# Towards Fine-grained Text Sentiment Transfer

**Fuli Luo[1], Peng Li[2], Pengcheng Yang[1], Jie Zhou[2],**
**Yutong Tan[3], Baobao Chang[1,4], Zhifang Sui[1,4], Xu Sun[1]**
[1]Key Lab of Computational Linguistics, Peking University
[2]Pattern Recognition Center, WeChat AI, Tencent Inc, China
[3]Computer Science and Technology, Beijing Normal University
[4]Peng Cheng Laboratory, China
luofuli@pku.edu.cn, patrickpli@tencent.com, yang_pc@pku.edu.cn,
withtomzhou@tencent.com, tanyt@mail.bnu.edu.cn, {chbb,szf,xusun}@pku.edu.cn

## Abstract

In this paper, we focus on the task of fine-grained text sentiment transfer (FGST). This task aims to revise an input sequence to satisfy a given sentiment intensity, while preserving the original semantic content. Different from conventional sentiment transfer task that only reverses the sentiment polarity (positive/negative) of text, the FTST task requires more nuanced and fine-grained control of sentiment. To remedy this, we propose a novel Seq2SentiSeq model. Specifically, the numeric sentiment intensity value is incorporated into the decoder via a Gaussian kernel layer to finely control the sentiment intensity of the output. Moreover, to tackle the problem of lacking parallel data, we propose a cycle reinforcement learning algorithm to guide the model training. In this framework, the elaborately designed rewards can balance both sentiment transformation and content preservation, while not requiring any ground truth output. Experimental results show that our approach can outperform existing methods by a large margin in both automatic evaluation and human evaluation. Our code and data, including outputs of all baselines and our model are available at `https://github.com/luofuli/Fine-grained-Sentiment-Transfer`.[1]

## 1 Introduction

Text sentiment transfer aims to rephrase the input to satisfy a given sentiment label (value) while preserving its original semantic content. It facilitates various NLP applications, such as automatically converting the attitude of review and fighting against offensive language in social media (dos Santos et al., 2018).

Previous work (Shen et al., 2017; Li et al., 2018; Luo et al., 2019) on text sentiment transfer mainly focuses on the coarse-grained level: the reversal of positive and negative sentiment polarity. They are confined to scenarios where there are two discrete sentiment labels. To achieve more nuanced and precise sentiment control of text generation, we turn to fine-grained text sentiment transfer (FTST) which revises a sequence to satisfy a given *sentiment intensity*[2], while keeping the semantic content unchanged. Taking Figure 1 as an example, given the same input and five sentiment intensity values ranging from 0 (most negative) to 1 (most positive), the system generates five different outputs that satisfy the corresponding sentiment intensity in a relative order.

| **Input Sentence** | |
|---|---|
| Tasty food and wonderful service. | |
| **Target Sentiment** | **Output Sentence** |
| **0.1** | Horrible food and terrible service! |
| **0.3** | Plain food, slow service. |
| **0.5** | Food and service need improvement. |
| **0.7** | Good food and service. |
| **0.9** | Amazing food and perfect service!! |

Figure 1: An example of the input and output of the fine-grained text sentiment transfer task. The output reviews describe the same content (e.g. *food/service*) as the input while expressing different sentiment intensity.

There are two main challenges of FTST task. First, it is tough to achieve fine-grained control of the sentiment intensity when generating sentence. Previous work about coarse-grained text sentiment transfer usually uses a separate decoder for each sentiment label (Xu et al., 2018; Zhang et al., 2018b) or embeds each sentiment label into a separate vector (Fu et al., 2018; Li et al., 2018). However, these methods are not feasible for fine-grained text sentiment transfer since the

[1]Joint work between WeChat AI and Peking University.

[2]The sentiment intensity is a real-valued score between 0 and 1, following sentiment intensity prediction task in sentiment analysis (Zhang et al., 2017; Mohammad et al., 2018).

target sentiment intensity value is a real value, other than discrete labels. Second, parallel data[3] is unavailable in practice. In other words, we can only access the corpora which are labeled with fine-grained sentiment ratings or intensity values. Therefore, in the FTST task, we can not train a generative model via ground truth outputs.

To tackle the two challenges mentioned above, we propose two corresponding solutions. First, in order to control the sentiment intensity of the generated sentence, we propose a novel sentiment intensity controlled sequence-to-sequence (Seq2Seq) model Seq2SentiSeq. It incorporates the sentiment intensity value into the conventional Seq2Seq model via a Gaussian kernel layer. By this means, the model can encourage the generation of words whose sentiment intensity closer to the given intensity value during decoding. Second, due to the lack of parallel data, we can not directly train the proposed model via MLE (maximum likelihood estimation). Therefore, we propose a cycle reinforcement learning algorithm to guide the model training without any parallel data. The designed reward can balance both sentiment transformation and content preservation, while not requiring any ground truth output.

Evaluation of the FTST task is also challenging and complex. In order to build a reliable automatic evaluation, we collect human references for FTST task on the Yelp review dataset[4] via crowdsourcing and design a series of automatic metrics.

