Specificity-Driven Cascading Approach for Unsupervised Sentiment Modification

Pengcheng Yang[1], **Junyang Lin**[2], **Jingjing Xu**[2], **Jun Xie**[3], **Xu Sun**[2], **Qi Su**[2]
[1] Deep Learning Lab, Beijing Institute of Big Data Research, Peking University
[2] MOE Key Lab of Computational Linguistics, School of EECS, Peking University
[3] Tencent, Beijing, China
{yang_pc,linjunyang,jingjingxu,xusun,sukia}@pku.edu.cn
stiffxie@tencent.com

Abstract

The task of unsupervised sentiment modification aims to reverse the sentiment polarity of the input text while preserving its semantic content without any parallel data. Most previous work follows a two-step process. They first separate the content from the original sentiment, and then directly generate text with the target sentiment only based on the content produced by the first step. However, the second step bears both the target sentiment addition and content reconstruction, thus resulting in a lack of specific information like proper nouns in the generated text. To remedy this, we propose a specificity-driven cascading approach in this work, which can effectively increase the specificity of the generated text and further improve content preservation. The experiments show that our approach outperforms competitive baselines by a large margin, which achieves 11% and 38% relative improvements of the overall metric on the Yelp and Amazon datasets, respectively.

1 Introduction

The goal of unsupervised sentiment modification is to reverse the sentiment polarity of the input text while preserving its semantic content without any parallel data. It not only has a variety of practical applications, e.g., fighting offensive language on social media (dos Santos et al., 2018), but also serves as an ideal testbed for controllable text generation. However, the input for this task usually contains some specific information like proper nouns. For instance, "*Michael*" in Table 1 is a human name, which belongs to the specific information. This specific information is important because it is often the subject or object of the sentence and the bearer of sentiment. Therefore, it needs to be fully preserved in the process of sentiment modification. We refer this attribute as the specificity of the output.

Input	***Michael*** *'s work exceeds my expectations.*
Shen et al. (2017)	*Service is terrible.*
Fu et al. (2018)	*I was very disappointed.*
Xu et al. (2018)	*The girls do horrible work.*

Table 1: Examples generated by four different previous methods. "***Michael***" is a proper noun, which belongs to specific information.

Most previous work on unsupervised sentiment modification follows a two-step process. They first separate the content from the original sentiment, either in an implicit (Hu et al., 2017; Shen et al., 2017; Fu et al., 2018; Tsvetkov et al., 2018) or explicit (Xu et al., 2018; Zhang et al., 2018a) way. Then, they directly generate text with the target sentiment only based on the content produced by the first step. In the second step, the decoder does not include the sentiment information in its source input, meaning that the source information is incomplete for generating a sentiment-transferred sentence. This *information gap* at both ends of the decoder causes the decoder needs to bear both the target sentiment addition and content reconstruction. Thus, it is difficult for the decoder to balance sentiment transformation and content preservation simultaneously, resulting in the loss of specific information contained in the source input during decoding. This not only makes the generated text lacks specificity, but also causes poor content preservation. Table 1 intuitively illustrates this problem. All three traditional methods miss proper noun "*Michael*" in the output.

In order to generate text containing more specific information, in this paper, we propose a specificity-driven cascading approach. Our approach consists of three modules: separation module, emotionalization module, and fusion module. The separation module is responsible for explicitly separating sentiment words from content words

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 5508–5517,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

via the self-attention mechanism. Then, different from previous work, we divide the process of generating text with the target sentiment based on content words into two steps. First of all, in order to achieve sentiment addition, the emotionalization module generates corresponding target sentiment words based on the content words. Then, the fusion module performs content reconstruction by fusing semantic content and target sentiment words to generate sentiment-transferred sentences. Each step is dedicated to their respective goals, which reduces the burden of the decoder in the fusion module. Besides, with the help of the target sentiment words generated by the emotionalization module, the *information gap* is also eliminated. This allows specific information contained in the input to be well preserved, resulting in more specific output and better content preservation.

The main contributions of this paper are summarized as follows:

- We propose an effective specificity-driven cascading approach for unsupervised sentiment modification, which not only increases the specificity of the generated text, but also further improves content preservation.
- Experimental results show that our approach can outperform the competitive baselines by a large margin. Further analysis demonstrates that the proposed method can achieve better reconcile of sentiment transformation and content preservation.

2 Background

Here we introduce some necessary background knowledge. The core component of our proposed approach is the Seq2Seq model (Sutskever et al., 2014; Bahdanau et al., 2014), which usually consists of an encoder and a decoder with the attention mechanism.

