# Unsupervised Style Transfer in News Headlines via Discrete Style Space

**Qianhui Liu, Yang Gao,[*] Yizhe Yang**
School of Computer Science and Technology,
Beijing Institute of Technology, Beijing, China
{3120201048,gyang,yizheyang}@bit.edu.cn

## Abstract

The goal of headline style transfer in this paper is to make a headline more attractive while maintaining its meaning. The absence of parallel training data is one of the main problems in this field. In this work, we design a **d**iscrete style space for unsupervised **h**eadline **s**tyle **t**ransfer, short for **D-HST**. This model decomposes the style-dependent text generation into content-feature extraction and style modelling. Then, generation decoder receives input from content, style, and their mixing components. In particular, it is considered that textual style signal is more abstract than the text itself. Therefore, we propose to model the style representation space as a discrete space, and each discrete point corresponds to a particular category of the styles that can be elicited by syntactic structure. Finally, we provide a new style-transfer dataset, named as **TechST**, which focuses on transferring news headline into those that are more eye-catching in technical social media. In the experiments, we develop two automatic evaluation metrics — style transfer rate (STR) and style-content trade-off (SCT) — along with a few traditional criteria to assess the overall effectiveness of the style transfer. In addition, the human evaluation is thoroughly conducted in terms of assessing the generation quality and creatively mimicking a scenario in which a user clicks on appealing headlines to determine the click-through rate. Our results indicate the D-HST achieves state-of-the-art results in these comprehensive evaluations.

## 1 Introduction

A style makes sense under pragmatic use and becomes a protocol to regularize the manner of communication [Jin et al.2022, Khalid and Srinivasan2020]. So, the task of text style transfer is to paraphrase the source text in a desired style-relevant application [Toshevska and Gievska2021]. In practical use, the style is data-driven and task-oriented in different area [Jin et al.2022].

The absence of parallel training data for a certain style is one of the difficult problems. Continuous latent space mapping is a typical method for unsupervised style transfer to address the issue. Guo et al. (2021; Liu et al. (2020) model the latent space to a Gaussian distribution. Points in latent space are moved to the target representation with some style guidance. Nangi et al. (2021; John et al. (2018; Romanov et al. (2018) disentangle the continuous latent representation purely according to its content, and replace the source attribute to the target one. However, there are two problems of the continuous space approach. Firstly, the style is highly abstract so that it is unstable and too sparse to accurately represent the style in the continuous space. Second, the continuous vector-based representation is difficult to manipulate and cannot be examined at a finer level. To control the style transfer and enhance its explainability, several kinds of discrete signals are used to represent the style. For instance, Reid and Zhong (2021; Tran et al. (2020; Li et al. (2018) employ Mask-Retrieve-Generate strategy to decompose style attributes by word-level editing actions. But, these methods express styles in a highly discrete way which fail to capture the relationships between words or sentences.

To more effectively describe the style in a highly abstract and discrete manner while also capturing the semantic relations in the texts, we propose a latent and **d**iscrete style space for **h**eadline **s**tyle **t**ransfer,

[*] Corresponding author.

abbreviated as **D-HST**. This model decomposes style-dependent text generation into content-feature extraction and style modeling. Therefore, we design a dual-encoder and a shared single-decoder framework to accomplish the overall generation. Due to the lack of parallel training data, we have to synthesize adequate training pairs to accommodate the content extraction and the style modeling. Given a target stylistic headline, we first automatically generate a content-similar input as well as style-consistent input for feeding the dual encoders. As the textual style signal is expected to be rather abstract and limited compared to the text itself, we propose to model the style representation space as a discrete space, with each discrete point denoting a particular category of the styles that can be elicited by syntactic structure.

Also, we provide a new style-transfer dataset derived from the real scenarios, named as **TechST**, which transfers news headlines into the ones that are more attractive to readers. Although several datasets are currently available for this purpose [Jin et al. (2020], but the appealing styles—such as humor and romance—are taken from fictional works of literature, which we believe makes them unsuitable for usage as an *attractive* style for headlines. In the experiments, we design two automatic evaluation metrics, including style transfer rate (STR) and style-content trade-off (SCT) - along with a few traditional criteria to assess the overall of the style transfer. Additionally, the quality of the generation is thoroughly evaluated, and the click-through rate is calculated by creatively simulating a scenario in which a user clicks on attractive headlines. Our findings show that the D-HST performs at the cutting edge in these thorough assessments. In conclusion, our article mainly has the following contributions.

- We propose an unsupervised style transfer method with discrete style space, which is capable of disentangling content and style.

- We propose new metrics in automatic evaluation and human evaluation, and achieves state-of-the-art results in these comprehensive evaluations.

- We provide a novel dataset derived from actual events to convert news headlines into catchy social media headlines.