The main contributions of this work are summarized as follows:

- We propose a sentiment intensity controlled generative model **Seq2SentiSeq**, in which a sentiment intensity value is introduced via a Gaussian kernel layer to achieve fine-grained sentiment control of the generated sentence.
- In order to adapt to non-parallel data, we design a cycle reinforcement learning algorithm **CycleRL** to guide the model training in an unsupervised way.
- Experiments show that the proposed approach can largely outperform state-of-the-art systems in both automatic evaluation and human evaluation.

[3]Parallel data in this paper denotes the corpus where each pair of sentences describes the same content while expressing the different sentiment intensity.

[4]https://www.yelp.com/dataset

## 2 Proposed Model

### 2.1 Task Definition

Given an input sequence $\boldsymbol{x}$ and a target sentiment intensity value $v_y$, the FTST task aims to generate a sequence $\boldsymbol{y}$ which not only expresses the target sentiment intensity $v_y$, but also preserve the original semantic content of the input $\boldsymbol{x}$. Without loss of generality, we limit the sentiment intensity value $v_y$ ranging from 0 (most negative) to 1 (most positive).

### 2.2 Seq2SentiSeq: Sentiment Intensity Controlled Seq2Seq Model

Figure 2 presents a sketch of the proposed Seq2SentiSeq model. The model is based on the encoder-decoder framework, which takes a source text $\boldsymbol{x}$ as the input and outputs a target sentence $\boldsymbol{y}$ with the given sentiment intensity $v_y$. In order to control the sentiment intensity of $\boldsymbol{y}$, we introduce a Gaussian kernel layer into the decoder.

#### 2.2.1 Encoder

We use a bidirectional RNN as the encoder to capture source content information. Each word in the source sequence $\boldsymbol{x} = (x_1, \cdots, x_m)$ is firstly represented by its semantic representation mapped by semantic embedding $\boldsymbol{E}_c$. The RNN reads the semantic representations from both directions and computes the forward hidden states $\{\overrightarrow{h}_i\}_{i=1}^m$ and backward hidden states $\{\overleftarrow{h}_i\}_{i=1}^m$ for each word. We obtain the final hidden representation of the $i$-th word by concatenating the hidden states from both directions $h_i = [\overrightarrow{h}_i; \overleftarrow{h}_i]$.

#### 2.2.2 Decoder

Given the hidden representations $\{h_i\}_{i=1}^m$ of the input sequence $\boldsymbol{x}$ and the target sentiment intensity value $v_y$, the decoder aims to generate a sequence $\boldsymbol{y}$ which not only describes the same content as the input sequence $\boldsymbol{x}$, but also expresses a close sentiment intensity to $v_y$.

In order to achieve the aim of controlling sentiment during decoding, we firstly embedded each word with an additional sentiment representation, besides the original semantic representation. The semantic representation characterizes the semantic content of the word, while the sentiment representation characterizes its sentiment intensity. Formally, the hidden state $s_t$ of the decoder at timestep $t$ is computed as follows:

$$s_t = f\left(s_{t-1}, [\boldsymbol{E}_c(y_{t-1}); \boldsymbol{E}_s(y_{t-1})], c_t\right) \qquad (1)$$

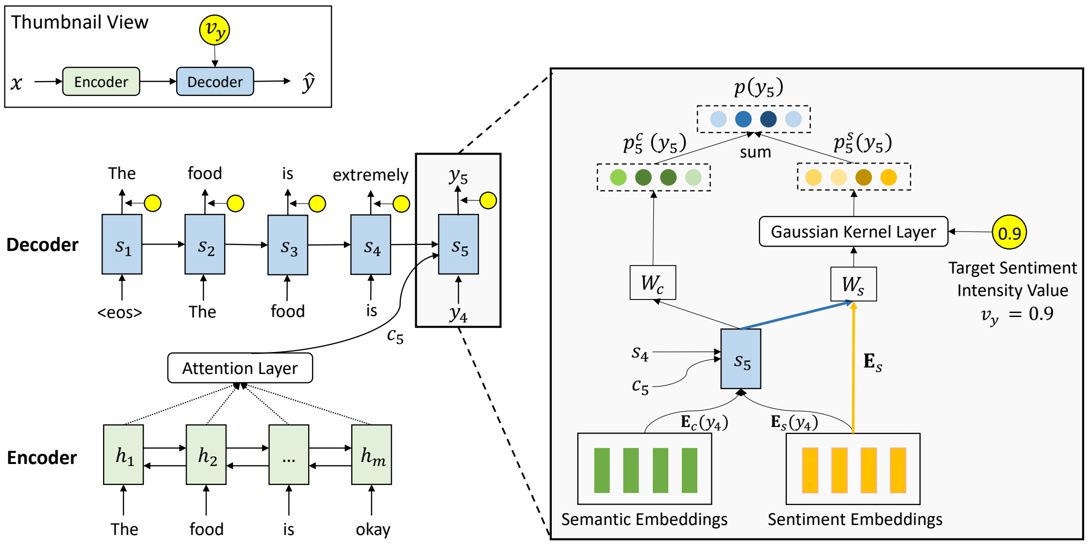

Figure 2: The proposed sequence to sentiment controlled sequence (Seq2SentiSeq) model.

where $\boldsymbol{E}_s(y_{t-1})$ refers to the sentiment representation of the word $y_{t-1}$ mapped by the sentiment embedding matrix $\boldsymbol{E}_s$, $\boldsymbol{E}_c(y_{t-1})$ is the semantic representation, and the context vector $c_t$ is computed by an attention mechanism in the same way as Luong et al. (2015).

