Learning Sentiment Memories for Sentiment Modification without Parallel Data

Yi Zhang, Jingjing Xu, Pengcheng Yang, Xu Sun
MOE Key Lab of Computational Linguistics, School of EECS, Peking University
{zhangyi16,jingjingxu,yang_pc,xusun}@pku.edu.cn

Abstract

The task of sentiment modification requires reversing the sentiment of the input and preserving the sentiment-independent content. However, aligned sentences with the same content but different sentiments are usually unavailable. Due to the lack of such parallel data, it is hard to extract sentiment independent content and reverse the sentiment in an unsupervised way. Previous work usually can not reconcile sentiment transformation and content preservation. In this paper, motivated by the fact the non-emotional context (e.g., "staff") provides strong cues for the occurrence of emotional words (e.g., "friendly"), we propose a novel method that automatically extracts appropriate sentiment information from the learned sentiment memories according to the specific context. Experiments show that our method substantially improves the content preservation degree and achieves the state-of-the-art performance. [1]

1 Introduction

Sentiment modification of natural language texts is a special task that connects sentiment analysis and natural language generation. It facilitates many NLP applications, such as news rewriting and automatic conversion of review attitude, which reduce the human effort. Sentiment modification presents two requirements: one is that the sentiment or the attitude of the text needs to be transformed to the opposite; the other is that the transformed text should maintain semantic relevance to the input text as much as possible.

Recently, there have been some researches which focus on the work of editing a sentence to alter specific attributes, like style and sentiment (Shen et al., 2017; Hu et al., 2017). Typically, the parallel data with the same content but different sentiment is not available. This line of work attempts to extract the attribute-independent content from a dense sentence representation by adversarial learning. However, it is hard to extract the attribute-independent content in such implicit ways, which makes these methods tend to generate input-irrelevant texts.

Most existing methods can not reconcile the performance of sentiment transformation and content preservation. Direct replacement of emotional words can keep the context but may lead to low-quality sentences. For example, given an input "*The food is cold like rock*", this method probably outputs "*The food is warm like rock*". State-of-the-art models using neural networks struggle to generate high-quality sentences. However, these models usually lead to poor content preservation. For instance, when the source text is "*This is a wonderful movie*", we expect an output like "*This movie is disappointing*". However, the generated sentence may be "*The waiters are very rude*", which has little relevance to the source text. In general, it is difficult to preserve semantic content and reverse the sentiment at the same time without parallel data.

To address this problem, we propose a novel model which performs well in both sentiment transformation and content preservation. Our model first learns two kinds of sentiment memories by explicitly separating emotional words. Then, according to the specific context, the model extracts appropriate sentiment information from the memory of target sentiment. The decoder takes the extracted memory and the context representation together to perform decoding. The overview of our model is shown in Figure 1. The main architecture of our model is a **S**entiment-**M**emory based **A**uto-**E**ncoder (**SMAE**). The proposed model achieves the state-of-the-art perfor-

[1] The code is available at https://github.com/lancopku/SMAE

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1103–1108
Brussels, Belgium, October 31 - November 4, 2018.

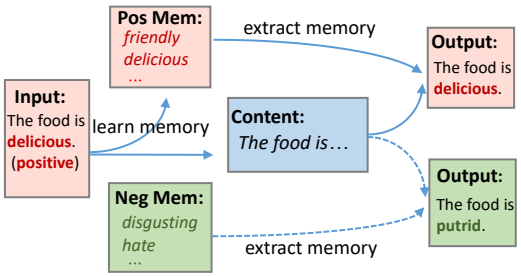

Figure 1: Illustration of the proposed model with a positive input. Solid and dashed lines indicate the training process and the testing process, respectively. The process with a negative input is in a similar way.

mance, especially improves content preservation degree by a large margin.

Our contributions are concluded as follows:

- We propose a method that uses sentiment memories to accomplish sentiment modification without any help of the parallel data.
- The proposed method improves the content preservation degree by a large margin when compared with current systems.