## 2 Related Work

**Attractive Headline Generation** It is crucial to generate eye-catching headlines for an article. Gan et al. (2017) proposes to generate attractive captions for images and videos with different styles. Jin et al. (2020) introduces a parameter sharing scheme to generate eye-catchy headlines with three different styles, humorous, romantic, and clickbait. Li et al. (2021) proposes a disentanglement-based model to generate attractive headlines for Chinese news. We build upon this task by rewriting source headlines to attractive ones.

**Text Style Transfer** There are mainly three kinds of methods used in TST task. 1) **Modeling in the Latent space** Mueller et al. (2017; Liu et al. (2020) use continuous space revision to search for target space. Shen et al. (2017; Sun and Zhu () learn a mapper function in source and target space. John et al. (2018; Romanov et al. (2018; Hu et al. (2017) explicitly disentangle content and style in latent space. However, the style is highly abstract so that it is unstable and too sparse to accurately represent the style in the continuous space. 2) **ProtoType Editing** It is a word replacement method. Li et al. (2018; Tran et al. (2020) propose three-stage methods to replace stylist words with retrieved words in the target corpus. Reid and Zhong (2021) uses Levenshtein editing to search target stylist words. These methods work well on Content-Preferences dataset, like sentiment, debias. 3) **Control Code Index** Keskar et al. (2019; Dai et al. (2019) use a control sign embedding to controls the attribute of generated text. Yi et al. (2021) controls style using a style encoder. These methods don't learn style in a fine-grained way and the style space is a block-box. We combine the first and third methods, using a control code to control style and modeling a style space with appropriate distribution.

To model the discrete style in an unsupervised fashion, we propose to inherit the third and fourth methods. Specifically, we construct pseudo data to enrich the content-based parallel data and style-based para. Further, different from the previously styled latent space, we model it as a discrete one based on the claim that style is highly abstract and more sparse compared to content. We will describe this in detail in the next section.

## 3 Methodology

We are given samples $Y = \{y_1, y_2, \dots y_m\}$ from the style dataset $S$. The objective of our task is to transfer a headline sentence to a new headline equipped with the style of the target data $S$, while maintaining its originally semantic content.

### 3.1 Model Overview

Our proposed **D-HST** model consists of a duel-encoder and a single shared decoder in an unsupervised setting. It begins by constructing a pseudo-parallel dataset which comprises of two pairs of inputs-and-outputs. One of the inputs is $X^Y_{cont}$, which is generated by using a pre-trained paraphrasing model and has input that is content-similar to output $Y$. The other input is $X^Y_{style}$, which is collected in style dataset $S$ and uses inputs of sentences with the same style as output $Y$ based on the defined style (Section 3.2).

The model structure is described in Section 3.3. One of the inputs is **content input** $X^Y_{cont}$ encoded by a content encoder, then fed into a content pooling to extract its sentence-level feature, denoted as $Z^Y_{cont} = pool_{cont}(enc_{cont}(X^Y_{cont}))$. Similarly, the other input is **style input** $X^Y_{style}$ encoded by a style encoder, then fed into a style pooling to get style representation $Z^Y_{style}$. The hypothesis is that the pooling serves as a bottleneck which can disentangle the representation of content and style with help of proper loss function (Section 3.4). The overall model architecture is shown in Figure 1.

### 3.2 Pseudo Parallel Data Construction

**Content Input** Prior work has demonstrated that paraphrasing techniques can translate source sentences into standard written sentences while maintaining their substance [Mitamura and Nyberg (2001)]. In our approach, we assume that a special style (such as attractiveness, informality in the experiment) of a sentence can be removed after paraphrasing. We use a pretrained paraphrasing model[0] to remove stylist attribute, and construct the content inputs $X^Y_{cont}$.

As the paraphrasing model often produces multiple outcomes, in the experiment, we select top 5 generations as a candidate set for the content input. Then, we calculate bertscore to estimate the similarity between the generated candidates and the output $Y$. Only candidates with similarity between 0.75 to 0.95 are kept to preserve as much content information as possible and prevent significantly overlapping generation.

**Style Input** We suppose that a certain syntactic structure can reflect the style. For example, attractive headlines often employ interrogative questions; informal conversations frequently use ellipsis; and impolite language often employ imperative sentences. To collect more parallel headlines to train the style-based modules, we construct the style input $X^Y_{style}$ that shares the same syntactic structure yet different content with target $Y$, from the data in style dataset $S$. In order to filter out the content information in the style input, we use a set of sentences $C^Y_{style}$ that share the same syntactic structure for $X^Y_{style}$, then average these sentences with a learnable parameter.

Specifically, we use a chunk parser FlairNLP[1] to get the syntactic structure of these headlines. We first get the chunk label for each word using the chunk parser. Then, we merge the spans having the same label. Based on the assumption that words such as ”who”, ”whether” and ”how” are function words that guide special sentence patterns, we set a separate label QP to mark the leading words of interrogative sentences. We get some distinct syntactic structures, each of which has some corresponding headlines. We assume that if one syntactic structure occurs in less than 10 headlines, it is not representative. Then, we filter the syntactic structure and its corresponding sentences if its syntactic structure occurs in less than 10 headlines. Table 1 shows examples of processed syntactic structures and their corresponding sentences.