Considering two goals of the FTST task: sentiment transformation and content preservation, we model the final generation probability into a mixture of semantic probability and sentiment probability, where the former evaluates content preservation and the latter measures sentiment transformation. Similar to the traditional Seq2Seq model (Bahdanau et al., 2014), the semantic probability distribution over the whole vocabulary is computed as follows:

$$p_t^c = \text{softmax}(\boldsymbol{W}_c s_t) \tag{2}$$

where $\boldsymbol{W}_c$ is a trainable weight matrix.

The sentiment probability measures how close the sentiment intensity of the generated sequence to the target $v_y$. Normally, each word has a specific sentiment intensity. For example, the word *"okay"* has a positive intensity around 0.6, *"good"* is around 0.7, and *"great"* is around 0.8. However, when involving to the previous generated words, the sentiment intensity of current generated word may be totally different. For example, the phrase *"not good"* has a negative intensity around 0.3, while *"extremely good"* is around 0.9. That is to say, the sentiment intensity of each word at time-step $t$ should be decided by both the sentiment representation $\boldsymbol{E}_s$ and the current decoder state $s_t$. Therefore, we define a sentiment intensity prediction function $g(\boldsymbol{E}_s, s_t)$ as follows:

$$g(\boldsymbol{E}_s, s_t) = \text{sigmoid}(\boldsymbol{E}_s \boldsymbol{W}_s s_t) \tag{3}$$

where $\boldsymbol{W}_s$ is a trainable parameter, and sigmoid is used to scale the predicted intensity value to [0, 1].

Intuitively, in order to achieve fine-grained control of sentiment, words whose sentiment intensities are closer to the target sentiment intensity value $v_y$ should be assigned a higher probability. Take Figure 2 as an example, at the 5-th time-step, word *"good"* should be assigned a higher probability than word *"bad"*, thus the predicted intensity value $g(\textit{"good"}, s_4)$ is closer to the target sentiment intensity than $g(\textit{"bad"}, s_4)$. To favor words whose sentiment intensity is near $v_y$, we introduce a Gaussian kernel layer which places a Gaussian distribution centered around $v_y$, inspired by Luong et al. (2015) and Zhang et al. (2018a). Specifically, the sentiment probability is formulated as:

$$o_t^s = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(g(\boldsymbol{E}_s, s_t) - v_y\right)^2}{2\sigma^2}\right) \tag{4}$$

$$p_t^s = \text{softmax}(o_t^s) \tag{5}$$

where $\sigma$ is the standard deviation.

To balance both sentiment transformation and content preservation, the final probability distribution $p_t$ over the entire vocabulary is defined as a mixture of two probability distributions:

$$p_t = \gamma p_t^s + (1-\gamma) p_t^c \tag{6}$$

where $\gamma$ is the hyper-parameter that controls the trade-off between two generation probabilities.

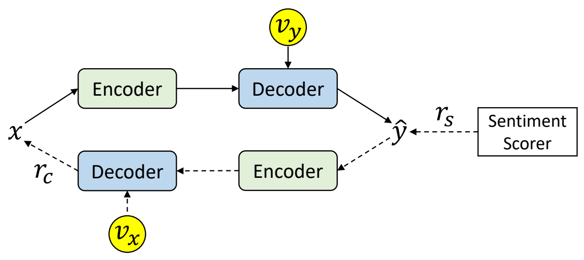

Figure 3: Cycle reinforcement learning. Note that the upper encoder-decoder model and the lower encoder-decoder are just one Seq2SentiSeq model.

## 2.3 Training: Cycle Reinforcement Learning

A serious challenge of the FTST task is the lack of parallel data. Since the ground truth output $\boldsymbol{y}$ is unobserved, we can not directly use the maximum likelihood estimation (MLE) for training. To remedy this, we design a cycle reinforcement learning (CycleRL) algorithm. An overview of the training process is summarized in Algorithm 1. Two rewards are designed to encourage changing sentiment but preserving content, without the need of parallel data. The definitions of the two rewards and the corresponding gradients for Seq2SentiSeq model $S$ are introduced as follows.

### 2.3.1 Reward Design

We design the respective rewards for two goals (sentiment transformation and content preservation) of the FTST task. Then, an overall reward $r$ is calculated to balance these two goals and guide the model training.