2 Related Work

Recently, there has been some studies for sentiment modification. Shen et al. (2017) learn an encoder that maps a sentence with its original style to a style-independent content representation. This is then passed to a style-dependent decoder for rendering. Fu et al. (2017) implement a multi-decoder auto-encoder (Bengio et al., 2009; Dai and Le, 2015) where the encoder is used to capture the content and the sentiment-specific decoders are used to generate target sentence. Hu et al. (2017) augment the unstructured variables z in vanilla VAE with a set of structured variables c each of which targets a salient and independent semantic feature of sentences, to control sentence sentiment. However, all of these work attempt to implicitly separate the non-emotional content from the emotional information in a dense sentence representation. Xu et al. (2018) explicitly filter out emotional words. They use two sentiment-specific decoders to attach sentiments to non-emotional context. The decoders bear all the burdens to generate sentiments. In our model, we use sentiment memories to assist generating sentiments with only one decoder, which results in fewer parameters.

The proposed sentiment-memory based auto-encoder (Bengio et al., 2009; Ma et al., 2018b) learns the idea of memory network (Weston et al., 2014; Sukhbaatar et al., 2015) but simplifies the process. Our work is also related to the generation tasks (Wang et al., 2017; Liu et al., 2018; Ma et al., 2018a; Lin et al., 2018). These tasks usually generate texts that preserve main information of input texts.

3 Proposed Model

We first use a variant of self-attention(Lin et al., 2017; Kim et al., 2017) mechanism to distinguish the emotional and non-emotional words. Then the positive words and negative words are used to update the corresponding memory modules. Finally, the decoder uses the target sentiment information extracted from the memory and the content representation to perform decoding.

3.1 Emotional Words Detection Model

We first find the emotional words that have the most discriminative power for sentiment polarity. This work is done by training a sentiment classifier with a simple self-attention mechanism. Here the sequence of inputs $\{h_1, ..., h_T\}$ are the hidden states of a LSTM, running over the words in the source sentence $\{x_1, ..., x_T\}$. The context vector can then be computed using a simple sum:

$$\boldsymbol{c} = \sum_{t=1}^{T} a_t \cdot \boldsymbol{h}_t \quad (1)$$

where a_t denotes the attention weight of the t-th word. The sentence vector $\boldsymbol{c}$ is then fed into a fully connected layer to predict the sentiment polarity of the source text. Since the words with obvious emotional tendencies will be given greater weights compared to those non-emotional words during training, a_t can be used to distinguish between emotional and non-emotional words.

The weights of standard attention mechanism sum to 1. When there are several emotional words, the sum 1 is distributed by these words. However, we expect that each emotional word has a weight close to 1 to identify its sentiment attribute. Hence, following (Kim et al., 2017), we modify the calculation of attention weights as follows to get more distinguishable weights:

$$a_t = \text{sigmoid}(\boldsymbol{v}^T \boldsymbol{h}_t) \quad (2)$$

where $\boldsymbol{v}$ is the parameter vector. The sigmoid function follows our intention that giving each in-

put word a distinguishable weight which is close to 1 or 0. However, these weights falls between 0 and 1. They still can not thoroughly distinguish the emotional words from non-emotional words without redundant information. Following Xu et al. (2018), we map attention weights to discrete values, 0 or 1, and we adopt their discrete method. The weights greater than the averaged attention value are assigned to 1 and the weights less than the averaged attention value are assigned to 0. The weight a_t after discretization is denoted as $\hat{a}_t$. Then, $\hat{a}_t$ can be regarded as the emotional word identifier. $1 - \hat{a}_t$ becomes non-emotional word identifier.

3.2 Sentiment-Memory Based Auto-Encoder

After the separation of emotional and non-emotional words, the proposed **SMAE** is used to process these two kinds of information. We employ the seq2seq based auto-encoder. Both the encoder and the decoder are LSTM networks (Hochreiter and Schmidhuber, 1997).