[0] https://huggingface.co/tuner007/pegasus_paraphrase

[1] https://github.com/flairNLP/flair

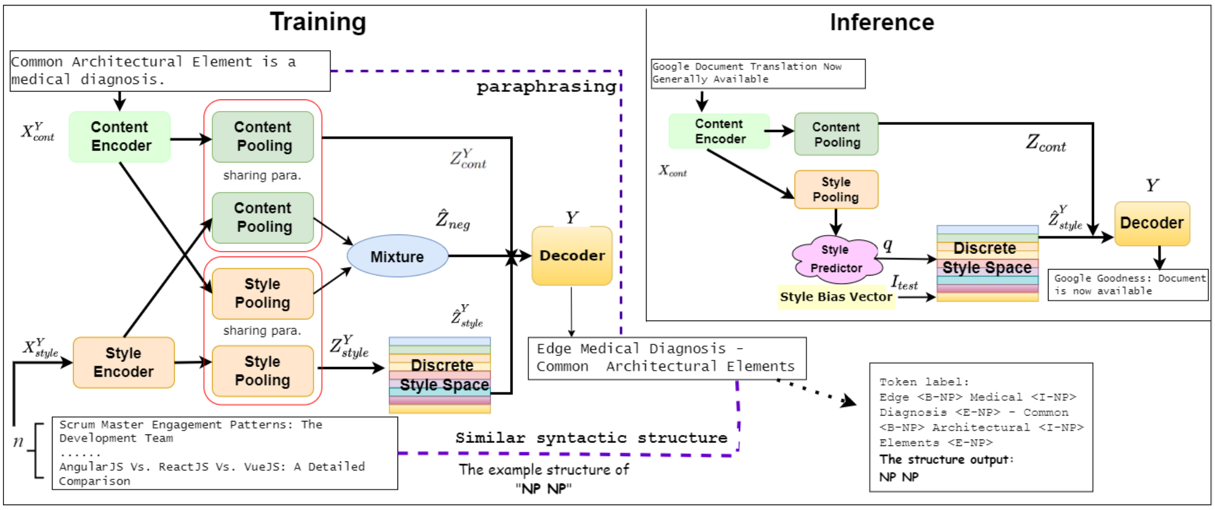
Training
Common Architectural Element is a medical diagnosis.
paraphrasing
$X^Y_{cont}$
Content Encoder
Content Pooling
sharing para.
Content Pooling
$Z^Y_{cont}$
$\hat{Z}_{neg}$
Mixture
Y
Decoder
Style Pooling
sharing para.
Style Pooling
$\hat{Z}^Y_{style}$
$X^Y_{style}$
Style Encoder
$Z^Y_{style}$
Discrete Style Space
Edge Medical Diagnosis - Common Architectural Elements
n
Scrum Master Engagement Patterns: The Development Team
......
AngularJS Vs. ReactJS Vs. VueJS: A Detailed Comparison
Similar syntactic structure
The example structure of "NP NP"
Inference
Google Document Translation Now Generally Available
Content Encoder
Content Pooling
$Z_{cont}$
$X_{cont}$
Style Pooling
Y
Decoder
$\hat{Z}^Y_{style}$
Style Predictor
q
Discrete Style Space
$I_{test}$
Style Bias Vector
Google Goodness: Document is now available
Token label:
Edge <B-NP> Medical <I-NP> Diagnosis <E-NP> - Common <B-NP> Architectural <I-NP> Elements <E-NP>
The structure output:
NP NP

Figure 1: The framework of the D-HST model. The training phase and inference phase are depicted in the figure.

| Syntactic Structure | QP VP NP | CD NP VP |
|---|---|---|
| Sentences | How to Become a DevOps Engineer<br>How to Scale Your SaaS Business<br>How to Do API Testing | 3 Tech Debt Metrics Every Engineer Should Know<br>7 Top Kubernetes Health Metrics You Must Monitor<br>10 Software Testing Interview Questions You Haven't Heard Before |

Table 1: Examples of syntactic structures and their corresponding headline sentences. These examples indicate that some sentences can express the same style representation by syntactic structure.

### 3.3 Model Architecture

The duel-encoder and the shared decoder are both based on standard Transformer model [Vaswani et al. (2017)]. The content inputs and style inputs are both encoded by their separate encoders, that are content encoder and style encoder, respectively. Each token is fed to the encoder and obtains embeddings $\{e_1, e_2, ..., e_{|X|}\} = enc(X)$, where $|X|$ is the length of the input sentence, $e_t \in R^H$, $H$ is the dimension of transformer.