Encoder: Given the input $\boldsymbol{x} = (x_1, \cdots, x_m)$, the encoder implemented as a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) structure reads $\boldsymbol{x}$ from both directions and computes the hidden states for each word,

$$\overrightarrow{h}_i = \overrightarrow{\text{LSTM}}(\overrightarrow{h}_{i-1}, x_i) \tag{1}$$

$$\overleftarrow{h}_i = \overleftarrow{\text{LSTM}}(\overleftarrow{h}_{i+1}, x_i) \tag{2}$$

Module	**Source Input**	**Target Output**	**Role**
SC$_\phi$	$\boldsymbol{x}^+/\boldsymbol{x}^-$	$+/-$	Separate content and sentiment
S2S$_{\theta_{pe}}$	$\boldsymbol{c}^+$	$\boldsymbol{s}^+$	Generate positive sentiment words
S2S$_{\theta_{ne}}$	$\boldsymbol{c}^-$	$\boldsymbol{s}^-$	Generate negative sentiment words
S2S$_{\theta_{pf}}$	$(\boldsymbol{c}^+, \boldsymbol{s}^+)$	$\boldsymbol{x}^+$	Generate positive sentence
S2S$_{\theta_{nf}}$	$(\boldsymbol{c}^-, \boldsymbol{s}^-)$	$\boldsymbol{x}^-$	Generate negative sentence

Table 2: The role of each component of our approach at the training stage. We use "+" and "-" to represent positive and negative sentiment, respectively.

where $e(x_i)$ denotes the embedding of the word x_i. The final hidden representation of the i-th word is $h_i = [\overrightarrow{h}_i; \overleftarrow{h}_i]$, which semicolon represents vector concatenation.

Decoder: Given the hidden representations $(h_1, \cdots, h_m)$, the decoder generates words sequentially. Besides, the attention mechanism is used to automatically select the most informative words at different time steps. In detail, the hidden state s_{t+1} of the decoder at time-step $t+1$ is computed as follows:

$$e_{t,i} = f(s_t, h_i) \tag{3}$$

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{m} \exp(e_{t,j})} \tag{4}$$

$$c_t = \sum_{i=1}^{m} \alpha_{t,i} h_i \tag{5}$$

$$s_{t+1} = \text{LSTM}(s_t, [e(y_t); c_t]) \tag{6}$$

$$y_t \sim \text{softmax}\left(g(s_t)\right) \tag{7}$$

where $[e(y_t); c_t]$ means the concatenation of vectors $e(y_t)$ and c_t. $e(y_t)$ is the embedding of the word y_t that has the highest probability. $f(s_t, h_i)$ is an aligned model, which measures the dependency between s_t and h_i. g is a function, which is used to transform the hidden state s_t into the probability distribution over the vocabulary space.

The model is trained by maximizing the conditional likelihood of the ground-truth sequence $\boldsymbol{y}^* = (y_1^*, \cdots, y_n^*)$. Specially, the objective is to minimize the cross-entropy loss:

$$\mathcal{L} = -\sum_{t=1}^{n} \log\left(p(y_t^* | \boldsymbol{y}_{<t}^*, \boldsymbol{x})\right)^{1} \tag{8}$$

3 Methodology

3.1 Overview

Here we define some notations and describe the sentiment modification task. We use $\mathcal{D}^+ =$

[1] In this work, $\boldsymbol{y}_{<t}^*$ denotes the sequence $(y_1^*, \cdots, y_{t-1}^*)$.

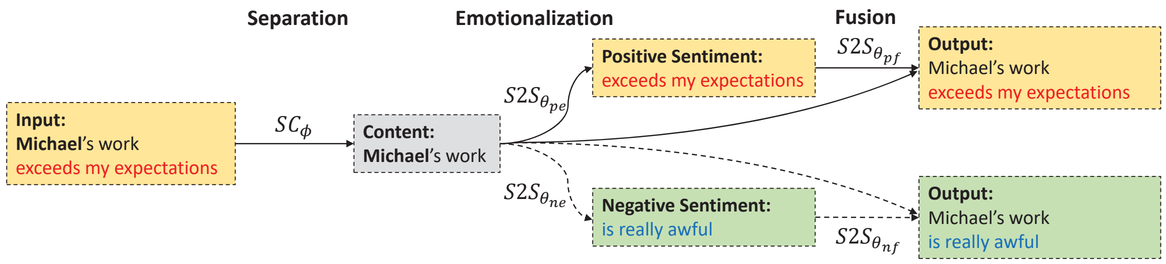

Figure 1: The overview of the proposed cascading approach with a positive input. The process with a negative input is in a similar way. Solid and dashed lines indicate the training process and the testing process, respectively.

