[SCIS] Decision on SCIS-2022-0232.R2 - Accept

2022-11-03 15:35:01

发件人： "Fei Song" <onbehalfof@manuscriptcentral.com> (由 010101843c6a2966-ba7d689f-b70d-4057-8d68-8e7c20d8fc28-000000@outbound.manuscriptcentral.com 代发)

收件人： linfuqiang13@alumni.nudt.edu.cn

抄 送： linfuqiang13@alumni.nudt.edu.cn "宋伊萍" <songyiping@nudt.edu.cn> ztianac@cse.ust.hk chenwangqun19@nudt.edu.cn ddw_bak@nudt.edu.cn

03-Nov-2022
Manuscript ID: SCIS-2022-0232.R2

Dear Dr. Fuqiang Lin:

Your manuscript "Memory-Enhanced Text Style Transfer with Dynamic Style Learning and Calibration" which you submitted to the SCIENCE CHINA Information Sciences, has been accepted.

To revise your manuscript, log into https://mc03.manuscriptcentral.com/scis and enter your Author Center, where you will find your manuscript title listed under "Manuscripts with Decisions". Under "Actions", click on "Create a Revision". Your manuscript number has been appended to denote a revision.

You may also click the below link to start the revision process (or continue the process if you have already started your revision) for your manuscript. If you use the below link you will not be required to login to ScholarOne Manuscripts.

*** PLEASE NOTE: This is a two-step process. After clicking on the link, you will be directed to a webpage to confirm. ***

https://mc03.manuscriptcentral.com/scis?URL_MASK=a8133848d5cb48aebd051e84464fb529

When submitting your revised manuscript, you will be able to respond to the comments made by the reviewers and editors in the space provided. You can use this space to document any changes you make to the original manuscript. In order to expedite the processing of the revised manuscript, please be as

SCIENCE CHINA
Information Sciences

• **RESEARCH PAPER** •

Memory-Enhanced Text Style Transfer with Dynamic Style Learning and Calibration

Fuqiang LIN[1], Yiping SONG[1*], Zhiliang TIAN[2], Wangqun CHEN[1],
Diwen DONG[1] & Bo LIU[1*]

[1] *National University of Defense Technology, Changsha* 410073*, China;*
[2] *The Hong Kong University of Science and Technology, Hong Kong* 999077*, China*

Abstract The goal of text style transfer is to rephrase a sentence to match the desired style while retaining the original content. As a controllable generation task, mainstream approaches use content-independent style embedding as control variables to guide stylistic generation. Nonetheless, stylistic properties are sensitive to the context even under the same style. For example, both “delicious” and “helpful” convey *positive* sentiment, while they are more likely to describe food and person, respectively. Therefore, desired style signals require to vary with the content. To the end, we propose a memory-enhanced transfer method, which learns fine-grained style representation concerning content to assist transfer. Rather than employing static style embedding or latent variables, our method abstracts linguistic characteristics from training corpora and memorizes subdivided content with corresponding style representations. The style signal is dynamically retrieved from memory using the content as a query, which provides a more expressive and flexible latent style space. To tackle the imbalance of quantity and quality under different content, we further introduce a calibration method to augment the construction of memory by modeling the relationship between styles. Experimental results on three benchmark datasets verify the superior performance of our model compared to competitive approaches. The evaluation metrics and case study also indicate that our model can generate diverse stylistic phrases matching context.

Keywords style transfer, memory-enhanced method, text generation, deep learning, text representation

Citation Lin F Q, Song Y P, Tian Z L, et al. Memory-Enhanced Text Style Transfer with Dynamic Style Learning and Calibration. Sci China Inf Sci, for review

1 Introduction

As an essential task of controllable text generation, text style transfer (TST) aims to modify the stylistic attributes (e.g., sentiment, genre, and formality) of text while maintaining underlying content. TST has been a research problem of interest due to broad applications, such as sentiment modification [21, 37], stylistic summarization [8], and text simplification [4]. Because of the difficulty of collecting parallel corpus, related research is typically conducted in the unsupervised learning setting.

To control style attributes of text, dominant approaches learn one individual style embedding or static latent style variables, and leverage such style signals for guiding transfer. One line of methods disentangles text into separated style and content representations, and applies a style-specific decoder to conduct transfer conditioned on non-stylistic content and desired style embedding. Representative works [14, 17, 30, 42] adopt adversarial discriminators on the latent space to achieve disentanglement. Following the trend, several methods [20, 21, 31] apply pipeline word-level processing that first obtains content-only sentence by explicitly removing stylistic tokens, and then merges it with target style signals for transfer. To enhance content preservation, another line [5, 39] proposes to encode text into entangled representation without explicit disentanglement, and incorporates the style embedding to attention-based structure for style control. Typically, Dai et al. [5] make no assumption of disentanglement and apply the Transformer architecture with attention mechanisms to learn style transfer, which achieves considerable improvement on content preservation.

[*] Corresponding author (email: songyiping@nudt.edu.cn, kyle.liu@nudt.edu.cn)

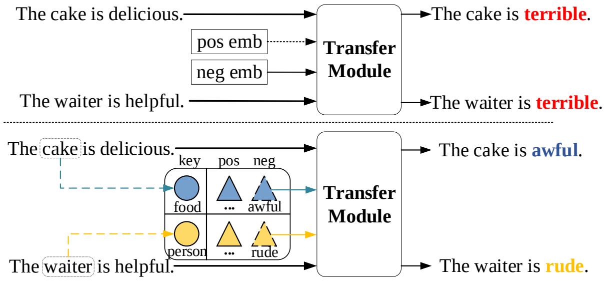


Figure 1 Illustration of difference in style signal between most previous methods (top) and our method (bottom).

Nevertheless, as the mainstream form of style signal, simple content-independent style embedding is insufficiently expressive to project the concept of style, especially not flexible enough when applied to various topics of content. Due to the deep fusion of content and style in text [19], style properties, e.g., style-related phrases, vary with the content of the text [15]. Take an example of restaurant reviews with the same *positive* sentiment, comparing with employing “great” to describe all objects, it is more reasonable to use stylistic tokens “kind”, “prompt”, and “delicious” for the description of “owner”, “service”, and “food”, respectively. That is, the desired style representation requires to match the semantics of the content. However, previous work overemphasizes learning content-dependent style representations, which is less sensitive to the content and leads to tired style-related phrases in the transferred sentences.

To address these issues, we propose a memory-enhanced transfer method to provide fine-grained content-dependent style signals. Instead of employing individual style embedding, our model extracts useful linguistic features from the training corpus, thereby constructing a content-style memory to create expressive and flexible style representations. In particular, we cluster the training corpus into multiple groups according to their content and extract attribute features for each style by the group. As illustrated in Figure 1, we memorize the common characteristics in the form of content-styles pair. Compared to solo style embedding, style representations in memories are more expressive and are fetched dynamically with respect to content.

Besides, due to the skew distribution of training samples, the quality of different memory units is uneven. Significantly, the memory unit with only a few samples is prone to suffer from error deviations of style representation caused by outliers. To this end, we further introduce a calibration method to optimize the style representations in memory. Inspired by the TransE [3], we consider the relationship between styles (i.e. value cells in the memory) to share a similar distribution across all memory units. Therefore, our method learns a standard relationship representation and uses it to calibrate the value cells of memory. In this way, our method can retrieve diverse stylistic phrases to match the content of input sentences, which contributes to improving content preservation and style transfer strength.

Our contributions can be summarized as follows:

1. We extract linguistic information from the training corpus using a content-style memory, thereby learning content-dependent style representations, which contributes to improving style control as well as enhancing content preservation.

2. We propose to learn the relationship between styles and use it as shared criteria to calibrate the memory, which deals with the uneven quality of different memory units.

3. Experimental results demonstrate that our proposed method generally outperforms state-of-the-art approaches on two benchmark style transfer tasks. Specifically, Our model achieves a better trade-off between style transfer accuracy and content preservation, and generates diverse stylistic phrases that vary with context.

2 Related work

2.1 Unsupervised Text Style Transfer

The method of representing content and style signal is the core topic of most existing works. In light of this criterion, we can roughly divide most previous studies into two directions.