**Reward for sentiment transformation.** A pre-trained sentiment scorer is used to evaluate how well the sampled sentence $\hat{\boldsymbol{y}}$ matches the target sentiment intensity value $v_y$. Specifically, the reward for sentiment transformation is formulated as:

$$r_s = 1/(|v_y - \varphi(\hat{\boldsymbol{y}})| + 1) \tag{7}$$

where $\varphi$ refers to the pre-trained sentiment scorer which is implemented as LSTM-based linear regression model.

**Reward for content preservation.** Intuitively, if the model performs well in content preservation, it is easy to back-reconstruct the source input $\boldsymbol{x}$. Therefore, we design the reward for content preservation to be the probability of the model reconstructing $\boldsymbol{x}$ based on the generated text $\hat{\boldsymbol{y}}$ and the source sentiment intensity value $v_x$.

$$r_c = p(\boldsymbol{x}|\hat{\boldsymbol{y}}, v_x; \theta) \tag{8}$$

where $\theta$ is the parameter of Seq2SentiSeq model.

**Algorithm 1** The cycle reinforcement learning algorithm for training Seq2SentiSeq.

```
Input: A corpora $\mathcal{D} = \{(x_{i,i})\}$ where each sequence $x_i$ is
    labeled with a fine-grained sentiment label $v_i$
 1: Initial the pseudo-parallel data $\mathcal{V}_0 = \{(x_i, \hat{y_i})\}$
 2: Pre-train Seq2SentiSeq model $S_\theta$ using $\mathcal{V}_0$
 3: for each iteration $t = 1, 2, ..., T$ do
 4:     Sample a sentence $x$ from $\mathcal{D}$
 5:     for $k = 1, 2, ..., K$ do
 6:         Sample a intensity value $v_y^{(k)}$ from interval $[0, 1]$
 7:         Generate a target sequence: $\hat{\boldsymbol{y}}^{(k)} = S(\boldsymbol{x}, v_y^{(k)}; \theta)$
 8:         Compute sentiment reward $r_s^{(k)}$ based on Eq. 7
 9:         Compute content reward $r_c^{(k)}$ based on Eq. 8
10:         Compute total reward $r^{(k)}$ based on Eq. 9
11:     end for
12:     Update $\theta$ using reward $\{r^{(k)}\}_{k=1}^K$ based on Eq. 11
13:     Update $\theta$ using cycle reconstruction loss in Eq. 12
14: end for
```

**Overall reward.** To encourage the model to improve both sentiment transformation and content preservation, the final reward $r$ guiding the model training is designed to be the harmonic mean of the above two rewards:

$$r = (1 + \beta^2) \frac{r_c \cdot r_s}{(\beta^2 \cdot r_c) + r_s} \tag{9}$$

where $\beta$ is a harmonic weight that controls the trade-off between two rewards.

### 2.3.2 Optimization

The goal of RL training is to minimize the negative expected reward,

$$\mathcal{L}(\theta) = -\sum_k r^{(k)} p_\theta(\hat{\boldsymbol{y}}^{(k)}|\boldsymbol{x}) \tag{10}$$

where $\hat{\boldsymbol{y}}^{(k)}$ is the $k$-th sampled sequence according to probability distribution $p$ in Eq. 6, $r^{(k)}$ is the reward of $\hat{\boldsymbol{y}}^{(k)}$, and $\theta$ is the parameter of the proposed model in Figure 2.

By means of policy gradient method (Williams, 1992), for each training example, the expected gradient of Eq. 10 can be approximated as:

$$\nabla_\theta \mathcal{L}(\theta) \simeq -\frac{1}{K} \sum_{k=1}^{K} \left(r^{(k)} - b\right) \nabla_\theta \log\left(p_\theta(\hat{\boldsymbol{y}}^{(k)})\right) \tag{11}$$

where $K$ is the sample size and $b$ is the greedy search decoding baseline that aims to reduce the variance of gradient estimate which is implemented in the same way as Paulus et al. (2017).

Nevertheless, RL training strives to optimize a specific metric which may not guarantee the fluency of the generated text (Paulus et al., 2017), and usually faces the unstable training problems (Li et al., 2017). The most direct way is to expose the sentences which are from the training corpus to the decoder and trained via MLE (also called teacher-forcing). In order to expose the decoder to the original sentence from the training corpus, we borrow ideas from *back-translation* (Lample et al., 2018a,b). Specifically, the model first generates a sequence $\hat{\boldsymbol{y}}$ based on the input text $\boldsymbol{x}$ and the target sentiment intensity value $v_y$, and then reconstructs the source input $\boldsymbol{x}$ based on $\hat{\boldsymbol{y}}$ and the source sentiment intensity value $v_x$. Therefore, the gradient of the cycle reconstruction loss is defined as:

$$\nabla_\theta \mathcal{J}(\theta) = \nabla_\theta \log\Big(p\big(\boldsymbol{x}|S(\boldsymbol{x}, v_y; \theta), v_x; \theta\big)\Big) \tag{12}$$

where $S$ refers to the Seq2SeniSeq model.