$\{\boldsymbol{x}_i^+\}_{i=1}^{n_1}$ and $\mathcal{D}^- = \{\boldsymbol{x}_i^-\}_{i=1}^{n_2}$ to represent the positive corpus and negative corpus, respectively, where $\boldsymbol{x}^+$ ($\boldsymbol{x}^-$) is a positive (negative) sentence. $\mathcal{D} = \mathcal{D}^+ \cup \mathcal{D}^-$ denotes the complete corpus. The sentiment modification task aims to take the sentence $\boldsymbol{x}^+$ ($\boldsymbol{x}^-$) as input and generate a sentiment-transferred sentence $\hat{\boldsymbol{x}}^-$ ($\hat{\boldsymbol{x}}^+$) as output.

The overview of our approach is shown in Figure 1. Our approach consists of three modules: separation module, emotionalization module, and fusion module. The separation module aims to explicitly separate the sentiment words from content words and the emotionalization module receives content words as input to generate the corresponding sentiment words. Finally, the fusion module fuses the content and sentiment words to generate fluent sentences. Table 2 summarizes the source input, target output, and function of each component at the training stage. Besides, the training algorithm and testing algorithm are shown in Algorithm 1 and Algorithm 2, respectively.

3.2 Separation Module

The separation module aims to explicitly separate the sentiment words from content words. Given a sentence $\boldsymbol{x} = (x_1, \cdots, x_m)$, the separation can be accomplished by identifying whether each word x_i is a sentiment word. Here we employ a sentiment classifier with the self-attention mechanism to achieve this goal. The classifier with the self-attention mechanism can select the most informative words automatically by assigning higher attention weights to sentiment words. Therefore, the attention weight α_i of the i-th word can be used to identify whether x_i is a sentiment word.

In detail, we use the bidirectional LSTM model for implementation. The LSTM model reads the input $\boldsymbol{x}$ from both directions and computes the hidden representation $h_i = [\overrightarrow{h}_i; \overleftarrow{h}_i]$ for the i-th word, where semicolon represents vector concatenation. The self-attention mechanism produces final context vector h by aggregating the hidden representations $(h_1, \cdots, h_m)$. Specifically,

$$\alpha_i = \text{softmax}(v^T h_i) \tag{9}$$

$$h = \sum_{i=1}^{m} \alpha_i h_i \tag{10}$$

where v is a parameter vector. At the training stage, we perform classification based on the final context vector h and the loss function is the cross-entropy loss. Then, for the well-trained sentiment classifier, the average attention value of sentence $\boldsymbol{x}$ is calculated as:

$$\bar{\alpha} = \frac{1}{m} \sum_{i=1}^{m} \alpha_i \tag{11}$$

Words with attention weight larger than $\bar{\alpha}$ can be regarded as sentiment words, and the remaining words can be regarded as content words.[2] Therefore, each sentence $\boldsymbol{x}$ can be separated into content words $\boldsymbol{c}$ and sentiment words $\boldsymbol{s}$.[3]

3.3 Emotionalization Module

In previous work, there is an *information gap* at both ends of the decoder, making the decoder to bear both the target sentiment addition and content reconstruction simultaneously. To eliminate the *information gap* and ease the burden of the decoder, we divide the process of generating sentiment sentences based on content into two separate steps: sentiment addition and content reconstruction. Each step is dedicated to their respective goals, which reduces the burden of the

[2] The reason is that there exists an obvious gap between the attention weights of content words and sentiment words, causing the weights of these two types of words to be distributed at both ends of the average. Exploratory experiments show that the separation model can achieve more than 80% F_1 score.

[3] Except for Section 2, we use $\boldsymbol{c}$ and $\boldsymbol{s}$ to represent content words and sentiment words, respectively. This is different from the context vector c and hidden state s in Section 2.

Algorithm 1 The training algorithm.

```
Require: positive corpus $\mathcal{D}^+ = \{\boldsymbol{x}_i^+\}_{i=1}^{n_1}$; negative corpus $\mathcal{D}^- = \{\boldsymbol{x}_i^-\}_{i=1}^{n_2}$; complete corpus $\mathcal{D} = \mathcal{D}^+ \cup \mathcal{D}^-$
Separation:
    Train the sentiment classifier $\mathbf{SC}_\phi$ on corpus $\mathcal{D}$
    Separate positive sentence $\boldsymbol{x}^+$ in $\mathcal{D}^+$ into $(\boldsymbol{c}^+, \boldsymbol{s}^+)$ using $\mathbf{SC}_\phi$ to construct new positive corpus $\mathcal{N}^+ = \{(\boldsymbol{c}_i^+, \boldsymbol{s}_i^+, \boldsymbol{x}_i^+)\}_{i=1}^{n_1}$
    Separate negative sentence $\boldsymbol{x}^-$ in $\mathcal{D}^-$ into $(\boldsymbol{c}^-, \boldsymbol{s}^-)$ using $\mathbf{SC}_\phi$ to construct new negative corpus $\mathcal{N}^- = \{(\boldsymbol{c}_i^-, \boldsymbol{s}_i^-, \boldsymbol{x}_i^-)\}_{i=1}^{n_2}$
Emotionalization:
    Train $\mathbf{S2S}_{\theta_{pe}}$ using MLE on new positive corpus $\mathcal{N}^+$
    Train $\mathbf{S2S}_{\theta_{ne}}$ using MLE on new negative corpus $\mathcal{N}^-$
Fusion:
    Train $\mathbf{S2S}_{\theta_{pf}}$ using MLE on new positive corpus $\mathcal{N}^+$
    Train $\mathbf{S2S}_{\theta_{nf}}$ using MLE on new negative corpus $\mathcal{N}^-$
```

decoder. Here the emotionalization module explicitly performs sentiment addition by generating corresponding sentiment words based on content words.