The first line of approaches (Disentanglement-based) follows style transfer in the computer vision field [10] to first strip style from text, and then fuse the style-independent content and the target style for transfer. Fu et al. [9] systematically explore two methods, multi-decoder model and style-embedding model, with the common ground to strip style information of the input sentence by the auto-encoder seq2seq model. To achieve further improvement, iterative optimization, reinforcement learning, and a series of all-around losses are applied to guarantee the quality of disentanglement [16,17,24]. Different from end-to-end training, there are also attempts to adopt a two-step pipeline for word-level disentanglement. These methods identify stylistic words by pre-trained discriminators and then replace them with words carrying the desired style [20,21,25,41]. Based on the assumption that stylistic properties of sentences are automatically diluted when translating to another language, another method for disentanglement utilizes unsupervised neural machine translation (UNMT) models with the back-translation strick [27,42]. Due to impracticable complete disentanglement, this paradigm performs well in transfer accuracy but suffers from poor content preservation [5,15]. For this sake, our work replaces the disentanglement constraint with two auxiliary tasks to make two latent representations biased towards modeling content and style information, respectively.

Instead of disentangling content and style separately, another line (Attention-based) directly edits an entangled latent representation and relies on the generator to rewrite the original stylistic information with the desired attributes [5,12,23,33,39]. Specifically, Yi et al. [39] learn latent style space from multiple instances via the generative flow method and combine it with an attention-based sequence-to-sequence structure for enhancing content preservation. Liu et al. [23] adopt the pre-trained language model GPT-2 with semantic similarity metrics as a direct reward for the stylistic generation. Generally, one significant advantage of this line is to avoid the loss of content information caused by disentanglement and thereby better preserve content information. Nevertheless, the style-specific decoders bear all the burdens to "overwrite" the original style in the entangled latent representation, which results in unsatisfactory style transfer strength [15,39].

Regardless of content representation, both lines employ content-independent embeddings or static latent variables to represent style, which is insufficient due to the limited capacity of individual vectors [36]. To tackle this problem, Xiao et al. [36] recently propose a TranSductive Style Transfer (TSST) model to employ a context-aware style representation by employing a retriever. Both our method and TSST verify that learning fine-grained content-dependent style characteristics is of great help to altering style while retaining content. Nevertheless, there are two non-trivial differences. First, TSST only fetches top-k relevant samples with the target style to facilitate style representation construction, while our method takes the insights from information retrieval and integrates all training samples via clustering to learn more expressive and flexible style representations. Second, our method considers the skew distribution on semantics space and proposes to calibrate stylistic properties through modeling the relationship between styles, which potentially provides further performance gains.

2.2 Memory-enhanced Text Generation

Our work is also related to memory-enhanced text generation methods [34, 35, 38, 41], which memorize external information to assist generation. The common source of information for memory construction includes internal knowledge extracted from corpora and external knowledge from structured bases. For example, memory augmented frameworks are widely adopted to project dialogue history in dialogue systems [35], while topic-to-essay generation systems integrate external knowledge base into the generator through dynamic memory mechanism [38]. In this paper, we follow the former that automatically learns useful memory information from the training corpus. The workflow is to store external memory into embedding vectors to form a memory matrix, update during training, and adopt query vectors to extract related memory as a part of the input for the generator [40].

A similar work to ours is SMAE [41], which learns sentiment memories in the form of style vectors to assist sentiment modification. The main differences are: 1) SMAE constructed memory based on the strong assumption that all words are either style-related or irrelevant, while our model does not require this assumption. In contrast, we propose a content-style memory module to memorize content-style pairs, as well as their dependencies. In this way, our model can extend to a range of complex linguistic styles, such as sentiment, formality, and authorship. 2) Our model addresses the imbalance sample quantity and quality problem via memory calibration, but SMAE did not. The skew distribution of training samples is likely to make the learned memory module of low quality, especially memory slots extracted from

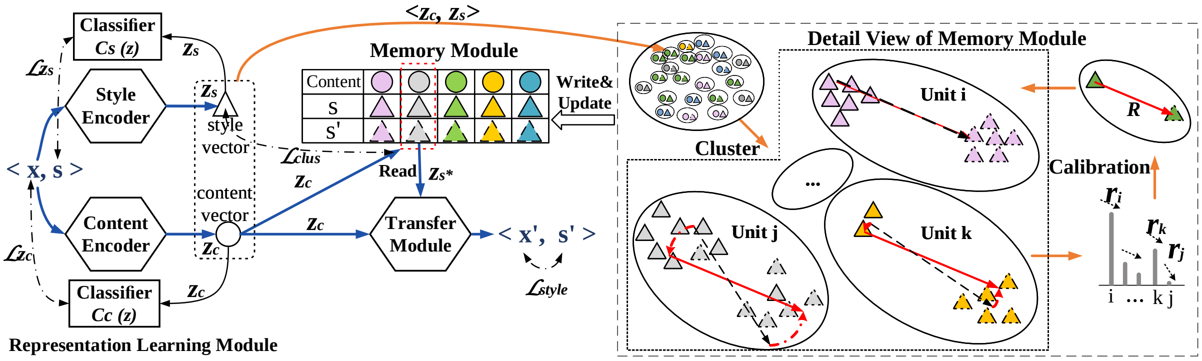


Figure 2 The architecture of our model. Blue arrows show the transfer process, and yellow arrows illustrate memory construction and calibration processes. $\langle \mathbf{x}, s \rangle$ indicates the input sequence $\mathbf{x}$ with its style s, and $\langle \mathbf{x}', s' \rangle$ is the transferred output $\mathbf{x}'$ with desired style s'. The content vector $\boldsymbol{z_c}$ is used as a query to fetch desired style signal from memory to assist transfer. The dotted box illustrates how to memorize information by clustering, and learn relationships to calibrate memory. The dashed arrows show the training objectives of our method. We omit the reconstruction and cycle reconstruction tricks, i.e. $\mathcal{L}_{self}$ and $\mathcal{L}_{cycle}$, for simplicity.

few-shot samples. To the end, we further incorporate an alignment constraint on the embedding space to augment these weak slots.

3 Methodology

3.1 Task Definition

Suppose there is a training corpus $\mathcal{X} = \{(\mathbf{x}_i, s_i)\}$, where each instance indicates a sentence $\mathbf{x}_i$ with its labeled style attribute s_i (e.g., sentiment, formality). Specifically, given a piece of sentence $\mathbf{x}$ carrying the original style s, the goal of text style transfer is to learn a transfer model $f_\theta(\mathbf{x}, s')$ to generate a sentence $\mathbf{x}'$ with the different desired attribute s' while maintaining content unchanged. Note that the dataset is non-parallel in our experiment setting, which means we have no access to ground truth sentence $\mathbf{x}'$.

3.2 Model Architecture

As illustrated in Figure 2, our model consists of three components: a latent representation learning module, a content-style memory module, and a memory-enhanced transfer module. The latent representation learning module maps text into separated content and style representations via two auxiliary tasks. Aside from using as the input of the transfer module, the learned latent representations are also collected for memory construction. The details of the content-style memory module will be emphatically introduced in the latter subsections (3.3 & 3.4). The memory-enhanced transfer module fetches information from memory and merges it with the content of the input sequence to achieve transfer.

3.2.1 *Latent Representation Learning Module*

The goal of the latent representation learning module is to model underlying latent factors of content and style. We adopt two encoders, i.e. content encoder and style encoder, with bidirectional GRU to encode the input sentence $\mathbf{x}$ into the two latent representations, namely the content vector $\boldsymbol{z_c}$ and style vector $\boldsymbol{z_s}$. To ensure $\boldsymbol{z_c}$ projects content information while $\boldsymbol{z_s}$ projects style information from the input sentence, we adopt multi-task learning and introduce two auxiliary tasks as supervision for training. The common intention is that an expressive latent representation can project and recover corresponding linguistic features or properties.

