A Personalized Cross-Platform Post Style Transfer Method Based on Transformer and Bi-Attention Mechanism

Zhuo Chen[1], Baoxi Liu[1], Peng Zhang[1]*, Tun Lu[1], Hansu Gu[2], Ning Gu[1]
[1]Fudan University, Shanghai, China, [2]Seattle, United States
{chenzhuo19,zhangpeng_,lutun,ninggu}@fudan.edu.cn,liu_baoxi@foxmail.com,hansug@acm.org

ABSTRACT

To meet different social purposes, users usually share content related to the same topic or event to multiple social media platforms (cross-platform content sharing). As the differences of social norms and audiences among these social ecosystems, there are differences in the use of words and expressions in different platforms, resulting in different language styles among different platforms. In reality, it is usually difficult for users to grasp the consistency between the language style of posts to be published and that of a platform as the problem of context collapse. To address this problem, firstly, we conduct an study to investigate users' content sharing practices across two Chinese popular social media platforms (Douban and Weibo). The results indicate that: 1) there are significant linguistic differences between different platforms; 2) users' content sharing practices are personalized, and the style of their newly shared content is correlated with their historical posts. Secondly, based on the above findings, we propose a personalized cross-platform post style transfer model. The model can automatically transfer users' posts from one platform's language style to the target platform's language style, while preserving the content and reflecting users' personalized characteristics as much as possible. Experiments on the datasets collected from Douban and Weibo show that our model generally outperforms other comparison models on both style transfer and personalization metrics.

CCS CONCEPTS

• Social and professional topics → User characteristics; **• Computing methodologies → Artificial intelligence**.

KEYWORDS

Cross-Platform Content Sharing; Text Style Transfer

ACM Reference Format:

Zhuo Chen, Baoxi Liu, Peng Zhang, Tun Lu, Hansu Gu, Ning Gu. 2022. A Personalized Cross-Platform Post Style Transfer Method Based on Transformer and Bi-Attention Mechanism. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3488560.3498528

*Corresponding author.

WSDM '22, February 21–25, 2022, Tempe, AZ, USA

ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498528

1 INTRODUCTION

Nowadays, social media platforms have become important channels for people to share opinions and record daily lives[36]. As different platforms have different functions and experiences, and users themselves have diverse needs, users often post content related to the same topic or event to multiple social media platforms (cross-platform content sharing)[38]. However, different social media platforms usually have different audiences and norms, and thus the user-generated content in different platforms has different language styles[38]. Therefore, to better adapt to the language styles of different platforms, users cannot simply copy and paste when sharing content across social media platforms, but need to adjust the words and expressions[28, 38]. At the same time, users usually have their own personalized language expression habits[17], such as the habit of using specific words, expressions, etc. So different users will organize content in a personalized way.

However, in the social media scenario, both users' needs for self-expression and impression management need to be satisfied, which complicates users' content posting decisions and faces the problem of context collapse[20]. When sharing content on multiple platforms, the scenarios they face are more complex and diverse, making it more difficult to grasp the consistency between the language style of posts to be published and that of a platform.

The development of natural language processing technology provides support for solving the problem faced by content sharing across platforms. Relying on these technologies, we can design and implement a model that automatically transfers user-generated content from the language style of one platform to that of another platform while preserving the original content, and making it as consistent as possible with user's personalized characteristics simultaneously, which will greatly facilitate users to share content on multiple platforms. Similarly, there have been some related works attempting to transfer the text style. According to the different types of datasets, it can be categorized into two types: style transfer with parallel datasets and style transfer with non-parallel datasets[33]. Parallel datasets contain texts expressing the same content in different styles, while non-parallel datasets do not. The models based on parallel datasets[24, 33] are easy to implement, but the construction of parallel datasets mostly relies on manual annotation[2]. To overcome this problem, more studies have explored text style transfer based on non-parallel datasets[11, 15, 27, 30]. For example, Hu et al.[11] attempt to transform texts expressing positive sentiments into texts expressing negative sentiments, and Li et al.[15] explore a simple method to transfer factual sentences into texts with romantic or humorous styles. These methods can achieve text style transfer without relying on parallel datasets, but cannot be directly applied to our task. First, most methods[11, 15, 27] attempt to precisely disentangle style and content, which may lead to the loss or destruction of the original content[29], and the quality of disentanglement is always difficult to judge[3]; second, existing models rely only on textual content but ignore users' personalized characteristics, thus failing to generate posts with personalization. So the existing models still cannot meet the demand for personalized cross-platform post style transfer.

However, it is very challenging to build a personalized cross-platform post style transfer model. The first challenge lies in the dataset. To the best of our knowledge, there is no relevant public cross-platform dataset. The construction of cross-platform dataset first needs to identify and match the same user from two different platforms, which is not easy. Second, cross-platform content sharing is a complex user behavior. The analysis requires extracting and analyzing related factors from both platform and user dimensions. The existence of so many factors and the complex relationship among them bring challenges to the analysis. Third, the model needs to learn the style characteristics of different platforms as well as the personalized characteristics of users. Integrating user's personalized information while achieving style transfer is also tricky.

To address the above questions and challenges, we focus on the user-generated content on two popular Chinese social media platforms—Douban and Weibo. We construct a dataset containing 96,233 Douban posts and 108,594 Weibo posts published by cross-platform users. Then we analyze the linguistic characteristics and user posting behaviors on different platforms using classification model and regression analysis methods. The results show that: 1) there are significant linguistic differences among different platforms; 2) the process of users' posting practice has obvious personalized characteristics, and the style of their newly shared content is correlated with their historical posts. Based on the above findings, we propose a personalized cross-platform post style transfer model. The basic model consists of two main parts: encoder and decoder, which takes Transformer[31] as the basic block. We introduce a classifier to create supervision from the non-parallel data and apply a cycle reconstruction training process[2] to better preserve the original content. Specifically, to generate posts that conform to the user's personalized style, we introduce the user's historical posting information through a Bi-LSTM layer[9]. Furthermore, We apply the bidirectional attention mechanism[26, 32] to capture the correlation between the user's historical posts and new posts and explore the interaction between the two parts. Lastly, in the decoder, we mix the encoded information with a gated attention routing mechanism[37] to generate new posts. To evaluate our model, we conduct experiments on our newly constructed cross-platform dataset, and the experimental results show that the proposed model generally outperforms other comparison models in terms of style transfer and personalization metrics. To conclude, our contributions are summarized as follows:

- We propose and explore a personalized cross-platform post style transfer task. To the best of our knowledge, this is the first work that attempts to transfer text styles between different social media languages.
- We use various methods to study the linguistic differences and users' posting practices on different platforms. We obtain two new insights, which provide valuable information for cross-platform user behavior research.
- Based on the analysis results, we propose a personalized cross-platform post style transfer model. The model introduces the bidirectional attention and the gated attention routing mechanism, which can generate posts that satisfy both platform style and user personalization requirements.
- Experimental results on our newly constructed dataset show that our model generally outperforms other comparison models in terms of style transfer accuracy, content preservation, fluency as well as personalization metrics: UMA and MRR.