In detail, through the separation module, the sentence $\boldsymbol{x}$ can be separated into $(\boldsymbol{c}, \boldsymbol{s})$, where $\boldsymbol{c}$ and $\boldsymbol{s}$ represent the content words and sentiment words, respectively. Therefore, the new corpus $\mathcal{N}^+ = \{(\boldsymbol{c}_i^+, \boldsymbol{s}_i^+, \boldsymbol{x}_i^+)\}_{i=1}^{n_1}$ can be constructed by explicitly separating the positive sentence $\boldsymbol{x}^+$. Similarly, $\mathcal{N}^- = \{(\boldsymbol{c}_i^-, \boldsymbol{s}_i^-, \boldsymbol{x}_i^-)\}_{i=1}^{n_2}$ can be constructed by separating the negative sentence $\boldsymbol{x}^-$.

The emotionalization module aims to automatically generate corresponding sentiment words according to the content. Since corpus $\mathcal{N}^+$ (or $\mathcal{N}^-$) provides pseudo-parallel data about content words $\boldsymbol{c}^+$ (or $\boldsymbol{c}^-$) and sentiment words $\boldsymbol{s}^+$ (or $\boldsymbol{s}^-$), here we use the Seq2Seq model to accomplish this goal. We train a Seq2Seq model $\mathbf{S2S}_{\theta_{pe}}$ on the new positive corpus $\mathcal{N}^+$, which is responsible for generating positive sentiment words $\boldsymbol{s}^+$ based on the content $\boldsymbol{c}^+$. Another Seq2Seq model $\mathbf{S2S}_{\theta_{ne}}$ is trained on the new negative corpus $\mathcal{N}^-$ to generate negative sentiment words $\boldsymbol{s}^-$ based on content words $\boldsymbol{c}^-$. Therefore, for the well-trained model $\mathbf{S2S}_{\theta_{pe}}$ (or $\mathbf{S2S}_{\theta_{ne}}$), given content words $\boldsymbol{c}$, $\mathbf{S2S}_{\theta_{pe}}$ (or $\mathbf{S2S}_{\theta_{ne}}$) is able to generate corresponding positive (or negative) sentiment words $\boldsymbol{s}^+$ (or $\boldsymbol{s}^-$).

3.4 Fusion Module

The fusion module aims to perform content reconstruction by fusing content words $\boldsymbol{c}$ and sentiment words $\boldsymbol{s}$ to generate fluent emotional sentence $\boldsymbol{x}$. Compared with the previous work, the decoder here has extra sentiment information $\boldsymbol{s}$, thus eliminating the *information gap* at both ends of the decoder. This allows the specific information to be better preserved during the generation process, leading to more specific generation results and better content preservation.

Algorithm 2 The testing algorithm.

```
Require: positive source input $\boldsymbol{x}^+$
Separate $\boldsymbol{x}^+$ using $\mathbf{SC}_\phi$ to get content words $\boldsymbol{c}^+$
Generate the negative sentiment words $\boldsymbol{s}^-$ using $\mathbf{S2S}_{\theta_{ne}}$ with input $\boldsymbol{c}^+$
Generate the negative sentence $\hat{\boldsymbol{x}}^-$ using $\mathbf{S2S}_{\theta_{nf}}$ with input $(\boldsymbol{c}^+, \boldsymbol{s}^-)$
```

Similar to the emotionalization module, the fusion module is also composed of two Seq2Seq models, one for generating positive sentences and the other for generating negative sentences. Here each Seq2Seq model consists of two encoders and one decoder. The two encoders are responsible for compressing the semantic content words and emotional words into a dense vector, respectively. The decoder aims to generate fluent emotional sentence based on compressed content vector and sentiment vector. In detail, we train two bi-encoder based Seq2Seq models $\mathbf{S2S}_{\theta_{pf}}$ and $\mathbf{S2S}_{\theta_{nf}}$ on the new positive corpus $\mathcal{N}^+$ and negative corpus $\mathcal{N}^-$, respectively. Each model aims to output $\boldsymbol{x}$ based on the input $(\boldsymbol{c}, \boldsymbol{s})$. Note that the Seq2Seq model here has two encoders, so we need to make modifications to the relevant formulas. Two attention mechanisms similar to Eq. 3 - Eq. 5 aggregate the hidden representations of the content words $\boldsymbol{c}$ and sentiment words $\boldsymbol{s}$ respectively to form the content-side context c_t^c and the sentiment-side context c_t^s. The hidden state s_{t+1} of the decoder at time-step $t+1$ is computed as follows:

$$s_{t+1} = \mathrm{LSTM}(s_t, [e(y_t); c_t^c; c_t^s]) \tag{12}$$

3.5 Training and Testing

At the training stage, each module is trained by optimizing the cross-entropy loss function. The testing algorithm is shown in Algorithm 2. Without loss of generality, we focus on reversing the positive input to the negative output. The process of reversing negative sentence to positive sentence is in a similar way. Given a positive input $\boldsymbol{x}^+$, the separation module can separate $\boldsymbol{x}^+$ into content words $\boldsymbol{c}^+$ and sentiment words $\boldsymbol{s}^+$. The model $\mathbf{S2S}_{\theta_{ne}}$ in the emotionalization module takes $\boldsymbol{c}^+$ as input to generate corresponding negative sentiment words $\boldsymbol{s}^-$. Finally, $(\boldsymbol{c}^+, \boldsymbol{s}^-)$ is inputted into the model

Dataset	Training	Validation	Test
Yelp	580K	15K	4K
Amazon	230K	10K	2K

Table 3: The statistics of two datasets.

S2S$_{\theta_{nf}}$ in the fusion module to generate a fluent negative sentence $\hat{\boldsymbol{x}}^-$.

4 Experiments

In this section, we first introduce the datasets, evaluation metrics, experimental details, and all compared baselines.

4.1 Datasets

We conduct experiments on two datasets, Yelp[4] and Amazon[5] (He and McAuley, 2016). Both datasets are composed of a large number of reviews. Following previous work (Shen et al., 2017), reviews with scores below 3 are labeled as negative, and reviews with scores above 3 are labeled as positive. The reviews which exceed 20 words or less than 5 words are further filtered out. In addition, we train the sentiment classifier (Kim, 2014) to filter samples with the category confidence below 0.8. We divide each dataset into training, validation and test sets. The statistics of the two processed datasets is shown in Table 3.

4.2 Evaluation Metrics

In this paper, we adopt two evaluation methods: automatic evaluation and human evaluation.

4.2.1 Automatic Evaluation

Following previous work (Xu et al., 2018; Shen et al., 2017), we perform automatic evaluation in terms of sentiment transformation and content preservation.

- **ACC:** We pre-train a sentiment classifier, which is implemented as CNN (Kim, 2014) structure, to evaluate whether the sentiment of the generated text matches the target sentiment. The pre-trained sentiment classifier achieves the accuracy of 87% and 89% on Yelp and Amazon datasets, respectively. The higher ACC value indicates better sentiment transformation.

	Yelp		Amazon	
Module	**LSTM Layer**	**Hidden Size**	**LSTM Layer**	**Hidden Size**
SC$_\phi$	2	512	2	256
S2S$_{\theta_{pe}}$	(2, 3)	(256, 512)	(2, 2)	(256, 512)
S2S$_{\theta_{ne}}$	(2, 2)	(256, 512)	(1, 2)	(128, 256)
S2S$_{\theta_{pf}}$	(2, 3)	(256, 512)	(2, 2)	(256, 512)
S2S$_{\theta_{nf}}$	(2, 2)	(256, 512)	(1, 2)	(128, 256)

Table 4: Main hyper-parameters. For the LSTM layer and hidden size, we use $(\cdot, \cdot)$ to represent the hyperparameter of the encoder and decoder, respectively.

- **BLEU:** We use the BLEU score (Papineni et al., 2002) between the transferred sentence and the source sentence to evaluate content preservation. The higher BLEU score indicates better content preservation.

- **G-Score:** In order to evaluate the overall performance of different systems, following Xu et al. (2018), we compute the geometric mean (G-score) of ACC value and BLEU score as an overall evaluation metric.

4.2.2 Human Evaluation

In order to evaluate the quality of generated sentences more accurately, we also perform human evaluation. Each item contains the source input, transfer direction, and the transferred sentences generated by different models. Then 200 items are distributed to 3 annotators with the linguistic background. The annotators are required to score the generated sentences from 1 to 5 based on the source input and transfer direction in terms of three criteria: sentiment, content, and overall performance. Sentiment represents whether the generated text matches the target sentiment. Content evaluates the content preservation degree. For each dataset and each metric, the average Pearson correlation coefficient of the scores given by three annotators is greater than 0.65, which indicates that the human scores are highly consistent.