For style representation learning, a simple practice to facilitate modeling stylistic information is to conduct an additional classification based on $\boldsymbol{z_s}$ for the prediction of style-oriented attributes. Since each sentence $\mathbf{x}$ are in pair with its style label s, we apply a simple softmax classifier $C_s(\boldsymbol{z})$ on the style representation $\boldsymbol{z_s}$ to predict the label. Similar as most classification tasks, the training objective is to minimize the cross-entropy loss:

$$\mathcal{L}_{z_s} = -\mathbb{E}_{(\mathbf{x},s)\sim\mathcal{X}}[\log p_{C_s}(s \mid \boldsymbol{z_s})]. \tag{1}$$

For content representation learning, we adopt a self-supervised learning method for the content-oriented constraint due to the lack of ready-made labels. Concretely, the content information is referred to as bag-of-words (BoW) features, and the auxiliary task is to recover such feature based on content vector $\boldsymbol{z_c}$. Since nouns in the text usually remain unchanged during transfer, we choose nouns as candidates. In practice, our method constructs an extra noun vocabulary $\mathbb{V}_{noun}$, which consists of all nouns extracted from our training corpus. Given a sentence $\mathbf{x}$ containing N noun words, the BoW probability is calculated as $p_c\,(w) = \frac{\sum_{i=1}^{N} \mathbb{I}\{w_i = w\}}{N}$ for each word w in vocabulary $\mathbb{V}_{noun}$, where $\mathbb{I}\{\cdot\}$ is the indicator function. Similarly, we adopt a softmax classifier $C_c(\boldsymbol{z})$ to predicts the BoW distribution on $\boldsymbol{z_c}$. The loss function is the cross-entropy loss against the automated BoW feature $p_c(\cdot)$:

$$\mathcal{L}_{z_c} = -\mathbb{E}_{(\mathbf{x},s)\sim\mathcal{X}}[\log p_{C_c}(p_c \mid \boldsymbol{z_c})]. \tag{2}$$

In the training stage, the $\langle \boldsymbol{z_c}, \boldsymbol{z_s} \rangle$ pairs are fed to the transfer module for rendering and also collected for the memory construction.

3.2.2 *Memory-Enhanced Transfer Module*

The transfer module reads out fine-grained style signals from memory to control the style attributes of generated text. We denote the memory as $\boldsymbol{M} = \{\boldsymbol{m_1}, \cdots, \boldsymbol{m_K}\}$, where $\boldsymbol{m_i}$ is a memory unit with learned content information as key cell and style information as value cells. Given the content vector $\boldsymbol{z_c}$ of the sentence $\mathbf{x}$, the transfer module use $\boldsymbol{z_c}$ as a query to fetch the corresponding desired style signal $\boldsymbol{z_{s^*}}$ from memory. After that, we concatenate $\boldsymbol{z_c}$ and $\boldsymbol{z_{s^*}}$ as the initial states of the decoder. Formally,

$$p_\theta(\mathbf{x}' \mid \mathbf{x}, \mathbf{s}') = \prod_{t=1}^{L} p_\theta(x'_t \mid \boldsymbol{z'}, x'_1, \ldots, x'_{t-1}, \mathbf{x}), \tag{3}$$

where θ represents the parameters of the transfer module, and $\boldsymbol{z'}$ is the concatenation of $\boldsymbol{z_c}$ and $\boldsymbol{z_{s^*}}$.

3.3 Content-Style Memory Construction

Our memory module aims to learn and store fine-grained style signals corresponding to different topics of content. Style in the text is a complex concept that even to the same style attribute, stylistic words vary with different contexts. For this sake, we group the training samples into K clusters according to the semantics of content and store the common linguistics of these clusters in the content-style memory to assist transfer. The expectation is that each memory unit can cover a series of transfer cases associated with a highly related topic (e.g., “cake”, “steak”, and “chicken” under the “food” topic), and also provide corresponding style signals to generate topic-related stylistic rephrases (e.g., “delicious” for positive sentiment, and “tasteless” for negative).

To the end, our memory $\boldsymbol{M} = \{\boldsymbol{m_1}, \boldsymbol{m_2}, \cdots, \boldsymbol{m_K}\}$ employs a one-key-two-value structure. That is, there are K memory units, and each unit $\boldsymbol{m_i} = \langle \boldsymbol{k_i}, \boldsymbol{v_i^s}, \boldsymbol{v_i^{s'}} \rangle$ consists of a key cell $\boldsymbol{k_i}$ and two value cells $\langle \boldsymbol{v_i^s}, \boldsymbol{v_i^{s'}} \rangle$. The key cell is expected to memorize the linguistic characteristics of a specific category of content, and two value cells represent corresponding content-dependent style signals with respect to the candidated styles, i.e. source style s and target style s', respectively.

3.3.1 *Memory Writing*

To construct a set of flexible and expressive content-dependent style representations, we cluster training samples in the corpus into K groups according to the semantics of the content. Concretely, we collect $\langle \boldsymbol{z_c}, \boldsymbol{z_s} \rangle$ pairs from latent representation learning module, and employ the K-Means algorithm to group all pairs into K clusters by $\boldsymbol{z_c}$. Take the i-th cluster C_i as an example, we regard the mean of $\boldsymbol{z_c}$ over all pairs in this cluster as the center and utilize it as the key cell $\boldsymbol{k_i}$ of the i-th memory unit. The two value cells $\langle \boldsymbol{v_i^s}, \boldsymbol{v_i^{s'}} \rangle$ are the corresponding style representations associated with two candidate styles. Specifically, we naturally classify $\boldsymbol{z_s}$ of pairs in each cluster by style labels, and then infill two value cells with the average style vectors with respect to different styles. In this way, we mark the K clusters as the essential content subdivisions and consider samples in the same cluster to describe highly related objects and share similar latent style spaces. Accordingly, the key embedding $\boldsymbol{k_i}$ projects common linguistics of the specific scope of content and value embeddings $\langle \boldsymbol{v_i^s}, \boldsymbol{v_i^{s'}} \rangle$ retain the characteristics of candidate styles w.r.t. $\boldsymbol{k_i}$.

3.3.2 *Memory Reading*

In our model, the memory read operation utilizes the content vector $\boldsymbol{z_c}$ of the input sequence as the retrieving head to fetch the most relevant memory slot by measuring the similarity between $\boldsymbol{z_c}$ and key embedding $\boldsymbol{k_i}$ of every memory slot. We design two strategies, **Hard Read** and **Soft Read**, to retrieve and construct the target style representation. Both strategies use a dot-product based attention mechanism to loop over K memory units, and the former fetch the most relevant memory slot (Eq. (4)) while the latter adopts a weighted summation over all memory slots (Eq. (5)).

$$\text{Read}_{\text{Hard}}(\boldsymbol{z_c}) = \{\boldsymbol{v_{i^*}^{s'}} \mid i^* = \mathop{\arg\max}_{i\in[1,K]}(\boldsymbol{k_i}\cdot\boldsymbol{z_c})\}, \tag{4}$$

$$\begin{aligned} \text{Read}_{\text{Soft}}(\boldsymbol{z_c}) &= \sum_{i=1}^{K}\alpha_i \boldsymbol{v_i^{s'}}, \\ \alpha_i &= \text{softmax}(\boldsymbol{k_i}\cdot\boldsymbol{z_c}), \end{aligned} \tag{5}$$

where $\boldsymbol{v_i^{s'}}$ is the representation for target style s' memoried in the unit $\boldsymbol{m_i} = \langle \boldsymbol{k_i}, \boldsymbol{v_i^s}, \boldsymbol{v_i^{s'}}\rangle$. We fetch the desired style representation, i.e. $\boldsymbol{z_{s^*}}$ in Section 3.2.2, from memory and take it as the input of transfer module.

3.3.3 *Iterative Update*

Since the memory update requires an amount of time steps accumulation for $\langle \boldsymbol{z_c}, \boldsymbol{z_s}\rangle$ pairs, it can not synchronize with the generative model training. Therefore, we divide the entire training into two stages, memory update and generative model training, and then train the two stages alternately.

In the generative model training stage, we fix the information stored in the memory and use it as a standard to participate in the transfer phase. The generative model fetches desired style signals from memory for style controlling of output, thereby optimizing itself. In the memory update stage, the optimized generative model by an epoch training creates a more precise $\langle \boldsymbol{z_c}, \boldsymbol{z_s}\rangle$ pairs of the training corpus, which contributes to improving clustering quality. We conduct clustering on the new collection and write the results into the memory for an update.