2 RELATED WORK

2.1 Content Analysis Across Social Media Platforms

Some works analyze the differences between different platforms. An important finding is that although the mainstream platforms provide roughly similar social functionalities, there are also differences in some features, especially in terms of platform culture and norms[38]. Therefore, the language and behavior of users is different on different platforms. For example, Manikonda et al.[19] analyze the differences in the posts posted by the same group of users on Twitter and Instagram. They find that the posts on Instagram are more positive, while there are more negative expressions on Twitter. Similarly, Lin et al.[16] find that the posts on Facebook is more emotional, while posts on Twitter are more casual. Jaidka et al.[12] reveal that users are more likely to disclose information or express emotion more "honest" on Facebook. Moreover, to study the linguistic differences between the two platforms, they build a binary classifier to distinguish which platform a post comes from. The accuracy reaches 84%, indicating that there are indeed distinguishable linguistic differences between the two platforms.

In addition, some works study the user behavior of cross-platform content sharing. To meet different social purposes, users usually share content related to the same topic or event to multiple platforms[4, 38]. However, the audience engagement on different platforms is different[1], different platforms usually have different audiences and norms, and the user-generated content of different platforms affected by them has different language styles[17, 38]. Therefore, to adapt to the language style of target platforms, it's better for users to adjust the words and expressions when sharing content across platforms. For example, Zhong et al.[38] suggest that users may enable different aspects of their personalities depending on the social context, and most users tend to fit into the conventions of a particular platform by modifying what they show. Sleeper et al.[28] find that users like sharing posts with similar topics to different platforms, but they generally adjust the language of posts considering platform norms and audiences. And by following in-group customs, users can transform from being an idiosyncratic individual to being a group memeber more easily[38].

Although there have been many works studying user behavior on multiple platforms, no work has explored the task of post style transfer, which is a gap our work fills in.

2.2 Text Style Transfer

Recently, many text style transfer methods have been proposed. For the parallel datasets, Wang et al.[33] extend the translation framework and propose a novel model for formality style transfer.

Table 1: The comparison of linguistic differences between Douban and Weibo

Linguistic features	Douban		Weibo		KS-test
Category	Mean	SD	Mean	SD	p-value
Pronouns	66.6	31.57	49.24	28.80	0.0003
1st person	31.1	17.73	22.34	14.26	0.0018
2st person	8.64	7.34	7.16	6.96	0.0994
3st person	5.12	4.45	4.01	3.97	0.0030
Family	5.14	5.66	3.53	4.77	0.0006
Friend	3.09	2.64	2.08	1.97	0.0006
Body	10.6	6.12	7.88	5.90	0.0001
Health	5.8	4.62	4.15	3.48	0.0001
Present tense	14	6.94	9.82	5.75	0.0000
Future tense	8.45	5.26	6.57	4.52	0.0030
Past tense	5.85	3.61	4.17	2.79	0.0030
Affective	65.3	24.39	58.59	23.29	0.0994
Swear	1.34	1.82	0.51	0.72	0.0000
Religion	2.55	2.64	2.65	2.51	0.0314
Work	18.8	8.95	16.73	9.97	0.0994
Negative	18.5	10.37	13.25	7.67	0.0018
Positive	36.5	14.95	37.77	15.96	0.6766
Leisure	16.6	10.61	19.49	14.11	0.0000
Anxiety	3.64	2.84	2.26	2.09	0.0000
Anger	4.23	3.24	2.64	2.34	0.0000

Pryzant et al.[24] focus on the bias in text language. They construct a new dataset containing about 180,000 parallel data and propose a model based on the BERT framework. However, the construction of parallel datasets mostly relies on manual annotation[2], so this type of method is not suitable for our task. For the similar reason, it motivates more works to investigate style transfer with non-parallel datasets. Shen et al.[27] propose a cross-aligned autoencoder to separate the underlying content from style. Fu et al.[5] aim to learn separate content representations and style representations using adversarial networks. The above methods share a common idea of disentangling the content and the style of the texts. But as they mainly focus on style transfer accuracy, they cannot achieve good content preservation. And Subramanian et al.[29] indicate that disentangled latent representation is hard to get and not necessary. Li et al.[15] propose to delete words associated with the source style and replace them with similar phrases retrieved from the target style corpus. This method can achieve a better balance between style transfer accuracy and content preservation. However, as the style of social media language is more ambiguous, it is difficult to achieve style transfer by simply replacing some specific words. Especially, Dai et al.[2] propose Style Transformer, which is the work most related to us. The model is in an adversarial generator-discriminator setting and takes Transformer[31] as the basic block. They can achieve better overall performance.

However, for our task, in addition to meeting the requirement of style transfer, the newly generated post should also be as consistent as possible with the user's personalized characteristics. The existing models rely only on textual content but ignore users' personalized characteristics. Therefore, the existing models still cannot meet the demand for personalized cross-platform post style transfer.

3 PRELIMINARY ANALYSIS

3.1 Data Preparation

Douban and Weibo are two of the most popular and biggest Chinese social media platforms[36]. The two platforms support some similar functions, such as sharing daily lives in the section of "`broadcast`" on Douban while "`weibo`" on Weibo. The users continuously generate a large amount of user-generated content, which contains valuable information for various studies.

