–2251, (2017).

[36] Zhikang Zou, Xinxing Su, Xiaoye Qu, and Pan Zhou, 'Da-net: Learning the fine-grained density distribution with deformation aggregation network', *IEEE Access*, **6**, 60745–60756, (2018).