The two stages interact to achieve overall optimization iteratively. During training, the latent representation learning module trains to optimize itself, and generate a collection of more accurate and expressive $\langle \boldsymbol{z_c}, \boldsymbol{z_s}\rangle$ pairs. Therefore, our memory is also updated iteratively to adapt to new $\langle \boldsymbol{z_c}, \boldsymbol{z_s}\rangle$ pairs regularly. In practice, the two stages interchange once per epoch, which means we update the memory when the generative model finishes each epoch training.

Intuitively, the performance of our memory is highly correlated with the quality of clustering. For this sake, we also pay attention to the potential error deviation caused by outliers or the skew distribution of samples in the clustering results, especially the few-shot cluster. To address such issues, we further introduce a way to calibrate style signals stored in value cells after updating the memory.

3.4 Memory Calibration

To alleviate the error deviation of latent style space caused by uneven qualities of clusters, we take the insights from TransE [3] and propose a memory calibration method. The motivation is based on the assumption that the relationship between styles follows a similar distribution across all memory units. Therefore, our method first learns a standard relationship representation and uses it as a shared criterion to calibrate value embeddings of memory.

Concretely, for two value cells $\langle \boldsymbol{v_i^s}, \boldsymbol{v_i^{s'}}\rangle$, the relationship is defined as a translation of value embeddings, given by:

$$\boldsymbol{r_i} = \boldsymbol{v_i^s} - \boldsymbol{v_i^{s'}}, \tag{6}$$

where $\boldsymbol{r_i}$ indicates the relationship between $\boldsymbol{v_i^s}$ and $\boldsymbol{v_i^{s'}}$.

Then, we construct a standard relation representation by taking the weighted average of all relationship vectors of K memory units by:

$$\boldsymbol{R} = \sum_{i=1}^{K}\alpha_i \boldsymbol{r_i}, \tag{7}$$

where α_i is the weight of $\boldsymbol{r_i}$, which is determined by the quality of style distribution in the i-th cluster.

We consider a cluster with good quality if groups of different styles in the cluster are well apart from each other and clearly distinguished. That is, we expect a sample with the style s is close to samples sharing the same style while keeping away from other samples with different style s' as much as possible. The Silhouette Coefficient algorithm [29] that measures the separation between clusters is adopted to evaluate whether different groups of style distributions are well separated, where a higher score means a better cluster quality. We calculate the Silhouette Coefficient scores over all clusters as the distribution weights. Specifically,

$$\alpha_i = \frac{\exp(e_i)}{\sum_{i=0}^{K} \exp(e_i)}, \tag{8}$$

where e_i is the Silhouette Coefficient score for cluster i.

Finally, we calibrate value embeddings over the memory module by decreasing the deviation between the relation vector of each cluster and the standard relation vector $\boldsymbol{R}$.

$$\begin{aligned} \boldsymbol{r_i^{bias}} &= \boldsymbol{R} - \boldsymbol{r_i}, \\ \hat{\boldsymbol{v_i^s}} &= \boldsymbol{v_i^s} + \lambda \boldsymbol{r_i^{bias}}, \\ \hat{\boldsymbol{v_i^{s'}}} &= \boldsymbol{v_i^{s'}} - \lambda \boldsymbol{r_i^{bias}}, \end{aligned}$$

where $\boldsymbol{r_i^{bias}}$ indicates the relationship bias on the embedding space and λ is a hyper-parameters to control modification strength. $\langle \hat{\boldsymbol{v_i^s}}, \hat{\boldsymbol{v_i^{s'}}} \rangle$ denote optimized latent style representations in the value cells.

3.5 Unsupervised Training

To effectively support training on non-parallel corpora, we define the following losses to provide supervision indirectly.

3.5.1 *Reconstruction Loss*

Suppose the transfer case that source style is the same as desired style, i.e. $s' = s$, our model is expected to reconstruct the original sentence:

$$\mathcal{L}_{self} = -\mathbb{E}_{(\mathbf{x},s)\sim\mathcal{X}}[\log p_\theta(\mathbf{x} \mid \mathbf{x}, s)]. \tag{9}$$

3.5.2 *Cycle Reconstruction Loss*

For the case $s' \neq s$, we first generate the transferred sentence $\mathbf{x}'$, and by turn switch the transfer direction to rephrase $\mathbf{x}'$ back to $\mathbf{x}$:

$$\mathcal{L}_{cycle} = -\mathbb{E}_{(\mathbf{x},s)\sim\mathcal{X}}[\log p_\theta(\mathbf{x} \mid \mathbf{x}', s)]. \tag{10}$$

3.5.3 *Style Transfer Loss*

Since the ground truth $\boldsymbol{x'}$ is unavailable, we apply an adversarial discriminator module that distinguishes the style of text to provide style supervision. Particularly, the discriminator is trained to identify the fake transferred sentence from the generator. In contrast, the training objective of the generator is to fool the discriminator, i.e. maximize the probability of s' when given the transferred sentence $\boldsymbol{x'}$:

$$\mathcal{L}_{style} = -\mathbb{E}_{(\mathbf{x},s)\sim\mathcal{X}}[\log p_C(s' \mid \mathbf{x}')]. \tag{11}$$

The sampling process or greedy decoding during generating tokens makes the gradients not propagate, thus we follow Dai et al. [5] to adopt the continuous decoding algorithm for token generation. Specifically, $\mathbf{x}'$ is generated by the weighted sum embedding of softmax distribution on the embedding matrix rather than selecting the maximum probability token each time step.

3.5.4 *Cluster Loss*

Given a pair $\langle \boldsymbol{z_c}, \boldsymbol{z_s} \rangle$, we use $\boldsymbol{z_c}$ as a query to fetch from memory a set of style representations, including the one associated with original style, denoted as $\boldsymbol{z_{\hat{s}}}$. We expect $\boldsymbol{z_s}$ to close to $\boldsymbol{z_{\hat{s}}}$ in the latent space, because they stand for the similar content-dependent style signal.

$$\mathcal{L}_{clus} = \mathbb{E}_{(\mathbf{x},s)\sim\mathcal{X}} \| \boldsymbol{z_s} - \boldsymbol{z_{\hat{s}}} \|_2^2 \tag{12}$$

Table 1 Dataset Statistics.

Dataset	Yelp		Amazon		GYAFC	
	Positive	Negative	Positive	Negative	Formal	Informal
Train	270k	180k	277k	278k	52k	52k
Dev.	2000	2000	985	1015	2247	2788
Test	500	500	500	500	1019	1332
Ref.	500	500	500	500	1019	1332
Avg.Len.	8.9		14.9		12.7	

3.5.5 *Overall Loss*

Recall the multi-task losses $\mathcal{L}_{z_s}$ (Eq. 1) and $\mathcal{L}_{z_c}$ (Eq. 2) in Section 3.2.1, our overall loss function is a synthesis of six parts:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{z_s} + \lambda_2 \mathcal{L}_{z_c} + \lambda_3 \mathcal{L}_{self} + \lambda_4 \mathcal{L}_{cycle} + \lambda_5 \mathcal{L}_{style} + \lambda_6 \mathcal{L}_{clus}, \tag{13}$$

where λ_i is the balancing parameter.

4 Experiment

4.1 Datasets

In this paper, we evaluate our method on two typical text style transfer task, sentiment modification [5, 21, 24, 39, 41] on the **Yelp** and **Amazon** corpora, and formality transfer [23, 36] on the **GYAFC** corpus. For fair comparisons, we adopt the common preprocessing methods proposed by Dai et al. [5], and Xiao et al. [36] for data construction. Table 1 illustrates the detailed dataset statistics. The first two datasets consist of positive and negative reviews for Yelp restaurants and Amazon products, respectively. The GYAFC dataset, i.e. Grammarly's Yahoo Answers Formality Corpus provided by Rao et al. [28], contains formal and informal sentences in two domains, namely Entertainment & Music and Family & Relationships. Following previous work [23, 36], we choose the latter and use it in an unpaired setting during training. Besides, All three datasets also release human-written references of the test set for direct evaluation.