If $\boldsymbol{x}_i$ is a context word, then $\hat{a}_i$ is 0, causing $(1-\hat{a}_i)\boldsymbol{x}_i$ to be $\boldsymbol{x}_i$. Therefore, the sequence $\{(1-\hat{a}_1)\boldsymbol{x}_1, \cdots, (1-\hat{a_T})\boldsymbol{x}_T\}$ can be regarded as non-emotional word embedding sequence. It is fed into the LSTM encoder sequentially. we select $\boldsymbol{h}_T$ in the last state tuple $(\boldsymbol{h}_T, \boldsymbol{c}_T)$ of the encoder as the content representation of the input.

Meanwhile, the embeddings of the emotional words of the source text are used to update the sentiment-memory. Since we have two kinds of sentiments, positive and negative, we use $\boldsymbol{M}^{pos} \in \mathbb{R}^{e\times\gamma}$ and $\boldsymbol{M}^{neg} \in \mathbb{R}^{e\times\gamma}$ to denote the positive memory and the negative memory, respectively. e is the embedding size and γ is a hyper-parameter which controls the size of the memory.

We illustrate the following part by using positive input as an example. We first sum the embedding of the emotional words to get a vector representation of the emotional information, which is denoted as $\boldsymbol{s}^{pos} \in \mathbb{R}^e$. We then use a simple attention mechanism to find the columns in $\boldsymbol{M}^{pos}$ that are most closely related to the emotional information. The outer product of the transposition of emotional information $\boldsymbol{s}^{pos}$ and the attention weights $\boldsymbol{w}$ broadcasts the sentiment vector $\boldsymbol{s}^{pos}$ to a matrix. Then, the matrix is added to the existing memory $\boldsymbol{M}^{pos}$. Due to the attention weight $\boldsymbol{w}$, the columns that are most closely related to the emotional information are updated more with the sentiment information $\boldsymbol{s}^{pos}$. Formally, we have:

$$\boldsymbol{s}^{pos} = \sum_{i=1}^{T} \hat{a}_i \cdot \boldsymbol{x}_i \tag{3}$$

$$\boldsymbol{w} = softmax\left((\boldsymbol{s}^{pos})^T \boldsymbol{M}^{pos}\right) \tag{4}$$

$$\boldsymbol{M}^{pos} = \boldsymbol{M}^{pos} + \boldsymbol{s}^{pos} \otimes \boldsymbol{w} \tag{5}$$

where $\otimes$ denotes the outer product.

Previous work employ two sentiment-specific decoders to generate text based on the supposed non-emotional representation. The decoders bear all the burdens to generate sentiments. In our model, we extract some sentiment information from the sentiment-memories to assist decoding. Intuitively, the context word "staff" is more likely to be associated with the emotional word "friendly", and "food" is more likely to be associated with "delicious". So we use the context vector $\boldsymbol{s}^{con}$ to extract the corresponding sentiment memory that is more likely to be used in the future decoding. The context vector $\boldsymbol{s}^{con}$ is represented as the sum of the embedding of non-emotional words. Then $\boldsymbol{s}^{con}$ is used to compute the attention weights $\boldsymbol{u}$ over the columns of sentiment memory matrix. We sum these weighted columns as the extracted memory $\tilde{\boldsymbol{m}}$ and add $\tilde{\boldsymbol{m}}$ to the last cell state $\boldsymbol{c}_T$ of the encoder:

$$\boldsymbol{s}^{con} = \sum_{i=1}^{T} (1 - \hat{a}_i) \cdot \boldsymbol{x}_i \tag{6}$$

$$\boldsymbol{u} = softmax\left((\boldsymbol{s}^{con})^T \boldsymbol{M}^{pos}\right) \tag{7}$$

$$\tilde{\boldsymbol{m}} = \sum_{j=1}^{\gamma} u_j \cdot \boldsymbol{M}_j^{pos} \tag{8}$$

$$\tilde{\boldsymbol{c}}_T = \boldsymbol{c}_T + W\tilde{\boldsymbol{m}} \tag{9}$$

where u_j denotes the j-th value in vector $\boldsymbol{u}$, $\boldsymbol{M}_j^{pos}$ denotes the j-th column of $\boldsymbol{M}^{pos}$ and W is the parameter matrix. The new tuple $(\boldsymbol{h}_T, \tilde{\boldsymbol{c}}_T)$ then acts as the initial state of the decoder.

