Specificity-Driven Cascading Approach for Unsupervised Sentiment Modification

Pengcheng Yang¹, Junyang Lin², Jingjing Xu², Jun Xie¹, Xu Sun², Qi Su²
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.

<table>
<thead>
<tr>
<th>Input</th>
<th>Michael’s work exceeds my expectations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shen et al. (2017)</td>
<td>Service is terrible.</td>
</tr>
<tr>
<td>Fu et al. (2018)</td>
<td>I was very disappointed.</td>
</tr>
<tr>
<td>Xu et al. (2018)</td>
<td>The girls do horrible work.</td>
</tr>
</tbody>
</table>

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

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

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

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:

$$e_{t,i} = f(s_t, h_i)$$  \hspace{1cm} (3)

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

$$c_t = \sum_{i=1}^{m} \alpha_{t,i} h_i$$  \hspace{1cm} (5)

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

$$y_t \sim \text{softmax}(g(s_t))$$  \hspace{1cm} (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 $y^* = (y_1^*, \cdots, y_n^*)$. Specially, the objective is to minimize the cross-entropy loss:

$$\mathcal{L} = -\sum_{t=1}^{n} \log(p(y_t^* | y_{<t}, x))$$  \hspace{1cm} (8)

3 Methodology

3.1 Overview

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

<table>
<thead>
<tr>
<th>Module</th>
<th>Source Input</th>
<th>Target Output</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC$_{p}$</td>
<td>$x^+/x$</td>
<td>$+/-$</td>
<td>Separate content and sentiment</td>
</tr>
<tr>
<td>S2S$_{p}$</td>
<td>$e^+$</td>
<td>$s^+$</td>
<td>Generate positive sentiment words</td>
</tr>
<tr>
<td>S2S$_{n}$</td>
<td>$e^-$</td>
<td>$s^-$</td>
<td>Generate negative sentiment words</td>
</tr>
<tr>
<td>S2S$_{p}$</td>
<td>$(e^+, s^+)$</td>
<td>$x^+$</td>
<td>Generate positive sentence</td>
</tr>
<tr>
<td>S2S$_{n}$</td>
<td>$(e^-, s^-)$</td>
<td>$x^-$</td>
<td>Generate negative sentence</td>
</tr>
</tbody>
</table>

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

\[ \{ x^+_i \}_{i=1}^{n_1} \text{ and } D^- = \{ x^-_i \}_{i=1}^{n_2} \text{ to represent the positive corpus and negative corpus, respectively, where } x^+ (x^-) \text{ is a positive (negative) sentence. } D = D^+ \cup D^- \text{ denotes the complete corpus. The sentiment modification task aims to take the sentence } x^+ (x^-) \text{ as input and generate a sentiment-transferred sentence } \hat{x}^- (\hat{x}^+) \text{ 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 \( x = (x_1, \cdots, x_m) \), 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 \( \alpha_i \) 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 \( x \) 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,

\[
\alpha_i = \text{softmax}(v^T h_i) \quad (9)
\]

\[
h = \sum_{i=1}^{m} \alpha_i h_i \quad (10)
\]

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 \( x \) is calculated as:

\[
\bar{\alpha} = \frac{1}{m} \sum_{i=1}^{m} \alpha_i \quad (11)
\]

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 \( x \) can be separated into content words \( c \) and sentiment words \( s \).\(^3\)

### 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% F1 score.

\(^3\)Except for Section 2, we use \( c \) and \( s \) 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 $D^+ = \{x^+_i\}_{i=1}^{n_1}$; negative corpus $D^- = \{x^-_i\}_{i=1}^{n_2}$; complete corpus $D = D^+ \cup D^-$

1: Separation:
   2: Train the sentiment classifier $SC_c$ on corpus $D$
   3: Separate positive sentence $x^+$ in $D^+$ into $(c^+, s^+)$ using $SC_c$ to construct new positive corpus $N^+ = \{(c^+_i, s^+_i, x^+_i)\}_{i=1}^{n_1}$
   4: Separate negative sentence $x^-$ in $D^-$ into $(c^-, s^-)$ using $SC_c$ to construct new negative corpus $N^- = \{(c^-_i, s^-_i, x^-_i)\}_{i=1}^{n_2}$

5: Emotionalization:
   6: Train $S2S_{\theta_{ne}}$ using MLE on new positive corpus $N^+$
   7: Train $S2S_{\theta_{ne}}$ using MLE on new negative corpus $N^-$

8: Fusion:
   9: Train $S2S_{\theta_{ne}}$ using MLE on new positive corpus $N^+$
   10: Train $S2S_{\theta_{ne}}$ using MLE on new negative corpus $N^-$