4.2 Baselines

We compare two versions of our proposed memory-enhanced transfer model (METM), i.e. **METM-S** and **METM-H** stand for soft reading and hard reading strategies, respectively, with several competitive baselines covering disentanglement-based and attention-based paradigms.

Specifically, we choose the following disentanglement-based competitor models that first learn style-independent content representation and fuse it and desired style control variables for transfer.

- **CrossAlign** [30]: CrossAlign learns style-independent content representation, and fuses it and desired style embedding to generate transferred sentences.
- **MultiDec** [9]: MultiDec adopts adversarial network to guarantee disentanglement and then feeds content representation to multiple style-specific generators without additional style control variables.
- **DelRetGen** [21]: DelRetGen extracts a sentence template by removing stylistic phrases and applies a generative model to fuse it with retrieved phrases carrying the desired attribute for transfer.
- **SMAE** [41]: SMAE deletes sentimental words via a self-attention based classifier and learns sentiment memories to adapt different contexts for sentiment modification.
- **Revision** [22]: Revision revises the original sentences in a continuous space by gradient optimization in the inference stage to achieve transfer.

We consider the following representative methods for the attention-based paradigm, which adopt the encoder-decoder structure with an attention mechanism to obtain entangled representation without disentanglement.

- **DualRL** [24]: DualRL is a dual reinforcement learning algorithm that treats style transfer as a dual task, without any separation of content and style.

- **StyTrans** [5]: StyTrans employs Transformer architecture to obtain entangled representation without disentanglement. It injects stylistic property as an extra embedding added to the entangled embedding.
- **IMaT** [16]: IMaT constructs a pseudo-parallel corpus by iteratively matching and refining semantically similar sentences, and applies a seq2seq based transfer model to learn attribute transfer.
- **PFST** [12]: PFST introduces a probabilistic generative paradigm and achieves transfer by modeling a transduction distribution on latent space.
- **StyIns** [39]: StyIns adopts the attention-based structure and learns latent style space from multiple instances. We consider this model as the strongest baseline.
- **DIRR** [23]: DIRR proposes a reward-based training algorithm and utilizes a semantic similarity metric to enhance content preservation.
- **TSST** [36]: TSST extracts stylistic properties from retrieval-based top-k relevant instances to provide strong style signal.

4.3 Automatic Evaluation

In view of the main characteristics of the transfer task, we follow the common practice to consider three aspects, i.e. style transfer strength, content preservation, and language fluency, in the automatic evaluation metrics [24,36,39]. In addition, we further pay attention to the lexical diversity of the output to judge whether the candidate systems can generate varied style-related phrases or just prefer similar generic phrases independent of the context.

- **Style Transfer Strength** To measure style transfer strength quantitatively, we calculate style accuracy (**Acc.**) of transferred sentences via a fine-tuned BERT-based [6] classifier. The accuracies of well-trained classifiers reach 98% on Yelp, 89% on Amazon, and 90% on GYAFC, respectively.
- **Content Preservation** Following the standard practice, we report the self-BLEU (**s-BLEU**) metric between the input and the output of transfer systems, and thereby a higher score indicates better content preservation to some extent. Besides, we also calculate the ref-BLEU (**r-BLEU**) score for direct evaluation by comparing the output with human-written references. To ensure the fairness of comparison, we adopt the NLTK BLEU scoring function [2] for all BLEU calculations as similar to Yi et al. [39] and Xiao et al. [36].
- **Language Fluency** The measurement of fluency is the perplexity (**PPL**) of generated sentences. Following previous work [5, 36, 39], we adopt KenLM [13] to train 5-gram language models on three datasets for the calculation of perplexity.
- **Diversity** We report the Distinct-1 (**Dist-1**) and Distinct-2 (**Dist-2**) scores [43] that calculate the proportion of distinct unigrams / bigrams in the transferred results to indicate the diversity.
- **GM** Intuitively, there is a trade-off between modifying the style and preserving the content [15,39]. For this sake, we further apply a geometric mean of transfer accuracy, self-BLEU, ref-BLEU, and $\frac{1}{\log \mathrm{PPL}}$ for overall quality evaluation, denoted as **GM** [36]. Note that we omit Distinct-1 and Distinct-2 for the calculation of GM, which remains the same as previous studies for a fair comparison.

4.4 Human Evaluation

Due to the time and economical consumption of human annotation, we choose the four most competitive methods with the highest GM metric and our METM models as candidates. In practice, we randomly sample 100 transfer cases (50 cases for each style) from the output of each transfer system (1800 cases in total on all three datasets). Each case contains the original sentence, target style attribute, and generated sentence. We then distribute those samples to three annotators for giving a score range from 1 (the worst) to 5 (the best) in terms of three common criteria: style transfer accuracy (**Style**), content preservation (**Content**), and **Fluency**. Similar to Distinct-n metrics for lexical diversity in automatic evaluation, these annotators also need to evaluate the overall linguistic diversity (**Diversity**) of stylistic expressions from different systems. Note that the diversity metric focuses on the overall lexical diversity, thus not a property for a single case. In practice, we divide the sampled cases into groups of 10, and then ask the annotator to calculate the number of unique style expressions (i.e. range from 1 to 10) in each group. The evaluation follows a strictly random and blind fashion to avoid human bias. Considering the workload and difficulty, the expected annotation time is 18 hours (100 sentences per hour). Therefore, each annotator is compensated with 1800 Chinese Yuan (CNY), and the hourly pay is CNY1800, which exceeds the Chinese statutory minimum wage.

Table 2 Automatic evaluation results on the sentiment modification task. Bold cells indicate the best performances. We conduct hypothesis testing and verify all the results in which our model performs best are statistically significant with $p < 0.05$ under t-test.

Type	Method	Yelp							Amazon						
		Acc.↑	r-BLEU↑	s-BLEU↑	PPL↓	GM↑	Dist-1↑	Dist-2↑	Acc.↑	r-BLEU↑	s-BLEU↑	PPL↓	GM↑	Dist-1↑	Dist-2↑
Disen.	CrossAlign	78.7	8.11	16.65	66	7.10	0.088	0.531	69.6	2.02	2.84	95	3.06	0.119	0.475
	MultiDec	45.4	15.07	40.07	188	8.51	0.111	0.701	66.5	6.99	16.34	88	6.42	0.065	0.513
	DelRetGen	88.1	16.66	36.75	100	10.40	0.154	0.617	52.4	29.14	53.31	85	11.63	0.151	0.583
	SMAE	76.6	15.24	43.05	65	10.47	0.172	0.683	70.4	15.44	45.43	92	10.22	0.164	0.627
	Revision	90.6	7.93	13.23	21	7.47	0.111	0.432	**80.7**	13.99	20.07	**38**	8.88	0.127	0.497
Attn.	DualRL	87.9	28.77	58.90	105	13.38	0.142	0.643	62.4	24.18	49.17	124	11.14	0.152	0.675
	StyTrans	86.0	27.32	59.46	154	12.91	0.168	0.691	65.5	23.58	61.92	208	11.57	0.142	0.636
	IMaT	**93.9**	11.26	16.92	**14**	9.07	0.120	0.480	72.4	12.65	33.95	58	9.35	0.140	0.633
	PFST	84.6	23.72	48.90	67	12.36	0.153	0.628	61.8	34.81	66.92	87	13.40	0.170	0.677
	StyIns	90.9	26.09	53.10	110	12.79	0.147	0.664	62.3	22.42	43.39	115	10.63	0.128	0.425
	DIRR	91.7	28.54	58.98	144	13.28	0.158	0.636	58.8	32.80	66.38	110	12.85	0.173	0.662
	TSST	91.8	28.89	59.34	108	13.54	0.141	0.655	59.4	41.59	69.95	130	13.73	0.171	0.694
Ours	METM-S	92.3	28.62	61.42	112	13.62	0.164	0.667	66.0	40.51	**70.02**	126	14.03	0.181	0.696
	METM-H	91.1	**29.34**	**63.49**	113	**13.76**	**0.174**	**0.701**	62.2	**43.68**	68.77	121	**14.05**	**0.189**	**0.707**

4.5 Implementation Details

In this paper, we employ a one-layer bi-directional GRU for the style and content encoders, and a one-layer unidirectional GRU with the attention mechanism for the decoder. We utilize 300-dim pre-trained GloVe [26] as word embeddings shared by both encoders and decoder. Correspondingly, the hidden state size is also set to 300 for both encoders and decoder. Due to the powerful ability of the Transformer, the discriminator adopts a 4-layer Transformer with 4-way multi-head attention. In the training stage, we set the memory size to 50 and employ the Adam optimizer [18] with the initial learning rate of 0.0001 and gradient clipping of 5.