**Feature Extractor** To facilitate the disentanglement between the content semantics and the stylistic attributes, we elicit their distinct features by pooling the multi-dimensional representation in accordance with the method used in Liu and Lapata (2019). Specifically, a multi-head pooling is adopted to extract features. We employed attention $a_t$, where $t$ represents a token, to calculate its importance score for the whole sentence. The equation is :

$$\alpha_t = \frac{\exp\ a_t}{\sum_{t\in|X|} \exp\ a_i} \tag{1}$$

$$a_t = k_t e_t \tag{2}$$

where $k_t \in R^H$ is a learnable parameter. The value of each token $V_t$ is also computed using a linear projection of $e_t$. Finally, we take a weighted average to get the pooling output $Z$.

$$Z = \sum_{t\in|X|} \alpha_t V_t \tag{3}$$

**Discrete Style Space** Inspired by Hosking and Lapata (2021) who claim style is limited and sparse, we therefore propose to extract a specific style from a discrete style space. The space maintains a discrete table $C \in R^{K \times D}$, $K$ is the number of style categories[2], equal to the number of distinct syntactic structure in style dataset $S$. We use $q$ to represent the category distribution and $\tilde{q} \in [0, K]$ to represent the sampled category. The category distribution $q$ is mapped from the style pooling $Z^{Y}_{style}$, and it can be formulated as $p(q|Z^{Y}_{style})$.

Finally, we draw $\tilde{q}$ from the Gumbel-Softmax distribution of $q$. The equation can be written as:

$$\tilde{q} \sim \text{Gumbel-Softmax}(q) \tag{4}$$

The style representation, $\hat{Z}^{y}_{style} = C(\tilde{q})$, maps from the discrete code $\tilde{q}$. $\hat{Z}^{Y}_{style}$ ought to be as near as the input $Z^{Y}_{style}$. So we get a loss term:

$$\mathcal{L}_q = \parallel Z^{Y}_{style} - sg(C(\tilde{q})) \parallel_2 \tag{5}$$

Because the gradient could be broken at stop gradient $sg$, the loss is not derivable. We employ a reparameterization trick [Kingma and Welling (2013)] to update parameters and exponential moving average [Roy et al. (2018)] strategy to update the discrete table.

**Style Bias** We assume that each sentence has its own style score. For example ”You Can’t Reset Your Fingerprint” is more obviously attractive than ”AI-Assisted Coding with Tabnine” in terms of its expressing style, although they are both in style dataset. Therefore, we manually rank each sentence in the style dataset $S$ based on external knowledge $I_{test}$. Details of the external knowledge are shown in Section 5.

We believe that syntactic structure can be used to define the style category and that sentences with the same structure may score similarly in terms of style. Each style is expected to be encoded into a specific category, and categories with higher style scores are more likely to be selected in inference. $I \in R^{K}$ is a one-hot vector and serves as a pre-labeled supervisory signal, representing the correspondence between styles and categories. For example, $I_m \in R^{K}$ encodes the style category to which the sentence $m$ belongs. In training phase, we expect each style is encoded into a specific category, so we let the output of the category distribution $q$ fit supervisory signal $I$. The equation can be written as:

$$\mathcal{L}_r = \parallel I - \text{softmax}(q) \parallel_2 \tag{6}$$

In inference phase, for all sentences, we use the fixed discrete style bias distribution $I_{test} \in R^{K}$ to increase the probability of choosing a high-scoring style. And we set the probability for each category in $I_{test}$ to be the normalized style score.

**Mixture Module** We also design a mixture module to serve as negative knowledge to guide decoder to leave away from the content of $X^{Y}_{style}$ and the style of $X^{Y}_{cont}$. We use a small full connect network with the concatenation of $Z_{cs} = pool_{cont}(enc_{style}(X^{Y}_{style}))$ and $Z_{sc} = pool_{style}(enc_{cont}(X^{Y}_{cont}))$ as input, written as $Z_{neg} = MLP(Z_{cs}, Z_{sc})$

Finally, the overall hidden representation $Z$ can be written as $Z = Z^{Y}_{cont} + \hat{Z}^{Y}_{style} + Z_{neg}$. And the target distribution $p(Y|Z) = dec(Z)$.

### 3.4 Model Training

We first describe the training process that makes the model to capture its local independence information separately.

We set triples $((X^{Y}_{cont}, X^{Y}_{style}), Y)$ as input and output, respectively. To produce strong style signals, we use a set of style sentences $C^{Y}_{style}$ in the same style as $Y$. The selection strategy has been described in Section 3.2 and the style representation is weighted with a learnable parameter $\kappa$, such as $Z^{Y}_{style} = \sum_{c_i \in C^{Y}_{style}} pool_{style}(enc_{style}(c_i))\kappa_{ci}$. Then, $\hat{Z}^{Y}_{style}$ is sampled from the style space. It is trained