Finally, we alternately update the model parameters $\theta$ based on Eq. 11 and Eq. 12.

## 3 Experimental Setup

In this section, we introduce the dataset, experiment settings, baselines, and evaluation metrics.

### 3.1 Dataset

We conduct experiments on the Yelp dataset[5], which consists of a large number of product reviews. Each review is assigned a sentiment rating ranging from 1 to 5. Since the label inconsistency between human is more serious in fine-grained ratings, we average the ratings for the sentences which have a Jaccard Similarity more than 0.9. Then, averaged ratings are normalized between 0 and 1 as the sentiment intensity. Other data preprocessing is the same as Shen et al. (2017). Finally, we obtain a total of 640K sentences. We randomly hold 630K for training, 10K for validation, and 500 for testing. Even though the sentiment intensity distribution of training dataset is not uniform, the proposed framework consists of a uniform data augmentation which generates sentences whose intensity is from interval [0, 1] with a step of 0.05 to guide the model training (Step 6 in Algorithm 1).

### 3.2 Experiment Settings

We tune hyper-parameters on the validation set. The size of vocabulary is set to 10K. Both the semantic and sentiment embeddings are 300-dimensional and are learned from scratch. We implement both encoder and decoder as a 1-layer LSTM with a hidden size of 256, and the former is bidirectional. The batch size is 64. We pre-train our model for 10 epochs with the MLE loss using pseudo-parallel sentences conducted by Jaccard Similarity, which is same as Liao et al. (2018). Harmonic weight $\beta$ in Eq. 9 is 1 and $\gamma$ in Eq. 6 is 0.5. The standard deviation $\sigma$ is set to 0.01 for yielding suitable peaked distributions. The sample size $K$ in Eq. 11 is set to 16. The optimizer is Adam (Kingma and Ba, 2014) with $10^{-3}$ initial learning rate for pre-training and $10^{-5}$ for cycleRL training. Dropout (Srivastava et al., 2014) is used to avoid overfitting.

### 3.3 Baselines

We compare our proposed method with the following two series of state-of-the-art systems.

**Fine-grained** systems aim to modify an input sentence to satisfy a given sentiment intensity. Liao et al. (2018) construct pseudo-parallel corpus to train a model which is a combination of a revised-VAE and a coupling component modeling pseudo-parallel data with three extra losses $\mathcal{L}_{extra}$. What's more, we also consider SC-Seq2Seq (Zhang et al., 2018a) which is a specificity controlled Seq2Seq model proposed in dialogue generation. In order to adapt to this unsupervised task, the proposed CycleRL training algorithm is used to train the SC-Seq2Seq model.

**Coarse-grained** systems aim to reverse the sentiment polarity (positive/negative) of the input, which can be regarded as a special case where the sentiment intensity is set below average (negative) or above average (positive). We compare our proposed method with the following state-of-the-art systems: CrossAlign (Shen et al., 2017), MultiDecoder (Fu et al., 2018), DeleteRetrieve (Li et al., 2018) and Unpaired (Xu et al., 2018).

### 3.4 Evaluation Metrics

We adopt both automatic and human evaluation.

#### 3.4.1 Automatic Evaluation

Automatic evaluation of FTST is an open and challenging issue, thereby we adopt a combination of multiple evaluation methods.

**Content:** To evaluate the content preservation performance, we hired crowd-workers on CrowdFlower[6] to write human references.[7] For each

[5]`https://www.yelp.com/dataset`

[6]`https://www.crowdflower.com/`

[7]We will release the collected human references and the

| Model | Automatic Evaluation | | | | | Human Evaluation | | | |
|---|---|---|---|---|---|---|---|---|---|
| | BLEU-1↑ | BLEU-2↑ | MAE↓ | MRRR↑ | PPL↓ | Content↑ | Sentiment↑ | Fluency↑ | **Avg**↑ |
| Revised-VAE | 22.6 | 7.2 | 0.24 | 0.62 | 102.2 | 2.64 | 2.52 | 2.13 | 2.43 |
| Revised-VAE + $\mathcal{L}_{extra}$ | 20.7 | 5.7 | 0.18 | 0.67 | 102.6 | 2.54 | 3.84 | 2.14 | 2.84 |
| SC-Seq2Seq | 23.9 | 3.8 | 0.25 | 0.69 | 41.2 | 2.37 | 3.85 | 3.41 | 3.21 |
| Seq2SentiSeq | **32.5** | **10.3** | **0.13** | **0.78** | **35.1** | **3.62** | **4.09** | **4.17** | **3.96** |
| Human Reference | 100.0 | 100.0 | 0.07 | 0.83 | 31.2 | 4.51 | 4.36 | 4.75 | 4.54 |

Table 1: Automatic evaluation and human evaluation in three aspects: Content (BLUE-1, BLUE-2), Sentiment (MAE, MRRR) and Fluency (PPL). **Avg** shows the average human scores. ↑ denotes larger is better, and vice versa. **Bold** denotes the best results.

review in the test dataset, crowd-workers are required to write five references with sentiment intensity value from $V' = [0.1, 0.3, 0.5, 0.7, 0.9]$. Therefore, the BLEU (Papineni et al., 2002) score between the human reference and the corresponding generated text of the same sentiment intensity can evaluate the content preservation performance.