4.6 Result and Analysis

4.6.1 *Automatic Evaluation Results*

Table 2 illustrates the automatic evaluation results on the sentiment modification task. Generally, disentanglement-based methods achieve comparable or even better transfer accuracy and PPL, but perform much worse in content preservation, i.e. ref-BLEU and self-BLEU scores. For instance, the Revision model achieves promising or near-best performances on transfer accuracy and language fluency, especially on Amazon reaching 80.7 and 38 in terms of accuracy and PPL metrics, respectively, but suffers from a substantial performance decrease on content preservation across all datasets. These results illustrate that disentanglement-based methods tend to generate fluent sentences with target stylistic properties but are irrelevant to original content semantics. We attribute this to the intractability of disentanglement, which causes damage to the content information when separating the style from the text.

In contrast, attention-based methods avoid explicit disentanglement and thus better preserve content information. Almost all attention-based models except IMaT marginally outperform disentanglement-based methods in terms of ref-BLEU and self-BLEU metrics. For example, TSST gains a remarkable improvement of nearly 10 points on all BLEU scores compared to well-performed disentanglement-based models. The exception of IMaT is likely because the method of constructing pseudo-parallel corpus requires the assumption of disentanglement implicitly, thereby conforming with a similar pattern as the disentanglement-based paradigm. Despite enhanced content retainment, attention-based methods performed slightly worse in transfer success ratio, especially a non-trivial drop on Amazon. Representative methods include DualRL, StyTrans and PFST consistently underperform in terms of transfer accuracy compared to state-of-the-arts obtained by disentanglement-based line, which verifies that taking simple style embedding is not enough for guiding transfer.

As for the formality transfer task on the GYAFC dataset, Table 3 reports the experimental results of our method and competing methods. Similarly, attention-based methods generally surpass disentanglement-based methods in terms of content preservation. Nevertheless, the comparison on transfer strength (Acc.) shows a slight difference from the previous observation. In the previous sentiment modification task, the state-of-the-art disentanglement-based methods achieve comparable or even superior accuracy compared to the attention-based line. However, as migrate to formality transfer task, disentanglement-based suffer from a dramatic decrease in transfer accuracy and are generally inferior to the attention-based line. Taking Revision as an example, it achieves promising performance in terms of accuracy on the Yelp and Amazon datasets, reaching 90.6 and 80.7, respectively, but sharply drops to only 39.6 on the GYAFC dataset. We

Table 3 Automatic evaluation results on the formality transfer task.

Type	Method	GYAFC						
		Acc.↑	r-BLEU↑	s-BLEU↑	PPL↓	GM↑	Dist-1↑	Dist-2↑
Disen.	CrossAlign	61.6	3.25	2.21	**37**	3.33	0.012	0.081
	MultiDec	24.5	11.95	16.08	151	5.53	0.021	0.320
	DelRetGen	58.2	21.88	31.57	103	9.65	0.020	0.246
	SMAE	59.5	23.46	43.15	87	10.78	0.066	0.451
	Revision	39.6	20.83	27.64	66	8.59	0.025	0.096
Attn.	DualRL	55.5	43.69	52.80	159	12.61	0.063	0.473
	StyTrans	60.3	43.95	61.15	168	13.34	0.067	0.469
	IMaT	63.8	4.18	53.77	46	7.82	0.072	0.434
	PFST	63.9	21.53	28.68	40	10.17	0.066	0.342
	StyIns	69.9	47.80	61.87	140	14.30	0.066	0.473
	DIRR	71.8	46.37	59.97	145	14.15	0.061	0.417
	TSST	74.1	50.49	63.70	103	15.06	0.061	0.468
Ours	METM-S	**75.6**	49.74	61.64	87	15.09	0.069	**0.478**
	METM-H	74.0	**52.05**	**64.08**	99	**15.22**	**0.072**	0.477

Table 4 Human evaluation results. The best performances are shown in bold. The Krippen-dorff's alpha of human rating is 0.7727, indicating moderate inter-annotator agreement.

Method	Yelp				Amazon				GYAFC			
	Style	Content	Fluency	Diversity	Style	Content	Fluency	Diversity	Style	Content	Fluency	Diversity
DualRL	3.40	3.75	**4.18**	5.13	3.09	2.96	3.47	6.87	1.97	3.07	3.20	5.17
StyIns	3.47	3.74	4.14	6.67	2.67	2.73	3.82	6.20	2.26	2.98	2.31	5.83
DIRR	3.69	3.87	3.79	5.37	2.26	3.32	**4.04**	7.07	2.78	3.41	3.51	4.86
TSST	3.81	4.05	4.07	6.17	2.43	3.74	3.79	7.83	2.73	3.47	3.77	5.37
METM-S	**3.82**	4.09	4.13	7.50	3.02	3.37	3.85	8.10	**2.96**	3.27	**4.05**	5.83
METM-H	3.78	**4.18**	4.02	**8.07**	**3.14**	**3.75**	3.98	**8.23**	2.78	**3.55**	3.84	**6.07**

deduce the substantial difference because the disentanglement of style attributes and content information is more challenging in the formality transfer task than sentiment modification. Such a phenomenon indicates that the assumption of disentanglement may limit the scope of application in the real world.

Besides, the diverse metrics, i.e. Distinct-1 and Distinct-2, vary substantially between different approaches. Despite the acceptable transfer accuracy, these baselines that adopt simple style embedding as the controllable signal, e.g., CrossAlign, Revision, and IMaT, consistently perform unsatisfactorily in lexical diversity. The results demonstrate that these models only generate universal style-related phrases independent of the input, and prove our claim that solo embedding is not sufficient to represent linguistic style. In comparison, representing style in a fine-grained form indeed provides noticeable gains in diversity metrics. For instance, TSST that enhances style signals via incorporating retrieval samples achieves considerable improvement across all three datasets, indicating the advantage of fine-grained style representation in expressing diverse stylistic phrases.

Our model consistently outperforms benchmark models on most metrics, especially encouraging improvements in terms of ref-BLEU score, which indicates that the output of our model has more similar context semantics and style attributes to manual references. Intuitively, there is a trade-off between the transfer accuracy and content preservation criteria [39,40]. As seen, our proposed METM model achieves a better balance of transfer accuracy and content preservation. The highest GM metrics on all three datasets show the superiority of our model on not only altering style but also retaining content. Crucially, both versions of our models achieve remarkable improvements in lexical diversity, verifying that fine-grained content-dependent style signals confer a significant advantage in expressing diverse stylistic attributes. We also find that our method achieves the most remarkable overall performance improvement on the Amazon dataset. To explain this phenomenon, we count the number of unique nouns in each dataset (Yelp: 7750, Amazon: 40505, GYAFC: 13285), showing that product reviews in Amazon involve the broadest range of objects. This supports that our memory module takes effects in the stylistic generation, especially when adapting to various topics of content. For two versions of our models, METM-H slightly surpasses METM-S on most quantitative metrics, which indicates that hard reading strategy is a little better than soft reading for fetching desired style representation.

Table 5 Model ablation study result on Yelp dataset.

Model	Acc.	ref-BLEU	self-BLEU	PPL	GM
METM-H	91.1	29.34	63.49	113	13.76
(-)Memory	91.0	26.88	55.63	115	13.01
(-)Calibration	91.3	28.41	60.72	108	13.54
(-)$\mathcal{L}_{z_s}$	92.2	27.57	59.69	111	13.40
(-)$\mathcal{L}_{z_c}$	90.5	28.55	63.80	116	13.65

Table 6 The performances with different memory sizes K.