[2] $K$=324 in our TechST dataset

to generate target $Y$ with the overall hidden representation $Z$, which is the sum of content encoding $Z^Y_{cont}$, style encoding $\hat{Z}^Y_{style}$ and negative knowledge encoding $Z_{neg}$. The factorised reconstruction loss term can be written as:

$$\mathcal{L}_Y = \sum_t \log p(w_t|w_1, w_2...w_{t-1}; Z) \tag{7}$$

The final objective function is:

$$\mathcal{L} = \mathcal{L}_Y + \delta\mathcal{L}_q + \epsilon\mathcal{L}_r \tag{8}$$

### 3.5 Inference

Since we don't have any style input $X^Y_{style}$ for inference, only $X_{cont}$ in source dataset is available and transferred to the defined target style. As such, the well-trained style encoder and mixture module can not be directly adopted in the inference. To fill this gap, we further train a style predictor module to alternatively select a sample to represent the most stylistic category for the following decoder. This predictor is formulated as $p(q|X^Y_{cont}) = MLP(pool_{style}(enc_{cont}(X^Y_{cont})))$. The additional predictor is trained to predict the well-trained style category distribution $q$ through $X^Y_{cont}$. $q$ is mapped from $Z^Y_{style}$ and represents as $p(q|Z^Y_{style})$. So we distill the distribution $p(q|X^Y_{cont})$ to the well-trained distribution $p(q|Z^Y_{style})$. The loss term is:

$$\mathcal{L}_{KL} = -KL(sm(p(q|X^Y_{cont}))||sm(p(q|Z^Y_{style}))) \tag{9}$$

where $sm$ is short for softmax function. In inference phase, we sample $\tilde{q} \sim ((1-\gamma)\text{sm}(q) + \gamma I_{test})$ 3 times and generate 3 candidate outputs. Finally, we select the one with highest content preservation with the input, calculated by bertscore.

## 4 Tasks and Datasets

For the headline style transfer task, we focus on attractive news headline transfer on technology topics. Technology news headlines are always formal. For example, "Google Document Translation Now Generally Available." is a common style for an event headline. On the contrary, technology blog headlines in social media tend to be special and catch readers' eyes. In this paper, we define this kind of headline as "**Attractive**" style. To highlight the characteristics of style, the previous example can be transferred as "Google Goodness: Document Now Available". The goal of this task is to transfer the formal news headlines to more attractive blog headlines in technology domain.

**Datasets** Our attractive technology dataset **TechST** was crawled from Dzone[3], including stylistic technology blog headlines and users' pageviews. This data was used to train the style transfer model. We also crawled technology news headlines from InfoQ[4] as non-stylistic headlines for testing. The task is to transfer the headlines in InfoQ to a new style that is modelled with the Dzone dataset. Both of them were crawled from the beginning to November 2011. We filtered out the blog headlines with pageviews less than 500 and the ones more than 22 words as we believe shorter headlines are attractive. Finally, we get 60,000 samples for training and 2,000 samples for testing.

We also use a cornerstone dataset Grammarly's Yahoo Answers Formality Corpus (GYAFC) [Rao and Tetreault (2018)] for formality transfer. It contains 53,000 paired formal and informal sentences in two domains. To meet our requirement of unsupervised style transfer setting, the task is to transfer the formal sentences to informal ones. Only informal sentences in the Family and Relationships categories were used for training and validation.

## 5 Experiments and Results

**External Knowledge** As mentioned in Section 3.3, external knowledge is used to estimate the style strength. To some extents, users' pageviews reflect attractiveness of the style. We first parsed all the

[3] https://dzone.com/

[4] https://www.infoq.com/

syntactic structures of sentences in the style dataset. Then, we calculate average-pageviews for each syntactic structure. The more average-pageviews the structure receives, the higher style score it has. We acknowledge that style isn't the only factor that affects pageviews, content also contributes to it. For example, headlines with syntactic structure like "NP VP" are common, but some headlines with such structure may have high pageviews. To eliminate the impact of content, we add a pageview variance term. Specifically, if sentences with same syntactic structure show little pageview variance, it is speculated that pageviews are determined by the syntactic structure. On the contrary, if the variance is significant, it suggests that other elements, such as content, are influencing pageviews. As such, the style score must be penalized. Finally, we define our style score as:

$$I_{test}^{i} = \frac{mean(a)^{\omega}}{var(a)^{\nu}} \tag{10}$$

$I_{test}^{i}$ represent the style score of category $i$, $a$ is the collection of the sentences having style $i$. $\omega$ and $\nu$ are hyperparameters.

For GYAFC dataset, no such corresponding information is provided, so we set all syntactic structures the same style score.

**Experiment Setup** We use 6-layers transformers to train our model. Each transformer has 8 attention heads and 768 dimensional hidden state. Dropout with 0.1 was added to each layer in the transformer. Encoder and decoder initialized from BART base. Hyperparameters $\delta$ and $\epsilon$ in loss function are set to 0.5. In external knowledge building, we set $\omega = 2$, $\mu = 0.05$.

We trained our model on a 3090 GPU for 20 epochs taking about 5 hours with gradient accumulation every 2 steps. We chose the best checkpoint for the testing through a validation process.