The fusion module aims to perform content reconstruction by fusing content words $c$ and sentiment words $s$ to generate fluent emotional sentence $x$. Compared with the previous work, the decoder here has extra sentiment information $s$, thus eliminating the information gap at both ends of the 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 $x$ can be separated into $(c, s)$, where $c$ and $s$ represent the content words and sentiment words, respectively. Therefore, the new corpus $N^+ = \{(c^+_i, s^+_i, x^+_i)\}_{i=1}^{n_1}$ can be constructed by explicitly separating the positive sentence $x^+$. Similarly, $N^- = \{(c^-_i, s^-_i, x^-_i)\}_{i=1}^{n_2}$ can be constructed by separating the negative sentence $x^-$. The emotionalization module aims to automatically generate corresponding sentiment words according to the content. Since corpus $N^+$ (or $N^-$) provides pseudo-parallel data about content words $c^+$ (or $c^-$) and sentiment words $s^+$ (or $s^-$), here we use the Seq2Seq model to accomplish this goal. We train a Seq2Seq model $S2S_{\theta_{ne}}$ on the new positive corpus $N^+$, which is responsible for generating positive sentiment words $s^+$ based on the content $c^+$. Another Seq2Seq model $S2S_{\theta_{ne}}$ is trained on the new negative corpus $N^-$ to generate negative sentiment words $s^-$ based on content words $c^-$. Therefore, for the well-trained model $S2S_{\theta_{ne}}$ (or $S2S_{\theta_{ne}}$), given content words $c$, $S2S_{\theta_{ne}}$ (or $S2S_{\theta_{ne}}$) is able to generate corresponding positive (or negative) sentiment words $s^+$ (or $s^-$).

3.4 Fusion Module

The fusion module aims to perform content reconstruction by fusing content words $c$ and sentiment words $s$ to generate fluent emotional sentence $x$. Compared with the previous work, the decoder here has extra sentiment information $s$, 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.

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 $S2S_{\theta_{ne}}$ and $S2S_{\theta_{ne}}$ on the new positive corpus $N^+$ and negative corpus $N^-$, respectively. Each model aims to output $x$ based on the input $(c, s)$. 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 $c$ and sentiment words $s$ respectively to form the content-side context $c^i$ and the sentiment-side context $c^i_s$. The hidden state $s_{t+1}$ of the decoder at time-step $t+1$ is computed as follows:

$$s_{t+1} = \text{LSTM}(s_t, [c(y_t); c^i; c^i_s])$$ \ (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 $x^+$, the separation module can separate $x^+$ into content words $c^+$ and sentiment words $s^+$. The model $S2S_{\theta_{ne}}$ in the emotionalization module takes $c^+$ as input to generate corresponding negative sentiment words $s^-$. Finally, $(c^+, s^-)$ 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\_n\_f in the fusion module to generate a fluent negative sentence \( \hat{x}^- \).

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.

<table>
<thead>
<tr>
<th>Module</th>
<th>Yelp</th>
<th>Amazon</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC_φ</td>
<td>2</td>
<td>512</td>
</tr>
<tr>
<td>S2S_p_r_φ</td>
<td>(2, 3)</td>
<td>(256, 512)</td>
</tr>
<tr>
<td>S2S_p_n_φ</td>
<td>(2, 2)</td>
<td>(256, 512)</td>
</tr>
<tr>
<td>S2S_p_f</td>
<td>(2, 3)</td>
<td>(256, 512)</td>
</tr>
<tr>
<td>S2S_n_f</td>
<td>(2, 2)</td>
<td>(256, 512)</td>
</tr>
</tbody>
</table>

Table 4: Main hyper-parameters. For the LSTM layer and hidden size, we use (\( \cdot \), \( \cdot \)) to represent the hyper-parameter 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/
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 outperforms the baselines by a large margin, especially in content preservation.
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.

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.

Table 7: Ablation study. “EM” denote the emotionalization module.

<table>
<thead>
<tr>
<th></th>
<th>ACC</th>
<th>BLEU</th>
<th>G-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yelp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full model</td>
<td>68.65</td>
<td>47.42</td>
<td>57.06</td>
</tr>
<tr>
<td>w/o EM</td>
<td>52.40</td>
<td>40.24</td>
<td>45.92</td>
</tr>
<tr>
<td>Amazon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full model</td>
<td>56.30</td>
<td>49.83</td>
<td>52.97</td>
</tr>
<tr>
<td>w/o EM</td>
<td>44.90</td>
<td>41.37</td>
<td>43.10</td>
</tr>
</tbody>
</table>

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

Table 9: Outputs of different systems on the Yelp dataset.

Table 10: Outputs of different systems on the Amazon dataset.
Input: Wendy's is my favorite chain, but this location is the worst.
CAAE: Great food and the best food!
MDAE: This place had amazing stars!
SEAE: Undoubtedly is my favorite chain.
Proposal: Wendy's 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. “Wendy’s” 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


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


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.