The negative input is processed in the same way. At the training stage, the decoder is encouraged to restore the source text. Therefore, the cross entropy loss function is optimized.

4 Experiments

4.1 Data Preprocessing

We use the Yelp Review Dataset (Yelp) provided by Yelp Dataset Challenge[2] to conduct experiments. Each item is a sentence from the review

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

Model	ACC	BLEU
CEA	71.96	2.77
MAE	74.59	5.45
SMAE	**76.64** (+2.05)	**24.00** (+18.55)

Table 1: Performance of the proposed method and state-of-the-art systems.

Model	Sentiment	Content	Fluency
CAE	6.55	4.46	5.98
MAE	6.64	4.43	5.36
SMAE	6.57	5.98	6.69

Table 2: Results of human evaluation.

Input: *Very helpful and informative staff!* **CAE**: *Worst service ever.* **MAE**: *Very nice here and poor!* **Proposed**: *Very rude and careless staff !*
Input: *I will never go here again.* **CAE**: *I love this place here!* **MAE**: *I had say this place here.* **Proposed**: *I will never go anywhere else.*
Input: *The worst and would never recommend anyone to use them.* **CAE**: *The best place I 've been to go here!* **MAE**: *The first experience is so happy and nice.* **Proposed**: *The best and would definitely recommend anyone to use them.*

Table 3: Examples generated by the proposed method and baselines. In comparison, our model changes the sentiment of inputs with higher semantic relevance.

on Yelp and is labeled as having either negative or positive sentiment. We train a CNN sentence classifier (Kim, 2014) to filter examples with ambiguous sentiment polarities (category probability $<$ 0.8). The processed dataset contains 510K, 20K, and 20K pairs for training, validation, and testing, respectively. The classifier achieves an accuracy of 94% on the processed dataset and is also used to test transformation accuracy.

4.2 Experiment Settings

We tune our hyper-parameters on the development set. The word embeddings are initialized randomly with a size of 128. The hidden size of the sentiment-memory based auto-encoder is 300. We use Adam optimizer (Kingma and Ba, 2014) with an initial learning rate set to 0.001 to train our model and the batch size is set to 64. The hyper-parameter γ which controls the size of memory matrix is 60.

4.3 Baselines

We compare our proposed method with two state-of-the-art systems that have been used for sentiment modification. We run the released code on our dataset.

Cross-aligned Auto-Encoder (CAE): This system, proposed by Shen et al. (2017), uses a shared latent content space across different sentiments and leverages refined alignment of latent representations to perform sentiment modification.

Multi-decoder Auto-Encoder (MAE): This system is proposed by Fu et al. (2017). They use a multi-decoder seq2seq model (Bengio et al., 2009; Dai and Le, 2015) where the encoder captures content information by adversarial learning (Goodfellow et al., 2014) and the sentiment-specific decoders are used to generate target sentences.

4.4 Results and Discussions

We use ACC to denote the transformation accuracy. Following Gan et al. (2017), we also compute BLEU (Papineni et al., 2002) between the output and the source text to evaluate the content preservation degree. A high BLEU score primarily indicates that the system can correctly preserve content by retaining the same words from the source sentence.