4.3 Baselines

We compare our proposed approach with the following competitive systems:

- **Cross-Aligned Auto-Encoder (CAAE)** is proposed by Shen et al. (2017). They propose to use the refined alignment of latent representations in hidden layers to perform sentiment modification.

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

[5] https://www.amazon.com/

Yelp	ACC	BLEU	G-Score
CAAE (Shen et al., 2017)	50.20	1.35	8.23
MDAE (Fu et al., 2018)	81.30	3.27	16.30
SEAE (Fu et al., 2018)	42.25	19.18	28.47
BT (Tsvetkov et al., 2018)	61.45	2.56	12.54
CRL (Xu et al., 2018)	73.70	35.62	51.24
Proposed Method	68.65	47.42	**57.06**
Amazon	**ACC**	**BLEU**	**G-Score**
CAAE (Shen et al., 2017)	51.75	2.17	10.60
MDAE (Fu et al., 2018)	67.85	4.54	17.55
SEAE (Fu et al., 2018)	47.50	28.79	36.98
BT (Tsvetkov et al., 2018)	53.75	3.73	14.16
CRL (Xu et al., 2018)	65.20	22.66	38.44
Proposed Method	56.30	49.83	**52.97**

Table 5: Automatic evaluations of our method and baselines, from which we can see that our approach achieves the best overall performance on both datasets.

Yelp	Sentiment	Content	Overall
CAAE (Shen et al., 2017)	3.48	1.87	2.03
MDAE (Fu et al., 2018)	3.56	2.07	3.11
SEAE (Fu et al., 2018)	2.67	2.71	3.26
BT (Tsvetkov et al., 2018)	1.81	3.14	2.52
CRL (Xu et al., 2018)	3.64	3.03	3.89
Proposed Method	3.31	4.21	**4.09**
Amazon	**Sentiment**	**Content**	**Overall**
CAAE (Shen et al., 2017)	3.37	2.74	2.72
MDAE (Fu et al., 2018)	3.31	2.89	2.58
SEAE (Fu et al., 2018)	2.82	3.05	3.13
BT (Tsvetkov et al., 2018)	2.04	2.58	2.94
CRL (Xu et al., 2018)	3.68	3.21	3.06
Proposed Method	3.42	3.83	**3.71**

Table 6: Human evaluations of different systems, showing that our approach outperforms the baselines by a large margin, especially in content preservation.

- **Multi-Decoder Auto-Encoder (MDAE)** is proposed by Fu et al. (2018). They use a separate decoder for each sentiment to generate transferred sentences based on the extracted content vectors.

- **Style-Embedding Auto-Encoder (SEAE)** is also proposed by Fu et al. (2018). Each sentiment has a unique embedding, which is used to control the sentiment transfer direction of the decoder.

- **Back-Translation (BT)** (Tsvetkov et al., 2018) focuses on learning a latent representation of the input by means of language translation model and further perform generation.

- **Cycled Reinforcement Learning (CRL)** is proposed by Xu et al. (2018). They use a cycled reinforcement learning model to explicitly extract content to further generate transferred sentences.

4.4 Experiment Settings

For both datasets, the vocabulary size is 30,000 and out-of-vocabulary words are replaced with unk. We train *word2vec* (Mikolov et al., 2013) on an English Wikipedia corpus dump to get 300-dimensional word embeddings. The batch size is 64 and the hyper-parameters of each module on two datasets are shown in Table 4. For each module, we use the Adam (Kingma and Ba, 2014) optimizer and the initial learning rate is 0.0003. Besides, we make use of the dropout method (Srivastava et al., 2014) to avoid over-fitting and clip the gradients (Pascanu et al., 2013) to the maximum norm of 10. During training, we train the model for a fixed number of epochs and monitor its performance on the validation set after 100 updates. Once the training is finished, we select the model with the best G-score on the validation set as our final model and evaluate its performance on the test set.

5 Results and Discussion

In this section, we report the experimental results. Besides, further analysis is also provided.

5.1 Experimental Results

The automatic evaluation results on two datasets are shown in Table 5. Results show that our approach achieves the best overall performance. For instance, our model achieves 11% and 38% relative improvement of G-score over the best baseline on the two datasets, respectively. This shows that our approach can further reconcile sentiment transformation and content preservation. We also note that our method achieves particularly high BLEU scores, indicating that the proposed method can enhance the specificity of the generated text, leading to better content preservation.