METM-H	Acc.	ref-BLEU	self-BLEU	PPL	GM
$K = 1$	91.3	26.42	54.48	109	12.94
$K = 10$	89.9	28.35	61.29	112	13.49
$K = 50$	91.1	**29.34**	**63.49**	113	**13.76**
$K = 100$	**91.9**	28.85	60.91	**109**	13.62
$K = 500$	91.1	28.60	61.30	113	13.56

4.6.2 *Human Evaluation Results*

We report human evaluation results in Table 4, which are highly consistent with automatic evaluation results in terms of transfer accuracy, content preservation, and diversity. Our proposed METM model achieves comparable fluency compared to the most well-performed models under manual evaluation, proving the capacity of generating fluent transferred sentences with desired style properties. The result indicates that a moderate PPL metric (not necessarily overly low PPL) meets fluency requirement, which is accordant with previous studies [12, 39]. Also notably, our model marginally outperforms all four competitor models in terms of diversity scores, verifying the superiority of our model in generating diverse stylistic phrases that vary with the context. In light that stylistic expressions vary substantially under different contexts, our proposed content-style memory module provides fine-grained content-dependent style signals, rather than taking a simple embedding to control transfer. The more expressive and flexible style representations make the model avoid generating tired content-irrelevant stylistic phrases and confers a significant advantage in improving lexical diversity.

By modeling an expressive latent style space via transductive learning, TSST achieves competitive performance in all four indicators. Even so, our model performs better than TSST under most metrics. We attribute the improvement to the superiority of style representation construction. Unlike only fetching several samples to distill desired style representation, our method integrates all training samples via clustering to learn more expressive and flexible latent style space. In addition, the calibration operation on stylistic properties also potentially provides further performance gains.

4.6.3 *Ablation Study*

To study the effects of critical components of our method, we conduct an ablation study of our method, which is present in Table 5 (only the Yelp dataset due to limited space). We first investigate the impact of the memory module by presenting two variants. The first one (denoted as (-)Memory) replaces the memory module with typical style embeddings. The variant consistently underperforms our model across all aspects, especially suffers from extremely substantial performance decrease on content preservation, i.e. ref-BLEU and self-BLEU scores. The result illustrates that only simple style embedding as guidance signals is likely to make the transferred sentence not irrelevant to the original content to some extent. Instead, fine-grained content-dependent style expressions in our memories enable the model to generate style-related phrases consistent with the original context, thus contributing to retaining content information. Another one (denoted as (-)Calibration) only removes the memory calibration part, by contrast, showing that calibration indeed further boosts overall performances. We conjecture this is because the style representations on cells with few samples and cells with many outliers bring in noise, while calibration works on those two types of cells and potentially provides further gains in performance. For better understanding the role of two auxiliary losses, i.e. $\mathcal{L}_{z_s}$ and $\mathcal{L}_{z_c}$ in the latent representation learning module, we further disable $\mathcal{L}_{z_s}$ and $\mathcal{L}_{z_c}$ by turns. The removal of $\mathcal{L}_{z_s}$ leads to a drop in transfer accuracy, which means the auxiliary classification task on latent style space help to obtain expressive and discriminative style representation. In contrast, $\mathcal{L}_{z_c}$ mainly contributes to enhancing content preservation.

4.6.4 *Memory Analysis*

To investigate the effect of different memory capacities, we conduct comparison experiments on METM-H with a range of memory sizes K and report the results in Table 6. Our memory with the setting of $K = 1$ is similar to typical individual style embeddings, correspondingly, the performances are comparable to the variant without the memory module. A small memory ($K \leqslant 10$) is not enough to memorize various linguistic characteristics, bringing only slight improvements. Generally, the increase of K promotes learning more expressive style representation, thereby achieving better performance. However, overly

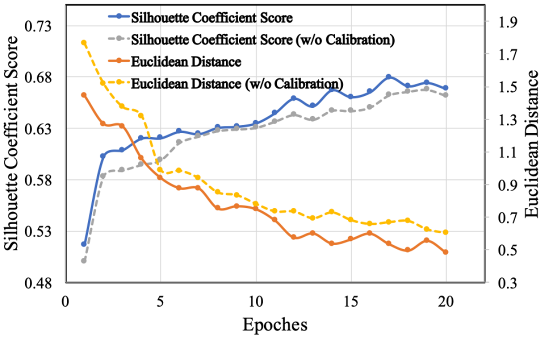
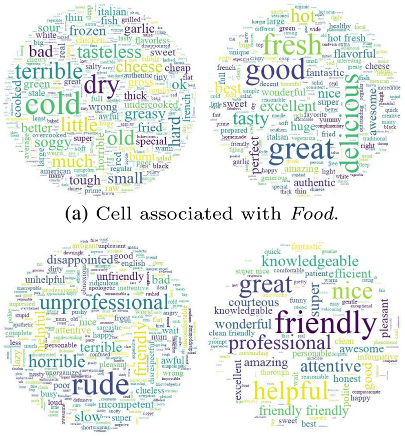


Figure 3 The variation of clustering attributes during training. A higher Silhouette Coefficient score indicates more discriminative latent style space. A lower Euclidean distance means a higher similarity.



(a) Cell associated with *Food*.

(b) Cell associated with *Person*.

Figure 4 Visualization of samples from different cells.

high memory sizes ($K \geqslant 100$) perform worse due to the raised error derivation in few-shot clusters. Besides, the required resources and training time increase significantly with excessive K. In conclusion, moderate K is appropriate, and we set $K = 50$ by comparison.

Figure 3 examines how clustering attributes (i.e. average silhouette coefficient scores and Euclidean distance between relationship vectors $\{\boldsymbol{r_1}, \boldsymbol{r_2}, \cdots, \boldsymbol{r_{|K|}}\}$) change during training. The increasing silhouette coefficient scores show that groups with different styles are becoming more and more discriminative, while the decrease of average distance validates that the relationship between different styles has a good similarity across various content topics. Crucially, the calculation operation helps to learn more discriminative and expressive latent style space.

Further, We pick up the samples divided into the same cluster (memory cell) and visualize their stylistic attributes via word cloud (two cells in Figure 4). We find that each memory cell shares a closely related topic and learns varied stylistic phrases with respect to specific content. Specifically, the top cell is related to the food topic and supplies corresponding expressions, while the bottom cell mainly focuses on properties associated with the person topic. In conclusion, the memory cells cluster similar cases together and leverage fine-grained style signals for controlling transfer.

4.6.5 *Case Study*

To intuitively compare the characteristics of different models, Table 7 illustrates sampled output sentences from our model and four of the most competitive baselines. In the sentiment modification case, almost all baseline can alter the style of the input sentences to expected stylistic attributes. Even so, several baselines simply add the adverb *not* to achieve sentimental transformation, e.g., change *great* to *not great* in the sentimental positive to negative transfer direction, resulting in a relatively monotonic pattern in the transferred sentences. Moreover, we can observe that the baselines tend to dismiss part of original semantics more or less, especially when there are more than two described objects, e.g., *their tone* and *customer service*, in the input sentences. As a comparison, our model can achieve successful transfer in generating more vivid sentimental phrases and better retaining original content. As for the formality transfer case, several baselines do not conduct effective rephrase, and similarly, also fail to remain the content information unchanged. In contrast, our model confers a significant advantage in preserving content under the premise of successfully altering formality. For example, as an alternative to the original phrase *when ur ready*, the generated phrase *when you are ready* correctly improves lexical formality and better preserves semantics compared to *when ready*.

5 Conclusion

In this paper, we propose a memory-enhanced method with dynamic style learning for text style transfer. Instead of representing style with individual embedding, we construct a content-style memory to learn a more expressive and flexible latent style space. Our model clusters the training corpora, extracts and saves common linguistic features for style transfer. Besides, we introduce a way to further calibrate memory via

Table 7 Case study on sentiment modification and formality transfer tasks. Phrases in bold mean successful transfer in style.