The experimental results of our proposed model and the baselines are shown in Table 1. Both baseline models have low BLEU score but high accuracy, which indicates that they may be trapped in a situation that they simply output a sentence with the target sentiment regardless of the content. The main reason is that these methods using adversarial learning attempt to implicitly separate the emotional information from the context information in a sentence vector. However, without parallel data, it is difficult to achieve such a goal. Our proposed SMAE model takes advantage of self-attention mechanism and explicitly removes the emotional words, leading to a significant improvement of content preservation and the state-of-the-art performance in terms of both metrics.

We also involve human evaluation to measure the quality of generated text. Each item contains an input and three outputs generated by different systems. Then 200 items are distributed to 2 annotators with linguistic background. The annota-

Models	ACC	BLEU
SMEA	76.64	24.00
SMEA (w/o memories)	14.08	26.09

Table 4: Ablation test of memory module.

The staff here is **very rude**.
It really is n't **worth coming** here .
Very pleased with this business.
Been here once and **loved** going **here**.

Table 5: The effectiveness of the memory module with examples. The red words are absent in the input but generated with the help of sentiment memories.

tors have no idea about which system the output is from. They are asked to score the output on three criteria on a scale from 1 to 10: the transformed sentiment degree, the content preservation degree, and the fluency. Table 2 shows the evaluation results. Our model has obvious advantage over the baseline systems in content preservation, and also performs well in other aspects.

Several randomly selected examples generated by different models are shown in Table 3. These examples clearly show our proposed model can generate sentences that are more semantically relevant to the input text compared to the baselines.

4.5 Effectiveness of Sentiment-Memories

To verify the effectiveness of the memory module of our model, we conduct ablation study by excluding the sentiment-memory module. The result is shown in Table 4. According to the result, the complete model achieves an improvement of 62.56% on transformation accuracy over the model that excludes the sentiment memories, which means the sentiment memories are key components to ensure successful sentiment modification. In addition, several examples are shown in Table 5 to visually demonstrate the effectiveness of the memory module. we can find that the proposed model is capable of generating appropriate emotional words (red words in Table 5) to adapt different contexts.

4.6 Error Analysis

To better interpret our model, we also analyze the failure examples whose sentiments are not transformed. We observe that in most cases, these inputs do not have emotional tendencies. Although we have filtered the sentiment-ambiguous examples in preprocessing, there are still a few ambiguous inputs such as "What can I say ?" and "Been here twice.". Since our model tries to preserve non-emotional content. These words are easily kept and then the decoder barely depends on sentiment-memories. Thus, it is difficult to handle the sentiment transformation with these examples.

5 Conclusion

In this paper, we propose a model that first learns sentiment memories without parallel data and then automatically extracts sentiment information to adapt different contexts when decoding. Experimental results show that our method substantially improves the content preservation degree and achieves the state-of-the-art performance.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (No. 61673028). We thank all the reviewers for providing the constructive suggestions. Xu Sun is the corresponding author of this paper.

References

Yoshua Bengio et al. 2009. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3079–3087.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2017. Style transfer in text: Exploration and evaluation. *CoRR*, abs/1711.06861.

Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. 2017. Stylenet: Generating attractive visual captions with styles. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 955–964.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Controllable text generation. *CoRR*, abs/1703.00955.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. 2017. Structured attention networks. *CoRR*, abs/1702.00887.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Junyang Lin, Xu SUN, Shuming Ma, and Qi Su. 2018. Global encoding for abstractive summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, pages 163–169.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*.

Shuming Ma, Xu Sun, Junyang Lin, and Xuancheng Ren. 2018a. A hierarchical end-to-end model for jointly improving text summarization and sentiment classification. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 4251–4257.

Shuming Ma, Xu SUN, Junyang Lin, and Houfeng WANG. 2018b. Autoencoder as assistant supervisor: Improving text representation for chinese social media text summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, pages 725–731.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. *CoRR*, abs/1705.09655.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. *CoRR*, abs/1503.08895.

Kexiang Wang, Tianyu Liu, Zhifang Sui, and Baobao Chang. 2017. Affinity-preserving random walk for multi-document summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 210–220.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *CoRR*, abs/1410.3916.

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.