In our task, we focus on the users who have accounts on Douban as well as Weibo, so we need to identify the users who are active on both platforms(cross-platform users). Following Liu et al.[17], some Douban users display self-introduction and Weibo account information in their personal descriptions. Based on this information, after matching and screening, we can get the cross-platform users. In total, we obtain a user list containing 1,390 cross-platform users. Based on the user list, we use the Douban and Weibo API to crawl more than 300,000 posts from each platform, these posts are all published after January 1, 2015. And we restrict the posts' length in our dataset to range from 3 to 100, any posts out of this restriction are removed, as too short or too long posts are always meaningless. Then we further take the following preprocessing steps. First, the URL in the post is deleted through regularized matching. Then mentions (@username) are replaced by @USER following Wang et al.[32]. Third, Jieba toolkit[1] is used for word segmentation, a vocabulary with 20K most frequent words is maintained. Finally, we obtain 96,233 Douban posts and 108,594 Weibo posts in total.

3.2 Data Analysis

3.2.1 The linguistic differences of users' posts on Douban and Weibo. To explore the linguistic differences between the two Chinese social media platforms, we compare their linguistic differences through deep learning as well as traditional data analysis methods.

Firstly, following Jaidka et al.[12], we attempt to classify whether a post is posted on Douban or Weibo by training a classification model. The classification model is trained using Transformer[31]. The accuracy reaches nearly 80%, which suggests there are identifiable differences between the language of the two platforms.

To further understand the differences, following Liu et al.[17] and Lin et al.[16], we analyze the frequency of different categories' words used on the two platforms utilizing the LIWC[23] dictionary. Concretely, we study the linguistic differences focusing on the use of pronouns words, emotion words, tense words, and some content words, etc. Similar to Liu et al.[17], we randomly sample 100 users and retrieve their recent 100 posts on the two platforms respectively. After some preprocessing steps, we apply the LIWC[23] dictionary to calculate the LIWC features of the 20,000 posts respectively.

The results are shown in Table 1. On the one hand, both the posts on Douban and Weibo mention many words or topics related to users' daily life, such as work, family, friend. This phenomenon indicates that the two platforms are both mainly used for sharing daily life by many users. On the other hand, there are significant linguistic differences between the two platforms. For example, the users express emotions more on Douban rather than Weibo, and more negative emotions are disclosed on Douban. The less negative words used on Weibo suggest that users tend to build a more positive social persona. The reason might be that the users try to avoid leaving negative impressions when facing a mixed audience[10]. More first-person pronouns are used on Douban, indicating that users prefer to use Douban rather than Weibo to express themselves.

[1]https://github.com/fxsjy/jieba

Table 2: Results of the regression model of users' posting behaviour on Douban

	First person	Swear	Tense	Family	Friend	Positive	Negative	Health
First person	0.263***	-0.007	0.051	0.150**	0.001	0.018	-0.084	0.004
Swear	-0.011	0.005	-0.023	0.008	-0.009	-0.007	0.021	-0.005
Tense	0.052*	-0.008	0.024	0.012	0.024	0.043	0.035	-0.023
Family	0.013	0.045*	0.042*	0.065***	0.001	-0.020	-0.014	0.001
Friend	0.053	-0.031	-0.049	-0.006	0.125***	0.009	0.001	0.111***
Positive	-0.015	0.040	0.015	-0.075*	-0.088**	0.028	-0.014	-0.071*
Negative	0.014	-0.017	0.002	0.022	0.052*	0.032	0.069***	0.040
Health	-0.028	0.020	-0.016	-0.019	0.002	-0.059**	-0.069**	0.063**

***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$

Table 3: Results of the regression model of users' posting behaviour on Weibo

	First person	Swear	Tense	Family	Friend	Positive	Negative	Health
First person	-0.044	-0.041	-0.022	0.094*	-0.094*	-0.037	-0.106*	0.044
Swear	0.052***	0.003	0.034*	0.030	0.025	0.018	0.016	0.030
Tense	0.051*	0.022	0.038	-0.035	0.011	-0.004	0.032	-0.063**
Family	-0.001	0.038*	-0.005	0.008	-0.053**	-0.054**	-0.052**	-0.043**
Friend	0.052	0.010	0.065*	0.074*	0.105***	0.101***	0.041	0.110**
Positive	0.071	0.080*	0.142***	-0.050	0.051	0.093*	0.027	0.007
Negative	-0.065**	-0.047	-0.070**	-0.001	-0.070**	-0.065**	-0.030	-0.037
Health	-0.034	0.000	-0.053**	-0.012	0.049*	-0.007	0.006	0.072***

***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$

Additionally, by counting the word frequency on the two platforms, we find that more emoticons are used on Weibo, which is related to the platform norms, because the Weibo platform officially provides many popular emoticons for the users. Moreover, the KS test is used to verify the significance of the linguistic difference between the two platforms. Therefore, we can obtain the insights that there are different language styles on different platforms and users tend to adapt their language to the style of the social media platforms. This phenomenon is common on the two Chinese platforms.

3.2.2 The correlation between user's new post and historical posts. Most of us have such an intuition that one user's posts on one platform usually have some common characteristics, such as the habit of using a certain kind of words, emoticons, etc. These behaviors reflect user's personalized characteristics to some extent. It is more likely that the user's newly published post has some similarities with the previously published posts in terms of expression.

To verify these intuitions, we analyze the use of affective words, pronouns and tense words, etc. to investigate the language correlation between the new post and historical posts. We randomly select 5,000 Douban posts and their corresponding historical posts, 5,000 Weibo posts and their corresponding historical posts for statistics. We use the LIWC[23] Chinese dictionary to count the usage of various words. For historical posts, we take the average of statistical values of various words used in historical posts.

After obtaining all this statistical information, we use the LIWC statistics of newly posted posts as dependent variables and the LIWC statistics of their corresponding historical posts as independent variables to establish regression models, respectively. The results(Coefficient and significance) of Douban and Weibo are shown in Table 2, Table 3 respectively. We can see that most of the LIWC features of new posts are significantly related to the LIWC features of historical posts. Therefore, the information of historical posts has a certain effect on new posts, users' new posts have some correlations to their historical posts. We can incorporate the information of historical posts to guide the generation of personalized posts.