Sentiment Modification		
Model	**positive to negative**	**negative to positive**
Input	i love this place, the service is always great!	always rude in their tone and always have shitty customer service!
DualRL	i **hate** this place, the service is **not** great!	always friendly in their best price and always have loved customer service!
StyIns	i do not know, the service is not even filtered cove.	always **nice** in their tone and always have **wonderful** customer service!
DIRR	**hate** this place, service was bad.	such **nice** customer service, they listen to anyones concerns and assist them with it.
TSST	i **hate** this place, the service is **not** great!	always friendly in their service and always have **great** customer service!
Ours	i **hate** this place, the service is always **terrible**!	always **friendly** in their tone and always have **great** customer service!

Formality Transfer		
Model	**informal to formal**	**formal to informal**
Input	it all depends on when ur ready. stay 100 miles away from this guy.	the two of you should kiss when you are ready. yes i am a male, therefore i do not really have to pay.
DualRL	it all depends on when ready. stay miles away from 100 guy.	the two of you should kiss when you are ready!!!! yes i am a male er.
StyIns	it all depends on when your ready. stay an miles away from this **man**.	**2** of you should kiss when you are ready. yes i am a male girl lol i do **n’t** really have to pay.
DIRR	it all depends on when **you are** ready. stay away from this guy.	the **2** of you dont kiss when you are ready. yes i **’m** a girl freind therefore i do **n’t** really have to pay.
TSST	it all depends on when your ready. stay 100 miles away from this **man**.	and **2** of you should kiss when you are ready. yes i **’m** a kid, therefore i do **n’t** really have to pay.
Ours	it all depends on when **you are** ready. stay 100 miles away from this **man**.	the **2** of you kiss when **u are** ready. yes i **’m** a **girl**, **yeah** i do **n’t** really have to pay.

projecting the relationship between styles. Thereby, our model can provide dynamic style signals with respect to content, which contributes to generating diverse and informative sentences. Experimental results on three datasets verify the superiority of our model on style control and content preservation.

6 Ethical Considerations

Our work focuses on text style transfer that controls stylistic properties of generated text while retaining content semantics. Such methods have a broad impact in the field of controllable natural language generation [11] and can also provide strong support for potential real-world applications, e.g., stylized response generation [1], stylistic summarization [8], text simplification [4], and offensive language transfer [7, 32]. Nonetheless, as with all text style transfer methods, our method is also potentially used maliciously with concealed intentions, including possible content manipulation and forgery issues, e.g., fake review generation. For this sake, we restrict the proposed method to academic use only, and it has to be coupled with strict misrepresentation, offensiveness, and bias checks. Further, with increasing attention on shared ethical issues in text generation models, we encourage future studies to tackle such cases.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 62106275).

References

1 Bai G, He S, Liu K, et al. Example-guided stylized response generation in zero-shot setting. Sci China Inf Sci, 2022, 65(4): 1-2

2 Loper E, Bird S. NLTK: The natural language toolkit. In: Proceedings of International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, 2002. 63-70

3 Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, Nevada, 2013. 2787-2795

4 Cao Y, Shui R, Pan L, et al. Expertise style transfer: A new task towards better communication between experts and laymen. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 2020. 1061-1071

5 Dai N, Liang J, Qiu X, et al. Style transformer: Unpaired text style transfer without disentangled latent representation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, 2019. 5997-6007

6 Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, 2019. 4171-4186

7 dos Santos C, Melnyk I, Padhi I. Fighting offensive language on social media with unsupervised text style transfer. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, 2018. 189-194

8 Fan A, Grangier D, Auli M. Controllable abstractive summarization. In: Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, Melbourne, 2018. 45-54

9 Fu Z, Tan X, Peng N, et al. Style transfer in text: Exploration and evaluation. In: Proceedings of the AAAI Conference on Artificial Intelligence, Louisiana, 2018. 663-670

10 Gatys L A, Ecker A S, Bethge M. Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, 2016. 2414-2423

11 Guo Q, Qiu X, Xue X, et al. Syntax-guided text generation via graph neural network. Sci China Inf Sci, 2021, 64(5): 1-10

12 He J, Wang X, Neubig G, et al. A probabilistic formulation of unsupervised text style transfer. In: Proceedings of International Conference on Learning Representations, Addis Ababa, 2020

13 Heafield K. KenLM: Faster and smaller language model queries. In: Proceedings of the sixth workshop on statistical machine translation, Scotland, 2011. 187-197

14 Hu Z, Yang Z, Liang X, et al. Toward controlled generation of text. In: Proceedings of International conference on machine learning, Sydney, 2017. 1587-1596

15 Jin D, Jin Z, Hu Z, et al. Deep learning for text style transfer: A survey. Comput Linguist, 2021: 1-51

16 Jin Z, Jin D, Mueller J, et al. IMaT: Unsupervised text attribute transfer via iterative matching and translation. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, 2019. 3097-3109

17 John V, Mou L, Bahuleyan H, et al. Disentangled representation learning for non-parallel text style transfer. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, 2019. 424-434

18 Kingma D P, Ba J. Adam: A method for stochastic optimization. In: Proceedings of International Conference on Learning Representations, San Diego, 2015

19 Lample G, Subramanian S, Smith E, et al. Multiple-attribute text rewriting. In: Proceedings of International Conference on Learning Representations, New Orleans, 2018

20 Lee J. Stable style transformer: Delete and generate approach with encoder-decoder for text style transfer. In: Proceedings of the 13th International Conference on Natural Language Generation, Dublin, 2020. 195-204

21 Li J, Jia R, He H, et al. Delete, retrieve, generate: A simple approach to sentiment and style transfer. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Louisiana, 2018. 1865-1874

22 Liu D, Fu J, Zhang Y, et al. Revision in continuous space: Unsupervised text style transfer without adversarial learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, New York, 2020. 8376-8383

23 Liu Y, Neubig G, Wieting J. On learning text style transfer with direct rewards. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 2021. 4262-4273

24 Luo F, Li P, Zhou J, et al. A dual reinforcement learning framework for unsupervised text style transfer. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, 2019. 5116-512

25 Malmi E, Severyn A, Rothe S. Unsupervised text style transfer with padded masked language models. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Online, 2020. 8671-8680

26 Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, 2014. 1532-1543

27 Prabhumoye S, Tsvetkov Y, Salakhutdinov R, et al. Style transfer through back-translation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, 2018. 866-876

28 Rao S, Tetreault J. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Louisiana, 2018. 129-140

29 Rousseeuw P J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math, 1987, 20: 53-65

30 Shen T, Lei T, Barzilay R, et al. Style transfer from non-parallel text by cross-alignment. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, 2017. 6833-6844

31 Tian Y, Hu Z, Yu Z. Structured content preservation for unsupervised text style transfer. 2018. ArXiv: 1810.06526

32 Tran M, Zhang Y, Soleymani M. Towards a friendly online community: An unsupervised style transfer framework for profanity redaction. In: Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, 2020. 2107-2114

33 Wang K, Hua H, Wan X. Controllable unsupervised text attribute transfer via editing entangled latent representation. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, 2019. 11036-11046

34 Wang Z, Xu L, Liu Z, et al. Topic-sensitive neural headline generation. Sci China Inf Sci, 2020, 63(8): 1-16

35 Wu C, Socher R, Xiong C. Global-to-local memory pointer networks for task-oriented dialogue. In: Proceedings of the 7th International Conference on Learning Representations, LA, 2019

36 Xiao F, Pang L, Lan Y, et al. Transductive learning for unsupervised text style transfer. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, 2021. 2510-2521

37 Xu J, Sun X, Zeng Q, et al. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, 2018. 979-988

38 Yang P, Li L, Luo F, et al. Enhancing topic-to-essay generation with external commonsense knowledge. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, 2019. 2002-2012

39 Yi X, Liu Z, Li W, et al. Text style transfer via learning style instance supported latent space. In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021. 3801-3807

40 Yu W, Zhu C, Li Z, et al. A survey of knowledge-enhanced text generation. 2020. ArXiv: 2010.04389

41 Zhang Y, Xu J, Yang P, et al. Learning sentiment memories for sentiment modification without parallel data. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, 2018. 1103-1108

42 Zhang Z, Ren S, Liu S, et al. Style transfer as unsupervised machine translation. 2018. ArXiv: 1808.07894

43 Li J, Galley M, Brockett C, et al. A Diversity-Promoting Objective Function for Neural Conversation Models. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, California, 2016. 110-119