**Fluency:** To measure the fluency, we calculate the perplexity (PPL) of each generated sequence via a pre-trained bi-directional LSTM language model (Mousa and Schuller, 2017).

**Sentiment:** In order to measure how close the sentiment intensity of outputs to the target intensity values, we define three metrics. Given an input sentence $x$ and a list of target intensity values $V = [v_1, v_2, ..., v_N]$, the corresponding outputs of the model are $[\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, ..., \hat{\boldsymbol{y}}_N]$. We then use a pre-trained sentiment regression scorer to predict the sentiment intensity values of outputs as $\hat{V} = [\hat{v}_1, \hat{v}_2, ..., \hat{v}_N]$. Following Liao et al. (2018), we use the mean absolute error (MAE $= \frac{1}{N}\sum_{i=1}^{N}|v_i - \hat{v}_i|$) between $V$ and $\hat{V}$ to measure the absolute gap.

Moreover, for fine-grained text sentiment transfer task, we expect that given a higher sentiment intensity value, the model will generate a more positive sentence. That is to say, the relative intensity ranking of all generated sentences of the same input is also important. Inspired by the Mean Reciprocal Rank metric which is widely used in the Information Retrieval area, we design a Mean Relative Reciprocal Rank (MRRR) metric to measure the relative ranking

$$\text{MRRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|rank(v_i) - rank(\hat{v}_i)| + 1} \quad (13)$$

In addition, we also compare our model with the coarse-grained sentiment transfer systems. In order to make the results comparable, we define the

generated test samples of all baselines for reproducibility.

sentiment intensity larger/smaller than 0.5 as positive/negative results. Then we use a pre-trained binary TextCNN classifier (Kim, 2014) to compute the classification accuracy.

### 3.4.2 Human Evaluation

We also perform human evaluation to assess the quality of generated sentences more accurately. Each item contains the source input, the sampled target sentiment intensity value, and the output of different systems. Then 500 items are distributed to 3 evaluators, who are required to score the generated sentences from 1 to 5 based on the input and target sentiment intensity value in terms of three criteria: content, sentiment, fluency. Content evaluates the content preservation degree. Sentiment refers to how much the output matches the target sentiment intensity. Fluency is designed to measure whether the generated texts are fluent. For each metric, the average Pearson correlation coefficient of the scores given by three evaluators is greater than 0.71, which ensures the inter-evaluator agreement.

## 4 Results and Discussion

### 4.1 Evaluation Results

The automatic evaluation and human evaluation results are shown in Table 1. It shows that our approach achieves the best performance in all metrics. More specifically, we have the following observations: (1) The proposed model Seq2SentiSeq obtains 8.6/3.1/0.98 points absolute improvement over the best results on BLEU-1/BLEU-2/Content score. It demonstrates the effectiveness of our approach in improving the content preservation of the input sentences. (2) Our model can more precisely control the sentiment intensity from human scores on sentiment, and it can also obtain both best results in sentiment mean absolute error (MAE) and relative sentiment rank (MRRR).

| Model | Automatic Evaluation | | | | | Human Evaluation | | | |
|---|---|---|---|---|---|---|---|---|---|
| | BLEU-1↑ | BLEU-2↑ | MAE↓ | MRRR↑ | PPL↓ | Content↑ | Sentiment↑ | Fluency↑ | **Avg**↑ |
| *Full Model* | **32.5** | **10.3** | **0.13** | **0.78** | 35.1 | 3.62 | 4.09 | 4.17 | 3.96 |
| *w/o Pre-training* | 14.3 | 0.7 | 0.32 | 0.48 | **7.2** | 1.01 | 1.30 | 3.86 | 2.06 |
| *w/o Cycle reconstruction* | 16.5 | 2.3 | 0.31 | 0.41 | 70.1 | 1.92 | 1.48 | 3.16 | 2.19 |
| *w/o Reinforcement learning* | 25.7 | 4.1 | 0.22 | 0.63 | 46.0 | 2.69 | 3.74 | 3.80 | 3.41 |

Table 2: Automatic evaluation and human evaluation of ablation study.

| **Model** | **neg-to-pos** | **pos-to-neg** |
|---|---|---|
| Multidecoder | 54.3 | 50.2 |
| CrossAlign | 73.3 | 71.7 |
| Unpaired | 78.9 | 73.0 |
| DeleteRetrieve | **89.6** | **83.1** |
| Revised-VAE | 64.3 | 62.0 |
| Revised-VAE + $\mathcal{L}_{extra}$ | 89.3 | 77.9 |
| SC-Seq2Seq | 67.2 | 59.6 |
| Seq2SentiSeq | **89.4** | **83.5** |