4 MODEL

4.1 Problem Formalization

In this section, we first formalize the personalized cross-platform post style transfer task. Considering that we have two datasets $\mathbf{D_1}$ and $\mathbf{D_2}$ which contain posts from different social media platforms, where $\mathbf{D}_\mathrm{i} = \left\{\mathbf{x}_1^p, \mathbf{x}_2^p, \ldots, \mathbf{x}_N^p\right\}$ contains N input posts. For any j, $\mathbf{x}_j^p = \left\{w_1^p, w_2^p, \ldots, w_{|p_j|}^p\right\}$ represents one post with $|p_j|$ words. As we mentioned above, suppose that each platform has its own culture or style, and users tend to adapt their language to the target platform. So for all the posts in one social media dataset, they share some common characteristics which are different from other platforms, which we refer to as the language style of this platform. And users will express the same content in different language styles on different platforms. We denote the style of each platform as $\mathbf{s}_1$ and $\mathbf{s}_2$ respectively. The goal of cross-platform post style transfer is that: given one post $\mathbf{x}_i$ and the desired platform style $\mathbf{s}_j$, rewrite this post to a new one which has the style $\mathbf{s}_j$ and preserve the information in the original post as much as possible. But for the personalized cross-platform post style transfer task, the ultimate goal is to generate personalized post simultaneously when doing style transfer. So we take users' posting history into consideration, as we think users' personalized characteristics can be represented by users' posting history to some extent. For each post $\mathbf{x}_k^p$ $(k = 1, 2, \ldots, N)$ in dataset $\mathbf{D}_\mathrm{i} = \left\{\mathbf{x}_1^p, \mathbf{x}_2^p, \ldots, \mathbf{x}_N^p\right\}$, according to the posting time and user information, we get the corresponding users' historical posts on different platforms. We represent it as $\mathbf{H}_{k,0} = \left\{\mathbf{x}_1^h, \mathbf{x}_2^h, \ldots, \mathbf{x}_M^h\right\}$ and $\mathbf{H}_{k,1} = \left\{\mathbf{x}_1^h, \mathbf{x}_2^h, \ldots, \mathbf{x}_M^h\right\}$ respectively, where $\mathbf{H}_{k,0}$ means the posting history corresponding to post $\mathbf{x}_k^p$ in one platform, and $\mathbf{H}_{k,1}$ in another platform, it contains M user's historical posts.

Formally, given the above information, our task can be defined as generating a personalized, style-changed, and content-preserved post based on the user's post $\mathbf{x}$, the desired style $\hat{\mathbf{s}}$ as well as the user's posting history $\hat{\mathbf{H}}$:

$$\mathbf{y} = \underset{\mathbf{y}'}{\operatorname{argmax}}\, p\left(\mathbf{y}' \mid \mathbf{x}, \hat{\mathbf{s}}, \hat{\mathbf{H}}\right) \tag{1}$$

We denote the generated post as $\mathrm{Generator}(\mathbf{x}, \hat{\mathbf{s}}, \hat{\mathbf{H}})$.

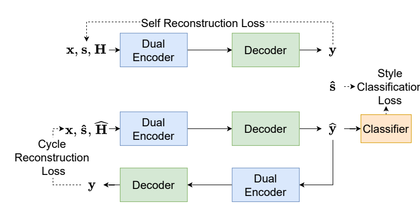

Figure 1: Model architecture of the proposed model.

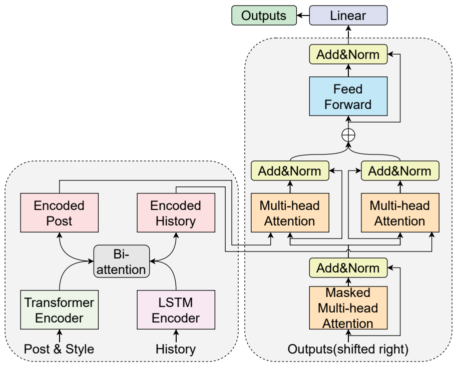

Figure 2: Overview of the generator of our model. The dual encoder includes a post encoder and a posting history encoder.

4.2 Model Overview

An overview of the proposed model is shown in Fig.1, the results of preliminary analysis provide much valuable information for the design of our model. There are two main modules: a generator for generating new post, and a classifier for classifying the posts with different styles, which give feedback to the extent to which the generated post conform to the expected style. So that the model can learn to generate post with the desired style gradually. Especially, as shown in Fig. 2, the generator contains two components: a dual encoder to encode the input post and user's posting history, and the bi-attention mechanism is applied to explore the interaction between new post and historical posts; a decoder to generate new post, where a gated attention routing layer is designed to control the expression of personality.

4.3 Dual Encoder

To get the representation both from the input post and its corresponding posting history, we design a dual encoder. The dual encoder consists of a post encoder and a posting history encoder, where the input of the post encoder is one post $\mathbf{x}$ with the desired style $\hat{\mathbf{s}}$ and the input of the posting history encoder is the corresponding posting history $\hat{\mathbf{H}}$ in the target social media platform.

For the post encoder, the backbone of the encoder is the Transformer framework[31]. In general, Transformer[31] has two major modules: encoder and decoder. We take the Transformer encoder to encode the input post $\mathbf{x}$. The input is mapped into a sequence of continuous representations $\mathbf{z}^p = \left(\mathbf{z}_1^p, \mathbf{z}_2^p, \ldots, \mathbf{z}_{|\mathbf{x}|}^p\right)$. Specifically, to tell the model which style of post needs to be generated, we add an extra style token to each input post to represent the desired style category. The calculation of $\mathbf{z}^p$ can be represented as:

$$\mathbf{z}^p = \mathrm{PostEncoder}(\mathbf{x}, \hat{\mathbf{s}}) \tag{2}$$

For the posting history encoder, we employ a bidirectional LSTM[9] to encode the corresponding posting history $\hat{\mathbf{H}} = \left\{\mathbf{x}_1^h, \mathbf{x}_2^h, \ldots, \mathbf{x}_M^h\right\}$. For each sentence $\mathbf{x}_i^h \in \hat{\mathbf{H}}(i = 1, 2, \ldots, M)$, each word $w_{i,j}^h \in \mathbf{x}_i^h \left(j = 1, 2, \ldots, \left|\mathbf{x}_i^h\right|\right)$ is first embedded into an embedding vector $v_{i,j}^h$, then mapped into hidden states $\mathbf{h}_{i,j}^h$, where $\mathbf{h}_{i,j}^h = \left[\overrightarrow{\mathbf{h}_{i,j}^h}; \overleftarrow{\mathbf{h}_{i,j}^h}\right]$ is the concatenation of the forward and backward hidden states. The process can be formalized as:

$$\overrightarrow{\mathbf{h}_{i,j}^h} = \mathrm{LSTM}\left(v_{i,j}^h, \overrightarrow{\mathbf{h}_{i,j-1}^h}\right) \tag{3}$$

$$\overleftarrow{\mathbf{h}_{i,j}^h} = \mathrm{LSTM}\left(v_{i,j}^h, \overleftarrow{\mathbf{h}_{i,j+1}^h}\right) \tag{4}$$