**Baselines** We compared the proposed model against the following three strong baseline approaches in text style transfer: **BART+R** [Lai et al. (2021)] is trained by fine-tune BART model with an extra BLEU reward and style classification reward. This model uses parallel dataset. In order to meet our requirement of unsupervised style transfer setting, we used pseudo-parallel data $X_{cont}^{Y}$ and $Y$ as input and target in the following experiment. **StyIns** [Yi et al. (2021)] leverages the generative flow technique to extract stylistic properties from multiple instances to form a latent style space, and style representations are then sampled from this space. **TSST** [Yi et al. (2021)] proposes a retrieval-based context-aware style representation that uses an extra retriever module to alleviate the domain inconsistency in content and style.

### 5.1 Automatic Metrics

To quantitatively evaluate the effectiveness of style transfer task which calls for both the transfer of styles as well as the preservation of content semantics, we newly designed two metrics of Style Transfer Rate (STR) and Style-Content Trade-off (SCT), respectively.

**Content Preservation (CP)** It is calculated by the similarity between the input and the transferred output leveraged by standard metric Bertscore [Zhang et al. (2019)].

**Style Transfer Rate** The traditional style transfer methods [Lai et al. (2021)] use a well-trained style classifier to testify if a sentence has been successfully transferred into a targeted style. But, this method is more suitable for polar word replacement, such as sentiment transfer in review generation. For the cases of eye-catching or written formality transfer, we propose a rule-based yet easy-to-use transfer metric, named as STR. We calculate the STR according to the percentage of syntactic structures changed between the generated output and its input as follows:

$$STR = \frac{\sum_{i \in C_{test}} \text{structure}(X_{cont}^{i}) \neq \text{structure}(O^{i})}{|C_{test}|} \tag{11}$$

where $|C_{test}|$ is the number of testing data, $X_{cont}^{i}$ and $O^{i}$ represent content input and generated output, respectively.

| Dataset | Model | CP | STR | SCT | PPL |
|---|---|---|---|---|---|
| TechST | StyIns | 0.773 | 0.377 | 0.253 | 48.39 |
| | BART+R | **0.962** | 0.394 | 0.280 | 92.48 |
| | TSST | 0.874 | 0.488 | 0.313 | 104.68 |
| | D-HST | 0.665 | **0.846** | **0.372** | **15.48** |
| GYAFC | StyIns | 0.811 | 0.666 | 0.366 | 26.51 |
| | BART+R | **0.896** | 0.663 | 0.381 | 12.61 |
| | TSST | 0.829 | 0.625 | 0.356 | 23.19 |
| | D-HST | 0.641 | **0.944** | **0.382** | **10.11** |

Table 2: The automatic evaluation results on our model and all baselines on both TechST and GYAFC datasets.

| Models | Interestedness | Fluency |
|---|---|---|
| D-HST | 1.711 | 1.763 |
| StyIns | 1.05 | 1.413 |
| BART+R | 1.219 | 1.906 |
| TSST | 1.181 | 1.463 |

Table 3: Human evaluation.

**Style-Content Trade-off** In order to integrate the STR and CP into a single measure, we take their harmonic means as follows:

$$SCT = \frac{2}{\frac{1}{\mathrm{STR}} + \frac{1}{\mathrm{CP}}} \tag{12}$$

**Language Fluency** We fine-tuned the GPT-2 model (Radford et al., 2019) on our stylistic dataset $S$ and use it to measure the perplexity (PPL) on the generated outputs.

### 5.2 Overall Performance

We compared the performance of our model against with the baselines in Table 2. D-HST performs the best across all the metrics except for the CP metric. From the results we can find that, firstly, our model achieves very obvious advantage in STR metric (nearly 50% margin) indicate the thorough and outstanding performance on style transfer; Secondly, our D-HST identifies the most harmonious balance point between content preservation and style transfer revealed by the SCT metric; Thirdly, our language model GPT-2 was fine-tuned in stylistic data, therefore, the PPL metric favors fluent sentences adhere more closely to the given style format. Although the BART+R model receives best fluency in human evaluation (Table 3), it mostly fails in our automatic fluency metric. When evaluating the content preservation, we discourage the CP metric from being as high as possible since the extremely high similarity (like close to 1) implies the exactly same words are used in sentences. However, what is required is a change in style that involves a particular number of words. Therefore, we argue that CP is acceptable around 0.64-0.66[5], which can preserve the source content while transferring the style.

To gain further insight on the performance of the style transfer, we sampled real examples from our model and baselines on TechST dataset, as shown in Table 4. StyIns and BART+R nearly copy the content of input; TSST has difficulty in generating fluent sentences. D-HST can transfer the style on the premise of basically preserving the content.