The human evaluation results are shown in Table 6, from which we can draw similar conclusions. It is obvious that our approach can outperform the baselines by a large margin, especially in terms of content preservation. The reason is that there is no *information gap* at both ends of our decoder when generating sentiment-transferred sen-

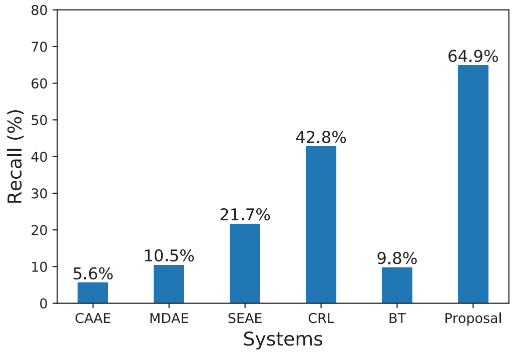

Figure 2: The recall scores of specific information (e.g. proper nouns) on different systems.

tences, which relieves the burden on the decoder. Therefore, more specific information is preserved.

5.2 Effectiveness of Enhancing Specificity

The analysis in Section 5.1 has shown that our approach can achieve excellent content preservation, which shows that the proposed model can generate text containing more specific information to some extent. To further verify this conclusion, we manually select test samples containing specific information (e.g. proper nouns) and then calculate the recall scores of this specific information on different systems. Figure 2 shows the related results, from which we can see that our approach achieves the highest recall score. This further illustrates that our approach can effectively improve the specificity of the generated text.

5.3 Ablation Study

We conduct an ablation study to verify the effectiveness of the emotionalization module. The results are shown in Table 7, illustrating that the removal of the emotionalization module leads to an obvious reduction in model performance. Compared to the ablated version, the full model achieves improvements of 11.14% and 9.87% G-score on two datasets, respectively. Besides, both sentiment transformation and content preservation can be improved with the emotionalization module. The emotionalization module can automatically generate corresponding sentiment words based on the extracted content, which eliminates *information gap* and relieves the burden of the decoder, leading to better model performance.

Yelp	**ACC**	**BLEU**	**G-score**
Full model	68.65	47.42	57.06
w/o EM	52.40	40.24	45.92
Amazon	**ACC**	**BLEU**	**G-score**
Full model	56.30	49.83	52.97
w/o EM	44.90	41.37	43.10

Table 7: Ablation study. "EM" denote the emotionalization module.

Input:	*The worst and unprofessional company.*
Separation:	*The worst and unprofessional company.*
Emotionalization:	*Exceeded my expectations*
Fusion:	*The company exceeded my expectations.*
Input:	*Staff is awesome and always friendly.*
Separation:	*Staff is awesome and always friendly.*
Emotionalization:	*So rude unprofessional*
Fusion:	*So rude and unprofessional staff.*

Table 8: Outputs of different modules of our method. The underlined words are sentiment words recognized by separation module.

5.4 Case Study

Table 8 shows several examples of the testing process for our approach, which intuitively reflects the function of each module. Given an input, the separation module can separate sentiment words from content words. For instance, the separation module successfully identifies the sentiment words "*worst and unprofessional*" in the first example. Then, the emotionalization module is able to generate corresponding target sentiment expression "*exceeded my expectations*" based on the given content words "*the company*". Beyond that, the fusion module can fuse content words and target sentiment expression to further generate a fluent sentiment-transferred sentence "*the company exceeded my expectations*".

Table 9 presents several outputs of different systems on the Yelp dataset. The baselines tend to generate outputs lacking specific information, resulting in poor content preservation. For instance, all baselines miss proper noun "*wendys*" in the first example. In contrast, our approach increases the specificity of the output, leading to better content preservation. Our method not only performs sentiment addition and content reconstruction independently, but also eliminates the *information gap*. This eases the burden of the decoder, allowing more specific information to be preserved.

5.5 Error Analysis

Although our approach outperforms the baselines by a large margin, we still observe some failed examples. We conduct a detailed analysis and categorize them into two types demonstrated below.

Input:	***Wendys*** *is my favorite chain, but this location is the worst.*
CAAE:	*Great food and the best food!*
MDAE:	*Best best i ever visit.*
SEAE:	*This place had amazing stars!*
CRL:	*Undoubtedly is my favorite chain.*
Proposal:	***Wendys*** *is my favorite chain, great location!*
Input:	***Annette*** *in optical was super helpful!*
CAAE:	*I have been to say and i have n't be them.*
MDAE:	*Terrible here is not disappointed , it!*
SEAE:	*Walked in the service was super well.*
CRL:	*Horrible stylist bank was extremely unprofessional.*
Proposal:	***Annette*** *in optical was rude!*

Table 9: Examples generated by different systems on the Yelp dataset. "***Wendys***" and "***Annette***" belong to the specific information contained in the input.

One type of error is the neutral output without any sentiment polarity, such as "*I went to the bar on Friday*". This is because both datasets contain a large number of neutral reviews, which can be viewed as noises. Since these neutral reviews are also labeled as positive or negative, the separation module may incorrectly identify some content words as sentiment words. Training samples lacking sentiment information make the emotionalization module unable to generate sentiment words, leading to the neutral output.