In specific, we use the last hidden state as the representation of each historical post. So we can get the representation of user's posting history $\mathbf{z}^h = \left(\mathbf{z}_1^h, \mathbf{z}_2^h, \ldots, \mathbf{z}_{|\hat{\mathbf{H}}|}^h\right)$ where each $\mathbf{z}_i^h = \left[\overrightarrow{\mathbf{h}_{i,|\mathbf{x}_i^h|}^h}; \overleftarrow{\mathbf{h}_{i,1}^h}\right]$ $\left(i = 1, 2, \ldots, |\hat{\mathbf{H}}|\right)$ means the representation of one historical post.

Based on our preliminary analysis, one user's new post usually have some common characteristics with the corresponding historical posts. To further capture useful information from our two encoders, we apply the bidirectional attention mechanism[26, 32] to explore the interaction and correlation between the input post and its corresponding historical posts. The use of bi-attention is inspired by Wang et al.[32] and Seo et al.[26]. Our intuition is that for some words in the user's post, there may be some words with similar meanings commonly used by the user in the user's historical posts, and these words are in the style of the target social media platform. On the other hand, for one historical post, the words in the new post correlate to it differently, so by establishing the interaction between historical posts and new posts, the representation of historical posts can be enriched.

To this end, we calculate post-aware attention and history-aware attention respectively. For the post-aware attention, the attention weights are defined as:

$$\alpha_{i,j}^h = \frac{\exp\left(f_{score}\left(\mathbf{z}_i^p, \mathbf{z}_j^h\right)\right)}{\sum_{j'=1}^{|\mathbf{z}^h|} \exp\left(f_{score}\left(\mathbf{z}_i^p, \mathbf{z}_{j'}^h\right)\right)} \tag{5}$$

$$f_{score}(\mathbf{Q}, \mathbf{K}) = \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} \tag{6}$$

where $f_{score}(\cdot)$ measures the semantic relations between the i-th word in new post and the j-th historical post. Especially, as the design in Transformer[31], we adopt the multi-head attention to jointly attend to information from different representation subspaces.

Then we get the post-aware historical posts representations $\mathbf{r}^h$, the i-th value can be defined as:

$$\mathbf{r}_i^h = \sum_{j=1}^{|\mathbf{z}^h|} \alpha_{i,j}^h \mathbf{z}_j^h \tag{7}$$

Analogously, we can calculate the history-aware attention in the similar way. Finally, we get the history-aware post representation as $\mathbf{r}^p$.

Lastly, to use the representations got from bi-attention layers as well as preserve the original information from the dual encoder as much as possible, we add the result of bi-attention to the output of the dual encoder, the result is defined as:

$$\mathbf{v}^p = \mathbf{z}^p + \mathbf{r}^h \tag{8}$$

$$\mathbf{v}^h = \mathbf{z}^h + \mathbf{r}^p \tag{9}$$

The final representation $\mathbf{v}^p$ and $\mathbf{v}^h$ both contain information from the new post and historical posts. Especially, $\mathbf{v}^p$ mainly includes the information from the new post and some auxiliary information from historical posts, $\mathbf{v}^h$ mainly includes the information from the historical posts and some auxiliary information from the new post.

4.4 Decoder

In general, conditioned on the post representation $\mathbf{v}^p$ and the historical posts representation $\mathbf{v}^h$, we adopt the Transformer[31] decoder to generate new post. We define the process to generate new post with the following probability:

$$\Pr\left(\mathbf{y} \mid \mathbf{v}^p, \mathbf{v}^h\right) = \prod_{t=1}^{|\mathbf{y}|} \Pr\left(y_t \mid \mathbf{v}^p, \mathbf{v}^h, \mathbf{y}_{<t}\right) \tag{10}$$

where $\mathbf{y}_{<t} = (y_1, y_2, \ldots, y_{t-1})$. And $\Pr\left(y_t \mid \mathbf{v}^p, \mathbf{v}^h, \mathbf{y}_{<t}\right)$ is a word distribution over vocabulary, it reflects the probability of generating the t-th word conditioned on the first $t-1$ generated words and other information in the t-th timestep. We introduce the procedure of calculating the probability in the following.

For the decoder, similar to the Transformer[31] encoder, it is also composed of one self-attention layer and one feedforward network layer[31]. In addition, we introduce two additional multi-head attention layers, which perform attention over the two encoders' output respectively. For the two additional attention layers, similiar to Zheng et al.[37], as different attention layers route to different input features, we name each group of attention operations as an attention route. We calculate the attention as the approach proposed by Zheng et al.[37]. For the post attention route, we take the representation of previously decoded tokens as the query and take the encoded post representation $\mathbf{v}^p$ as the key and value. The history attention route is similar to the post attention route, but takes the encoded historical posts representation $\mathbf{v}^h$ as the key and value. The calculation can be expressed as:

$$\mathbf{o}^p = \text{MultiHead}\left(\mathbf{e}_{prev}, \mathbf{v}^p, \mathbf{v}^p\right) \tag{11}$$

$$\mathbf{o}^h = \text{MultiHead}\left(\mathbf{e}_{prev}, \mathbf{v}^h, \mathbf{v}^h\right) \tag{12}$$

where $\mathbf{e}_{prev}$ represents the representation of previously decoded tokens. And we also employ a residual connection[7] around each attention module.

Finally, conditioned on the result of $\mathbf{o}^p$ and $\mathbf{o}^h$, we get the t-th word distribution over the vocabulary. Especially, we design an gated sum module to dynamically control the expression of personality. This process is described as:

$$\alpha = \sigma\left(\mathbf{W}\left[\mathbf{o}^p; \mathbf{o}^h\right]\right) + \mathbf{b}) \tag{13}$$

$$\Pr\left(y_t \mid \mathbf{v}^p, \mathbf{v}^h, \mathbf{y}_{<t}\right) = \text{softmax}\left(\alpha \odot \mathbf{o}^p + (1-\alpha) \odot \mathbf{o}^h\right) \tag{14}$$

where σ denotes the logistic sigmoid function, and $\mathbf{W}$ and $\mathbf{b}$ are learnable parameters, [;] denotes the concatenation operation.