### 5.3 Human Evaluation

To assess the quality of text generated using D-HST from human perspective, we designed two human evaluations based on the performance in TechST dataset. First, we randomly sampled 20 groups head-

[5] we randomly sample 60 headlines from the baseline model and our model evenly, and ask the annotators to select the ones that transfer style and preserve content, and the bertscore of the selected headlines mostly falls between 0.63-071.

| | Example #1 | Example #2 | Example #3 |
|---|---|---|---|
| **Input** | IBM to Acquire Red Hat for $34 Billion | Microsoft Releases Azure Open AI Service Including Access to Powerful GPT-3 Models | EF Core Database Providers |
| **StyIns** | IBM to Acquire Red Hat for $34 Billion | AWS and Cloudflare Add Bot Management Features to Their Firewalls | A Core Database Providers |
| **BART+R** | IBM to Acquire Red Hat for $ 34 Billion | Microsoft Releases Azure Open AI Service with Powerful GPT-3 Models | EF Core Database Providers |
| **TSST** | IBM to Acquire Red Hat for $ 34 Billion | Microsoft Releases Azure Open AI Service Including Access to Powerful QR Models | Using Core Database Providers |
| **D-HST** | Why IBM Acquires Red Hat for $34M | Microsoft Azure: Accessing Open-Source Microsoft Machine Models | Going Into Core Database Providers |

Table 4: Example outputs generated by different models. Red parts represent stylistic attributes D-HST captures.

| | Example #1 | Example #2 |
|---|---|---|
| **Input** | The New Microsoft Edge - Microsoft Build 2020 | Qwik, a Resumable Javascript Framework |
| **Category** | Category2: VP NP | |
| | Introducing Microsoft's New Microsoft Edge | Using a Resumable JavaScript Framework |
| | Category95: QP VP NP | |
| | How to Build Microsoft's New Microsoft Edge | How to Develop a Resumable Javascript Framework |

Table 5: Examples of generated headlines given specific style category.

lines generated from baselines and D-HST, respectively. 10 postgraduates annotators were asked to score the candidates according to the following attributes from 0 to 2. *Fluency*: how fluent and readable the headline is? *Interestedness*: is the generated headline interesting? The final score of each model is calculated by averaging all judged scores. The results in Table 3 show that headlines generated by our proposed D-HST model receives most popularity compared to other models, indicated by the *Interestedness* metric. Additionally, both BART+R and D-HST generate fluent headlines.

The second human evaluation was designed to compute the click-through rate based on users' real click behavior. It is the most straightforward method of testifying **attractiveness**. When giving many headlines to real readers, we will examine which model receives the most clicks in this evaluation. Specifically, we selected 11 postgraduate annotators, each of whom was given a list of news headlines. The annotators were asked to click on those headlines that are most attractive to them. To make the selection as fair as possible, we carefully design to let the headlines generated by each model distribute evenly across the list, and the headline order was randomly shuffled to eliminate the effect of position on the probability of being clicked. Finally, each list contained 36 headlines (Each model generates 9 headlines, D-HST and three baselines models compared in this experiment) and the annotators were asked to click on 5 most attractive ones. As shown in Figure 2, the largest rate (reaches 58%) obtained by

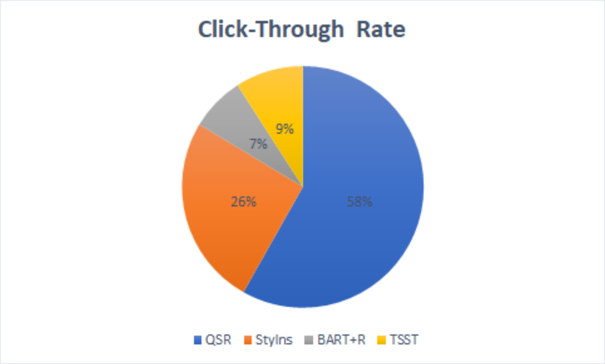

Figure 2: Human evaluation of click-through rate.

| Dataset | Strength | CP | STR | SCT | PPL |
|---|---|---|---|---|---|
| TechST | $\gamma$=0 | 0.669 | 0.817 | 0.368 | 14.97 |
|  | $\gamma$=0.1 | 0.668 | 0.824 | 0.369 | 15.17 |
|  | $\gamma$=0.3 | 0.665 | 0.844 | 0.372 | 15.48 |
|  | $\gamma$=0.5 | 0.661 | 0.857 | 0.373 | 15.81 |

Table 6: Evaluation of the style bias strength $\gamma$

the D-HST mainly conform to the previously quantitative results. We can conclude that D-HST generates the most appealing and acceptably fluent headlines.

### 5.4 Discrete Style Space and Controllability

To investigate whether style information is encoded in categories of discrete style space, we inspect to select two kinds of structures to control the generated headlines' styles in the inference stage. The outcomes are shown in Table 5. As it clearly demonstrates, category 2 and category 95 contain two distinct syntactic structures which are "VP NP" and "QP VP NP", respectively. Based on them, given the same input, our D-HST model is capable of generating different attractive headlines match the chosen structures. The results again indicate that the stylistic features are well disentangled and it is easy to control the style of generated results.