The other type of error is the output that contains the original sentiment, which can be attributed to the failure of the separation of content and sentiment. Such failure usually occurs when the input contains sentiment words that are unseen in the training data, or the sentiment is implicitly expressed. The separation module may not recognize these unseen or obscure sentiment words, causing that the extracted content still contains original sentiment information. This content is further inputted to the emotionalization module, causing the output to contain original sentiment.

6 Related Work

This paper is mainly related to the following two series of work.

Unsupervised sentiment modification. One line of research for this task focuses on implicitly separating content from original sentiment. To control the sentiment, Hu et al. (2017) combines VAE and attribute discriminator while Shen et al. (2017) strives to align latent representations in hidden layers. Both Fu et al. (2018) and Yang et al. (2018) adopt adversarial training to ensure the success of separation. Tsvetkov et al. (2018) strives to obtain style-independent content representations through additional translation models. However, this implicit separation tends to result in poor content preservation since content and sentiment are entangled in an uninterpretable way. Different from this line of work, we adopt explicit separation of content and sentiment via self-attention, which achieves better content preservation. Another line strives to explicitly separate content and sentiment at the word-level. Li et al. (2018) proposes a retrieval system, which may lead to low-quality outputs. Xu et al. (2018) presents a reinforcement learning (RL) approach to remove sentiment words. However, the training of RL is unstable and sensitive to hyper-parameters. Further, Zhang et al. (2018a) tries to automatically extracts appropriate sentiment information from learned sentiment memories. Different from Zhang et al. (2018a) which extracts entangled sentiment memories from capacity-constrained memory matrices, our work can generate accurate explicit sentiment words flexibly based on specific content, leading to better performance. There also exist some other endeavors which construct pseudo-parallel data by means of back-translation (Sennrich et al., 2015). For instance, Zhang et al. (2018b) and Luo et al. (2019) jointly train two transfer systems via iterative back-translation. Further, Lample et al. (2019) extend this training framework to support multiple attribute control .

Sentiment analysis. Our work is also related to: sentiment analysis (Severyn and Moschitti, 2015; Cambria, 2016; Poria et al., 2017; Lam et al., 2018), sentiment embeddings (Tang et al., 2014; Barnes et al., 2018), and aspect extraction (Poria et al., 2016; He et al., 2017; Pablos et al., 2018).

7 Conclusion

In this paper, we propose a simple but effective specificity-driven cascading approach for unsupervised sentiment modification. The proposed approach performs target sentiment addition and content reconstruction independently, so that more specific information is preserved. Extensive experimental results show that our approach outperforms several competitive systems by a large margin. Further analysis demonstrates that the proposed method not only increases the specificity of the generated text, but also further improves content preservation.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde. 2018. Bilingual sentiment embeddings: Joint projection of sentiment across languages. In *ACL*, pages 2483–2493.

Erik Cambria. 2016. Affective computing and sentiment analysis. *IEEE Intelligent Systems*, 31(2):102–107.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *AAAI*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *ACL*, pages 388–397.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*, pages 507–517.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. volume 9, pages 1735–1780. MIT Press.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *ICML*, pages 1587–1596.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Wai Lam, Xin Li, Bei Shi, and Lidong Bing. 2018. Transformation networks for target-oriented sentiment classification. In *ACL*, pages 946–956.

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

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *NAACL-HLT*, pages 1865–1874.

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. *arXiv preprint arXiv:1905.10060*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Aitor García Pablos, Montse Cuadros, and German Rigau. 2018. W2VLDA: almost unsupervised system for aspect based sentiment analysis. *Expert Syst. Appl.*, 91:127–137.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *ICML*, pages 1310–1318.

Soujanya Poria, Erik Cambria, and Alexander F. Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowl.-Based Syst.*, 108:42–49.

Soujanya Poria, Haiyun Peng, Amir Hussain, Newton Howard, and Erik Cambria. 2017. Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis. *Neurocomputing*, 261:217–230.

Cícero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *ACL*, pages 189–194.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Aliaksei Severyn and Alessandro Moschitti. 2015. Twitter sentiment analysis with deep convolutional neural networks. In *SIGIR*, pages 959–962.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *NIPS*, 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. *Journal of Machine Learning Research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL*, pages 1555–1565.

Yulia Tsvetkov, Alan W. Black, Ruslan Salakhutdinov, and Shrimai Prabhumoye. 2018. Style transfer through back-translation. In *ACL*, pages 866–876.

Jingjing Xu, Xu Sun, Qi Zeng, Xuancheng Ren, Xiaodong Zhang, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *ACL*, pages 979–988.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P. Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *NeurIPS*, pages 7298–7309.

Yi Zhang, Jingjing Xu, Pengcheng Yang, and Xu Sun. 2018a. Learning sentiment memories for sentiment modification without parallel data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1103–1108.

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