4.5 Classifier

In the process of post generation, suppose that use $\mathbf{x}$ to represent the input post and $\mathbf{s}$ to represent its style. If we tend to do style transfer for the post $\mathbf{x}$ and the desired style is $\hat{\mathbf{s}}$. When $\hat{\mathbf{s}} = \mathbf{s}$, that is the self-reconstruction process, we can compare the generated post with the input post to evaluate the model performance. But when $\hat{\mathbf{s}} \neq \mathbf{s}$, we cannot establish supervision over the generation of such post, as we don't have a parallel corpus.

To solve the above challenges, we introduce a post style classifier, which can be trained using our non-parallel dataset, so as to feedback the style transfer accuracy to the generation model.

Similar to the method in Dai et al.[2], input one post to the classifier, the classifier will output its style category. Concretely, We train our classifier to classify three classes, the class 0 represents the post generated by Generator$(\mathbf{x}, \hat{\mathbf{s}}, \hat{\mathbf{H}})$ where $\hat{\mathbf{s}} \neq \mathbf{s}$, and the class 1 and class 2 represent the posts from two social media platforms respectively. When training the classifier, we label the original post $\mathbf{x}$ and the reconstructed post Generator$(\mathbf{x}, \mathbf{s}, \mathbf{H})$ as its corresponding style class, but label the generated post Generator$(\mathbf{x}, \hat{\mathbf{s}}, \hat{\mathbf{H}})$ where $\hat{\mathbf{s}} \neq \mathbf{s}$ as class 0. When training the generation model, we train the generator to maximize the probability of the generated post belonging to the expected style. So the generator can generate post with the desired style.

4.6 Learning Algorithm

When training the classifier, we minimize the cross-entropy loss, the loss function can be defined as:

$$\mathcal{L}_{\text{classifier}}(\phi) = -p_\phi(\mathbf{c} \mid \mathbf{x}) \tag{15}$$

where $\mathbf{x}$ is the input post, ϕ denotes the parameter set of the classifier.

In the generator training stage, when the input is one post $\mathbf{x}$ and its corresponding style $\mathbf{s}$, we can train the generator to reconstruct the post $\mathbf{x}$ by minimizing the negative log-likelihood, the loss function can be defined as:

$$\mathcal{L}_{\text{self}}(\theta) = -p_\theta(\mathbf{y} = \mathbf{x} \mid \mathbf{x}, \mathbf{s}, \mathbf{H}) \tag{16}$$

where θ denotes the parameter set of the generator. When input one post $\mathbf{x}$ and another style class $\hat{\mathbf{s}} \neq \mathbf{s}$, we use the classifier to judge how close the style of generated post Generator$(\mathbf{x}, \hat{\mathbf{s}}, \hat{\mathbf{H}})$ where $\hat{\mathbf{s}} \neq \mathbf{s}$ to the desired style. We train the generator by minimizing the negative log-likelihood of the class of the desired style $\hat{\mathbf{s}}$. The loss function is:

$$\mathcal{L}_{\text{style}}(\theta) = -p_\phi(\mathbf{c} = \hat{\mathbf{s}} \mid \text{Generator}(\mathbf{x}, \hat{\mathbf{s}}, \hat{\mathbf{H}})) \tag{17}$$

With the above loss function, the generation model can generate post in the desired style. However, in the case of Generator$(\mathbf{x}, \hat{\mathbf{s}}, \hat{\mathbf{H}})$ where $\hat{\mathbf{s}} \neq \mathbf{s}$, a potential problem is that if only the style classification loss is used, the generation model can simply generate keywords with strong $\hat{\mathbf{s}}$ style to fit the classification loss, which might cause great loss in content. So the cycle reconstruction process is introduced. We take the generated post Generator$(\mathbf{x}, \hat{\mathbf{s}}, \hat{\mathbf{H}})$, the original style $\mathbf{s}$, and the corresponding historical posts in the platform of post $\mathbf{x}$ as the input, and re-enter the information to the generation model, hoping that the model can restore the original sentence. Here we train the model by minimizing the negative log-likelihood loss, the loss function is defined as:

$$\mathcal{L}_{\text{cycle}}(\theta) = -p_\theta(\mathbf{y} = \mathbf{x} \mid \text{Generator}(\mathbf{x}, \hat{\mathbf{s}}, \hat{\mathbf{H}}), \mathbf{s}, \mathbf{H}) \tag{18}$$

Where $\hat{\mathbf{H}}$ denotes the historical posts in the platform corresponding to the style $\hat{\mathbf{s}}$, and $\mathbf{H}$ denotes the historical posts in the platform corresponding to the style $\mathbf{s}$. The training of the whole model contains the training of the classifier and the generator. We perform the training as the training process of GAN[2, 6]. We first pre-train our generator through the self reconstruction process, so that the generator has the generation ability to some extent. Then for each training iterator, we train the classifier for $N_{Classifier}$ steps, so that the classifier can classify posts with different styles. Then train the generator for $N_{Generator}$ steps, so that the generation model can have the ability of personalized post style transfer. With the continuous iteration, the modeling ability of the generator and the classifier is gradually enhanced. When the training tends to converge, we get the final trained model.

5 EXPERIMENT

5.1 Experimental Settings

Our model is implemented on the Pytorch framework. For the dual encoder, the post encoder uses 3 layers Transformer[31], and uses 4 attention heads for the multi-head attention. The history encoder employs 3 layers of bidirectional LSTM[9], and takes the last timestep hidden state as the representation of each historical post. For the decoder, 3 layers Transformer[31] is used. And when do self reconstruction, the teacher forcing method[34] is employed. But when do cycle reconstruction, the model generates the next token conditioned on the output of the last timestep. For the classifier, its architecture is similar to the Transformer encoder. Following Dai et al.[2], we add a <cls> token at the beginning of each input post, and feed the corresponding position output vector to the softmax layer to get the classification result. The hidden size, word embedding size, and the position embedding size for each module are 256 dimensions. The Adam optimizer[14] with the learning rate of 0.0001 is used for stochastic gradient descent. For the dataset, we randomly select 2000/2000 posts as the dev/test set, and the remaining data as the training set.