### 5.5 Style Bias Strength

As mentioned in Section 3.5, external knowledge $I_{test}$ is inserted as the style bias in the inference. The style category $K = 324$ in TechST dataset. To investigate how the style bias strength $\gamma$ affects the final generation, we chose different values on $\gamma$ and evaluate the performance in a series of automatic metrics, presented in Table 6. Through the experiment, we find that adding a style bias is effective for style transfer, and the scores of STR and SCT increase. The generation quality of the model has no significant fluctuation as the style strength increase, indicating that the model has strong generalization and is insensitive to the parameter.

## 6 Conclusion

This paper presents an unsupervised model for headline style transfer. It consists of content, style and their mixing components, which are together fed to decoder for headline generation. In particular, we propose to extract the style features in a discrete style space, and each discrete point corresponds to a particular category of the styles. Our system is comprehensively evaluated by both quantitative and qualitative metrics, and it produces cutting-edge outcomes in two typical datasets. Our work can be applied in the scenarios of formality machine translation, politeness transfer in intelligent customer service, spoken language transfer in live broadcast delivery. It can also be followed by the task of paraphrase and data augmentation.

## References

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. *arXiv preprint arXiv:1905.05621*.

Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. 2017. Stylenet: Generating attractive visual captions with styles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3137–3146.

Qipeng Guo, Zhijing Jin, Ziyu Wang, Xipeng Qiu, Weinan Zhang, Jun Zhu, Zheng Zhang, and Wipf David. 2021. Fork or fail: Cycle-consistent training with many-to-one mappings. In *International Conference on Artificial Intelligence and Statistics*, pages 1828–1836. PMLR.

Tom Hosking and Mirella Lapata. 2021. Factorising meaning and form for intent-preserving paraphrasing. *arXiv preprint arXiv:2105.15053*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International conference on machine learning*, pages 1587–1596. PMLR.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, Lisa Orii, and Peter Szolovits. 2020. Hooks in the headline: Learning to generate headlines with controlled styles. *arXiv preprint arXiv:2004.01980*.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep learning for text style transfer: A survey. *Computational Linguistics*, 48(1):155–205.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2018. Disentangled representation learning for non-parallel text style transfer. *arXiv preprint arXiv:1808.04339*.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Osama Khalid and Padmini Srinivasan. 2020. Style matters! investigating linguistic style in online communities. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 360–369.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2021. Thank you bart! rewarding pre-trained models improves formality style transfer. *arXiv preprint arXiv:2105.06947*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*.

Mingzhe Li, Xiuying Chen, Min Yang, Shen Gao, Dongyan Zhao, and Rui Yan. 2021. The style-content duality of attractiveness: Learning to write eye-catching headlines via disentanglement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13252–13260.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. *arXiv preprint arXiv:1905.13164*.

Dayiheng Liu, Jie Fu, Yidan Zhang, Chris Pal, and Jiancheng Lv. 2020. Revision in continuous space: Unsupervised text style transfer without adversarial learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8376–8383.

Teruko Mitamura and Eric Nyberg. 2001. Automatic rewriting for controlled language translation. In *The Sixth Natural Language Processing Pacific Rim Symposium (NLPRS2001) Post-Conference Workshop, Automatic Paraphrasing: Theories and Applications*.

Jonas Mueller, David Gifford, and Tommi Jaakkola. 2017. Sequence to better sequence: continuous revision of combinatorial structures. In *International Conference on Machine Learning*, pages 2536–2544. PMLR.

Sharmila Reddy Nangi, Niyati Chhaya, Sopan Khosla, Nikhil Kaushik, and Harshit Nyati. 2021. Counterfactuals to control latent disentangled text representations for style transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 40–48.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. *arXiv preprint arXiv:1803.06535*.

Machel Reid and Victor Zhong. 2021. Lewis: Levenshtein editing for unsupervised text style transfer. *arXiv preprint arXiv:2105.08206*.

Alexey Romanov, Anna Rumshisky, Anna Rogers, and David Donahue. 2018. Adversarial decomposition of text representation. *arXiv preprint arXiv:1808.09042*.

Aurko Roy, Ashish Vaswani, Arvind Neelakantan, and Niki Parmar. 2018. Theory and experiments on vector quantized autoencoders. *arXiv preprint arXiv:1805.11063*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. *Advances in neural information processing systems*, 30.

Shangquan Sun and Jian Zhu. Plug-and-play textual style transfer.

Martina Toshevska and Sonja Gievska. 2021. A review of text style transfer using deep learning. *IEEE Transactions on Artificial Intelligence*.

Minh Tran, Yipeng Zhang, and Mohammad Soleymani. 2020. Towards a friendly online community: An unsupervised style transfer framework for profanity redaction. *arXiv preprint arXiv:2011.00403*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Xiaoyuan Yi, Zhenghao Liu, Wenhao Li, and Maosong Sun. 2021. 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*, pages 3801–3807.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.