Table 3: Binary sentiment classification accuracy of the coarse-grained (upper) and fine-grained (lower) text sentiment transfer systems. **Bold** denotes the best results of each task.

| **Input** | the beer isn't bad, but the food was less than desirable. |
|---|---|
| **Output** | **Seq2SentiSeq** |
| **V=0.1** | the beer is terrible, and the food was the worst. |
| **V=0.3** | the beer wasn't bad, and the food wasn't great too. |
| **V=0.5** | the food is ok, but not worth the drive to the strip. |
| **V=0.7** | the beer is good, and the food is great. |
| **V=0.9** | the wine is great, and the food is extremely fantastic. |
| **Output** | **Revised-VAE + $\mathcal{L}_{extra}$** |
| **V=0.1** | n't no about about no when about that was when about |
| **V=0.3** | the beer sucks , but the food is not typical time. |
| **V=0.5** | the beer is cheap, but the food was salty and decor. |
| **V=0.7** | i just because decent management salty were impersonal. |
| **V=0.9** | n't that about was that when was about as when was |

Table 4: Example outputs with five sentiment intensity values V ranging from 0 to 1.

However, SC-Seq2Seq gets the second best MAE score while Revised-VAE + $\mathcal{L}_{extra}$ gets the second best MRRR score. We can infer that the two models excel at different aspects. And MRRR provides a different perspective on the sentiment results. (3) The proposed model can generate more fluent sentences than all baselines. The main reason for these three phenomenons is that we design two rewards that can directly ensure the content preservation and sentiment transformation in the cycle reinforcement training process. In addition, the cycle reconstruction loss can effectively guarantee the fluency of generated sentences, which has been further verified in the ablation study.

What's more, we also simplify our task to the setting of coarse-grained (positive/negative) sentiment transfer task. Table 3 shows the binary sentiment accuracy of the representative systems. We can find that the proposed model achieve the best results over the fine-grained systems, and it is comparable to the best coarse-grained system.

## 4.2 Ablation Study

In this section, we further discuss the impacts of the components of the proposed model. We retrain our model by ablating multiple components of our model: without pre-training, without cycle reconstruction (Eq. 12), without reinforcement learning ( Eq. 11). Table 2 shows the corresponding automatic and human evaluations. The performance declines most when without pre-training. This reveals that reinforcement learning is heavily dependent on pre-training as a warm start because it is hard for RL architecture to train from scratch. Moreover, no pre-training will lead the model to generate frequent words and short sentence which gets low PPL score. What's more, the performance of ablated version without cycle reconstruction also drops significantly, since cycle reconstruction plays an important role of teacher-forcing in our paper. Finally, even though the proposed Seq2SentiSeq without reinforcement learning can beat the best baseline in terms of human average score, reinforcement learning still helps to boost the performance of the proposed model by a large margin.

## 4.3 Case Study

Table 4 shows the example outputs on the YELP datasets with five sentiment intensity values. This case demonstrates that our model can both preserve the content (*"beer"*, *"food"*) and change the sentiment to the desired intensity. More importantly, our model can capture the subtle sentiment difference of the words or phrases, e.g., *"the worst"* → *"bad"* → *"ok"* → *"good"* → *"extremely fantastic"*. However, the Revised-VAE + $\mathcal{L}_{extra}$ system does not show this sentiment trend and may collapse when intensity value V is very

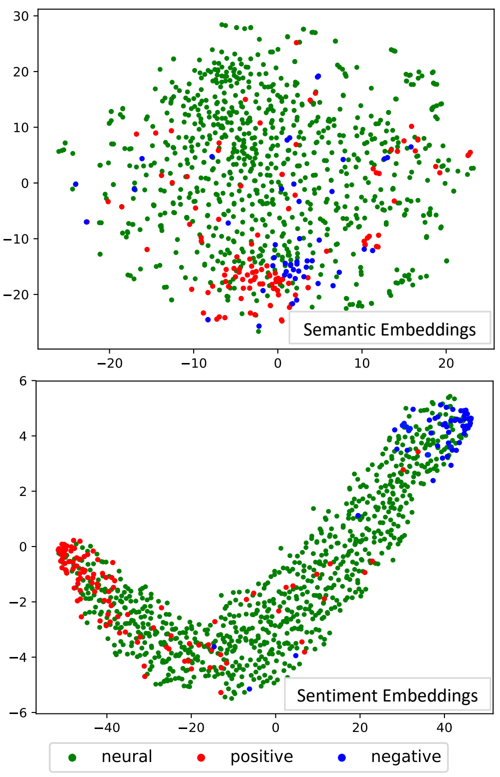

Figure 4: t-SNE visualization of the semantic embeddings (upper) and sentiment embeddings (lower) in the Seq2SentiSeq model.

small (0.1) or very big (0.9). And our model sometimes may also suffer from semantic drift, e.g., *"beer"* is revised to *"wine"*.