5.2 Comparisons

To evaluate the performance of style transfer, we compare our model with 5 baselines. They are CrossAlignment (Shen et al.[27]), MultiDecoder (Fu et al.[5]), StyleTransformer (Dai et al.[2]), DeleteOnly (Li et al.[15]), DeleteAndRetrieve (Li et al.[15]), which perform well when focusing transferring the attributes or stylistic paraphrases such as sentiment, tense[5, 11, 27].

To measure the performance of personalization, we compare our model with several variants of our proposed model. They are Persona-S, Persona-T, and Persona-C. For Persona-S, following Sennrich et al.[25], we feed the historical posts into the encoder as an extra token to the input post, so that the historical information can be merged into the model. As there is more than one historical post, we first get the representation of each historical post using the posting history encoder, then average these embeddings of

Table 4: Automatic evaluation results of style transfer

Model	ACC	BLEU	PPL
CrossAlignment	0.539	1.127	**105.069**
MultiDecoder	0.702	0.432	378.732
DeleteOnly	0.514	33.638	310.757
DeleteAndRetrieve	0.515	20.551	442.873
StyleTransformer	0.661	47.227	166.095
Our Model	**0.715**	**51.590**	203.041

Table 5: Automatic evaluation results of personalization

Model	ACC	BLEU	PPL	UMA	MRR
Persona-S	0.617	46.350	237.635	0.506	0.594
Persona-T	0.594	47.468	201.347	0.444	0.511
Persona-C	0.678	48.625	214.111	0.525	0.599
StyleTransformer	0.661	47.227	**166.095**	0.100	0.293
Our Model	**0.715**	**51.590**	203.041	**0.698**	**0.736**

the historical posts as the representation of historical information following Subramanian et al.[29]. For Persona-T, following Michel et al.[21], we set the representation of historical information as the start token of the target text. Concretely, we use the representation to guide the generation of the first token in the decoder rather than using a randomly initialized vector. For Persona-C, the model is similar to our final model, but no gated mechanism is employed in the decoder, the final representation is calculated by simply adding the outputs of the two multi-head attention layers.

5.3 Evaluation Metrics

Basically, Our task is similar to the basic text style transfer task, the generated posts should be fluent, content-preserved, and style-changed. So we evaluate the generated post in three dimensions.

Style accuracy: The style transfer accuracy measures the accuracy of the model in controlling the post style. We train one classifier on the training set of the experiment using FastText[13].

Content preservation: The content preservation aims to measure how much content is preserved when transferring the post from one platform style to another. We calculate the 5-gram BLEU score[22] between the generated post and the input post. Higher BLEU score[22] means better content preservation.

Fluency: Fluency is usually measured by the perplexity of the generated post. For an accurate evaluation of fluency, we train a 3-gram model using the KenLM toolkit[8].

Although the above three metrics can measure the models' performance on style transfer, there are no metrics related to personalization. Therefore, we introduce two additional personalized metrics, which measure how close the generated post is related to the corresponding users' posting history. For each data in the test set, we calculate the generation probabilities of the generated posts using the same input post but conditioned on different historical posts—one gold historical posts set corresponding to the input post and nine randomly generated historical posts sets. Following Majumder et al.[18], we expect the generated post conditioned on the gold historical posts set can get the highest generation probability.

Table 6: Model ablation study results on our dataset

Model	ACC	BLEU	PPL	UMA	MRR
Our model	**0.715**	51.590	203.041	**0.698**	**0.736**
-Post-aware Attn	0.643	**53.361**	177.518	0.672	0.716
-History-aware Attn	0.620	49.523	161.879	0.600	0.650
-Bi-Attn	0.605	46.660	**127.887**	0.533	0.601
-History	0.661	47.227	166.095	0.100	0.293

Table 7: Effects of the number of historical posts

Model	ACC	BLEU	PPL	UMA	MRR
History-5	0.666	48.205	**169.323**	0.317	0.429
History-10	0.658	47.740	188.916	0.470	0.576
History-15	0.685	48.112	175.745	0.537	0.636
Our model (History-20)	**0.715**	**51.590**	203.041	**0.698**	**0.736**

We measure the personalization performance through the following two metrics: the user matching accuracy (UMA)—the proportion where the generated post conditioned on the gold historical posts set is ranked first. And the mean reciprocal rank (MRR)—the average ranking of the generated posts conditioned on the gold historical posts. A higher UMA or MRR score indicates the model can make more use of the historical posts and achieve better personalization.

5.4 Experimental Results

The results for style transfer are shown in Table 4. It can be seen that our model achieves competitive results on almost all the style transfer metrics compared to the previous methods. We can further observe that: 1) the BLEU scores of CrossAlignment[27] and MultiDecoder[5] are relatively low, this indicates that these models' content preservation performance is poor. Although these models can change the post's style, the content of the generated post may be different from the input post completely. Xu et al.[35] get similar results, which supports our experimental results. The main reason is that these methods attempt to disentangle the content and style in the latent space, where all information is complicatedly mixed together. 2) DeleteOnly[15] and DeleteAndRetrieve[15] achieve higher BLEU scores, while their style transfer accuracy is relatively low and the generated posts are not fluency. The reason might be that the methods are a bit too simple. The style of social media posts is vaguer and the expression is more diverse, so the style transfer cannot be simply achieved by deleting or replacing some words. 3) The StyleTransformer[2] gets the best overall performance among these baselines, achieves a better balance of style transfer accuracy and content preservation. But our model still achieves competitive performance compared with StyleTransformer, which indicates the strong ability of our proposed model in style transfer.

The Table 5 presents the models' performance on personalization. We compare our model with the StyleTransformer and some variants of our proposed model. Our model outperforms all the comparison models on the two metrics. We can further observe that: 1) All the personalized models beat the baseline—StyleTransformer, which reveals that the user's historical posts are useful for generating personalized posts. 2) Our model achieves the best MRR and UMA by a large margin compared with the variants, which indicates the superiority of our model's design in capturing personalized information.

5.5 Further Discussions

5.5.1 Ablation Study. In this section, we study the effect of different components in our model. We compare the performance of our entire proposed model and its variants with some main components eliminated. Especially, we test the following components: 1) post-aware attention, 2) history-aware attention, 3) users' historical posts. As shown in Table 6, our model achieves the best overall performance among all the ablation models. Removing the history-aware attention decreases the MRR from 0.736 to 0.650 and the UMA from 0.698 to 0.600, which indicates the history-aware attention can help improve personalization significantly. Removing the post-aware attention also causes a slight decrease in MRR and UMA. When the bi-attention component is removed, we can observe a sharp decrease in MRR and UMA, but the results still surpass the baseline—StyleTransformer. Finally, when we remove the historical posts and only input the input post to the model, the ablation model cannot achieve any personalization.

5.5.2 Effects of the Number of Historical Posts. To investigate the impact of the number of historical posts M, we test our model by varying M and fitting all the other hyperparameters. The experimental results are shown in Table 7. As we can see, although there are some differences, all the models can achieve style transfer and personalization. And the best overall performance is achieved when the number is set to about 20. As the number of historical posts becomes larger, the personalization metrics increase gradually. The results indicate that within a certain range, as the number of historical posts increases, the model can capture more personalized information, which is consistent with our intuition.

6 CONCLUSION

In this paper, we focus on the user behavior of cross-platform content sharing. When users share content on multiple social media platforms, it is usually difficult for users to grasp the consistency between the language style of posts to be published and that of a platform as the problem of context collapse. To address this problem, we propose a novel task: automatically transferring users' posts from one platform's language style to the target platform's language style, while preserving the content and reflecting users' personalized characteristics as much as possible. We first conduct an study to investigate users' content sharing practices across two Chinese social media platforms. Then, we propose a novel model based on the Transformer framework and Bi-attention mechanism, which incorporates users' historical posts information to generate style-changed and personalized posts. Experimental results on the newly constructed dataset show that our model can achieve competitive performance compared to previous methods, especially in personalization.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (NSFC) under the Grants nos. 61902075 and 61932007.

REFERENCES

[1] Kholoud Khalil Aldous, Jisun An, and Bernard J Jansen. 2019. View, like, comment, post: Analyzing user engagement by topic at 4 levels across 5 social media platforms for 53 news organizations. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 13. 47–57.

[2] Ning Dai, Jianze Liang, Xipeng Qiu, and Xuan-Jing Huang. 2019. Style Transformer: Unpaired Text Style Transfer without Disentangled Latent Representation. In *ACL*. 5997–6007.

[3] Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. *arXiv preprint arXiv:1808.06640* (2018).

[4] Reza Farahbakhsh, Angel Cuevas, and Noël Crespi. 2016. Characterization of cross-posting activity for professional users across Facebook, Twitter and Google+. *Social Network Analysis and Mining* 6, 1 (2016), 1–14.

[5] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *AAAI*, Vol. 32.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[8] Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*. 187–197.

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

[10] Bernie Hogan. 2010. The presentation of self in the age of social media: Distinguishing performances and exhibitions online. *Bulletin of Science, Technology & Society* 30, 6 (2010), 377–386.

[11] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*. PMLR, 1587–1596.

[12] Kokil Jaidka, Sharath Guntuku, and Lyle Ungar. 2018. Facebook versus Twitter: Differences in Self-Disclosure and Trait Prediction. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 12.

[13] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2016).

[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[15] Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, Retrieve, Generate: a Simple Approach to Sentiment and Style Transfer. 1865–1874.

[16] Han Lin and Lin Qiu. 2013. Two sites, two voices: Linguistic differences between Facebook status updates and tweets. In *International Conference on Cross-Cultural Design*. Springer, 432–440.

[17] Baoxi Liu, Peng Zhang, Tun Lu, and Ning Gu. 2020. A reliable cross-site user generated content modeling method based on topic model. *Knowledge-Based Systems* 209 (2020), 106435.

[18] Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating personalized recipes from historical user preferences. *arXiv preprint arXiv:1909.00105* (2019).

[19] Lydia Manikonda, Venkata Vamsikrishna Meduri, and Subbarao Kambhampati. 2016. Tweeting the mind and instagramming the heart: Exploring differentiated content sharing on social media. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 10.

[20] Alice E Marwick and Danah Boyd. 2011. I tweet honestly, I tweet passionately: Twitter users, context collapse, and the imagined audience. *New media & society* 13, 1 (2011), 114–133.

[21] Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. *arXiv preprint arXiv:1805.01817* (2018).

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

[23] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. *The development and psychometric properties of LIWC2015*. Technical Report.

[24] Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. In *Proceedings of the aaai conference on artificial intelligence*, Vol. 34. 480–489.

[25] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *HLT-NAACL*. 35–40.

[26] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* (2016).

[27] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. *arXiv preprint arXiv:1705.09655* (2017).

[28] Manya Sleeper, William Melicher, Hana Habib, Lujo Bauer, Lorrie Faith Cranor, and Michelle L Mazurek. 2016. Sharing personal content online: Exploring channel choice and multi-channel behaviors. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 101–112.

[29] Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text style transfer. *arXiv preprint arXiv:1811.00552* (2018).

[30] Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "Transforming" Delete, Retrieve, Generate Approach for Controlled Text Style Transfer. In *EMNLP-IJCNLP*. 3269–3279.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[32] Yue Wang, Jing Li, Irwin King, Michael R Lyu, and Shuming Shi. 2019. Microblog hashtag generation via encoding conversation contexts. *arXiv preprint arXiv:1905.07584* (2019).

[33] Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2020. Formality Style Transfer with Shared Latent Space. In *Proceedings of the 28th International Conference on Computational Linguistics*. 2236–2249.

[34] Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation* 1, 2 (1989), 270–280.

[35] 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 *ACL*. 979–988.

[36] Peng Zhang, Haiyi Zhu, Tun Lu, Hansu Gu, Wenjian Huang, and Ning Gu. 2017. Understanding relationship overlapping on social network sites: a case study of Weibo and Douban. *Proceedings of the ACM on Human-Computer Interaction* 1, CSCW (2017), 1–18.

[37] Yinhe Zheng, Rongsheng Zhang, Minlie Huang, and Xiaoxi Mao. 2020. A pre-training based personalized dialogue generation model with persona-sparse data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9693–9700.

[38] Changtao Zhong, Hau-wen Chang, Dmytro Karamshuk, Dongwon Lee, and Nishanth Sastry. 2017. Wearing many (social) hats: How different are your different social network personae?. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 11.