### 4.4 Analysis on Sentiment Representation

We also conduct analysis to understand the sentiment representations of words introduced in our model. We use the 1000 most frequent words from the training dataset. Then, we use a human annotated sentiment lexicon (Hutto and Gilbert, 2014) to classify them into three categories: positive, neutral and negative. After that, we get 112 positive words, 841 neutral words and 47 negative words. Finally, we apply t-SNE (Rauber et al., 2016) to visualize both semantic and sentiment embeddings of the proposed model (Figure 2) when finished training. As shown in Figure 4, we can see that the distributions of the two embeddings are significantly different. In the semantic embedding space, most of the positive words and negative words lie closely. On the contrary, in the sentiment embedding space, positive words are far from negative words. In conclusion, neighbors on semantic embedding space are semantically related, while neighbors on sentiment embedding space express a similar sentiment intensity.

## 5 Related Work

Recently, there is a growing literature on the task of unsupervised sentiment transfer. This task aims to reverse the sentiment polarity of a sentence but keep its content unchanged without parallel data (Fu et al., 2018; Tsvetkov et al., 2018; Li et al., 2018; Xu et al., 2018; Lample et al., 2019). However, there are few researches focus on the fine-grained control of sentiment. Liao et al. (2018) exploits pseudo-parallel data via heuristic rules, thus turns this task to a supervised setting. They then propose a model based on Variational Autoencoder (VAE) to first disentangle the content factor and source sentiment factor, and then combine the content with target sentiment factor. However, the quality of the pseudo-parallel data is not quite satisfactory, which seriously affects the performance of the VAE model. Different from them, we dynamically update the pseudo-parallel data via on-the-fly back-translation (Lample et al., 2018b) during training (Eq. 12).

There are some other tasks of NLP also show interest in controlling the fine-grained *attribute* of text generation. For example, Zhang et al. (2018a) and Ke et al. (2018) propose to control the specificity and diversity in dialogue generation. We borrow ideas from these works but the motivation and proposed models of our work are a far cry from them. The main differences are: (1) Since *sentiment* is dependent on local context while *specificity* is independent of local context, there is a series of design in our model to take the local context (or previous generated words) $s_t$ into consideration (e.g., Eq. 1, Eq. 3). (2) Due to the lack of parallel data, we propose a cycle reinforcement learning algorithm to train the proposed model (Section 2.3).

## 6 Conclusion

In this paper, we focus on solving the fine-grained text sentiment transfer task, which is a natural extension of the binary sentiment transfer task but with more challenges. We propose a Seq2SentiSeq model to achieve the aim of controlling the fine-grained sentiment intensity of the generated sentence. In order to train the proposed model without any parallel data, we design a cycle reinforcement learning algorithm. We apply the proposed approach to the Yelp review dataset, obtaining state-of-the-art results in both automatic evaluation and human evaluation.

## Acknowledgments

This paper is supported by NSFC project 61751201, 61772040 and 61876004. The contact authors are Baobao Chang and Zhifang Sui.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations, ICLR 2014*.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-18*, pages 663–670.

Clayton J. Hutto and Eric Gilbert. 2014. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014*.

Pei Ke, Jian Guan, Minlie Huang, and Xiaoyan Zhu. 2018. Generating informative responses with controlled sentence function. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 1746–1751.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations, ICLR 2014*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *Proceedings of the International Conference on Learning Representations, ICLR 2018*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pages 5039–5049.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *International Conference on Learning Representations, ICLR 2019*.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 2157–2169.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*, pages 1865–1874.

Yi Liao, Lidong Bing, Piji Li, Shuming Shi, Wai Lam, and Tong Zhang. 2018. QuaSE: Sequence editing under quantifiable guidance. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pages 3855–3864.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 1412–1421.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT*, pages 1–17.

Amr Eldesoky Mousa and Bjorn W Schuller. 2017. Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 1023–1032.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, ACL 2002*, pages 311–318.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. In *Proceedings of the International Conference on Learning Representations, ICLR 2017*.

Paulo E. Rauber, Alexandre X. Falcão, and Alexandru C. Telea. 2016. Visualizing time-dependent data using dynamic t-SNE. In *Eurographics Conference on Visualization, EuroVis 2016*, pages 73–77.

Cícero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, pages 189–194.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems, NIPS 2017*, pages 6833–6844.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. In *Journal of Machine Learning Research*, pages 1929–1958.

Yulia Tsvetkov, Alan W. Black, Ruslan Salakhutdinov, and Shrimai Prabhumoye. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, pages 866–876.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning*, pages 229–256.

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, pages 979–988.

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2018a. Learning to control the specificity in neural response generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, pages 1108–1117.

You Zhang, Hang Yuan, Jin Wang, and Xuejie Zhang. 2017. YNU-HPCC at EmoInt-2017: Using a CNN-LSTM model for sentiment intensity prediction. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP 2017*, pages 200–204.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. 2018b. Style transfer as unsupervised machine translation. In *arXiv preprint arXiv:1808.07894*.