Stage-wise Stylistic Headline Generation: Style Generation and Summarized Content Insertion

Jiaao Zhan[1] , **Yang Gao**[1,2*] , **Yu Bai**[1] , **Qianhui Liu**[1]
[1]School of Computer Science and Technology,
Beijing Institute of Technology, Beijing, China
[2]Beijing Engineering Research Center of High Volume Language Information Processing
and Cloud Computing Applications, Beijing, China
{jiaao_zhan,gyang,yubai,qhliu98}@bit.edu.cn

Abstract

A quality headline with a high click-rate should not only summarize the content of an article, but also reflect a style that attracts users. Such demand has drawn rising attention to the task of stylistic headline generation (SHG). An intuitive method is to first generate plain headlines leveraged by document-headline parallel data then transfer them to a target style. However, this inevitably suffers from error propagation. Therefore, to unify the two sub-tasks and explicitly decompose style-relevant attributes and summarize content, we propose an end-to-end stage-wise SHG model containing the style generation component and the content insertion component, where the former generates stylistic-relevant intermediate outputs and the latter receives these outputs then inserts the summarized content. The intermediate outputs are observable, making the style generation easy to control. Our system is comprehensively evaluated by both quantitative and qualitative metrics, and it achieves state-of-the-art results in SHG over three different stylistic datasets.

1 Introduction

Although neural-based sequence-to-sequence (seq2seq) models are able to generate factual, concise, and fluent headlines [Chopra *et al.*, 2016], the headlines generated are austere and lack stylistic attributes to draw audiences into an article. Stylistic Headline Generation, or SHG for short, is a deep learning task proposed by Jin *et al.* [2020] that seeks to generate headlines with a specified style, such as humorous, romantic or clickbait. The theory of the model is very sound: headlines containing the same content as the plain headlines are generated with the desired stylistic attributes. Unfortunately, in practice, acquiring enough parallel data to train the model is almost impossible. Hence approaches to SHG that do not rely on large scale labeled training sets need to be explored [Sudhakar *et al.*, 2019; Dai *et al.*, 2019; Jin *et al.*, 2020; Krishna *et al.*, 2020].

SHG in an unsupervised setting, however, presents several challenges. The first is how to decompose stylistic patterns and summarized content during the generation process. The second is how to explicitly interpret the style generated with controlled style constraints. The third is how to ensure the content of the headline is consistent regardless of the style. Several researches have already attempted to answer some of these questions. For instance, latent space representations can be disentangled into semantic content and stylistic attributes [Zhao *et al.*, 2018; Shen *et al.*, 2017], but the disentanglement process can mean some of the semantic information get lost with long text strings, plus it is no longer possible to control the quality of the headlines being generated [Dai *et al.*, 2019]. By contrast, style-specific decoder approaches can generate output according to the target style while preserving the original content to a relatively high degree [Li *et al.*, 2020b; Jin *et al.*, 2020]. Delete-Retrieve-Generate (DRG) [Li *et al.*, 2018] and its extension Generative Style Transformer (GST) [Sudhakar *et al.*, 2019] alternatively decompose style factors by word-level editing actions, which allows for fine-grained control over style attributes.

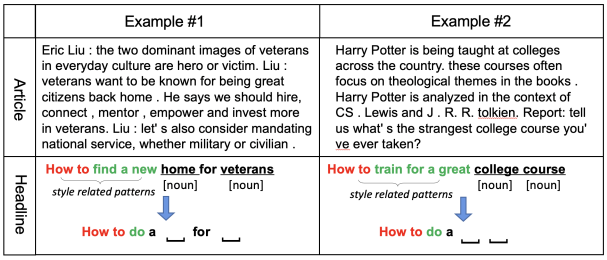

Figure 1: Two pairs of new articles and headlines demonstrating an inherent language pattern giving rise to a specific style.

Our approach does not directly build on any of these methods. Instead, we are inspired by the observation that certain styles of headlines share the same inherent linguistic patterns: the stem and syntax of the sentence is the same while some key words change to express the content. Figure 1 shows an example. Here, there are two article-headline example pairs. Although each of the articles is written from a different perspective, the headlines for each are written to evoke the informative style. Both of them follow the syntax 'How to [verb] [noun] __ [noun]', where the verbs and nouns represent the substantive content, i.e., find and home in Example #1, train and college course in Example #2. To simplify the syntax

* Corresponding Author

even more, verbs are doing words and could be replaced with the word 'do'. From this observation, we conjecture that the SHG task could be creatively framed as a stage-wise process: constructing the style first and then populating that style with substantive content. Hence, we propose an end-to-end stage-wise model for the SHG task consisting of a style generation component and a content insertion component. For brevity, we name this S-SHG, short for stage-wise SHG.

The style generation component first identifies inherent linguistic patterns and generates style attributes as an intermediate output with placeholders for the content to be inserted in the second component. Generating the intermediate outputs is done through supervised signals. As Figure 1 shows, distinguishing style from content strongly relates to the parts of speech (POS) in the target sentence. Hence, we design a masking strategy based on POS tagging to transform the target sentence into its intermediate supervised signal.

To implement these two component, we leverage two individual transformer decoders—a stylistic decoder and an insertion decoder. These decoders, along with a shared encoder, are then optimized via a seq2seq headline generation task and a style-specific sentence reconstruction task within a multi-task learning framework. The approach follows an end-to-end asynchronous training strategy.

This end-to-end stage-wise approach addresses all three of the challenges associated with unsupervised data raised earlier. That is, the model is able to **decompose** stylistic patterns and summarized content, making the outputs **interpretable**. Moreover, it assures accurate style control while **preserving the summarized content** as much as possible. The superiority of the framework lies in its ability to explicitly learn stylistic attributes, with the supervision of the constructed intermediate supervised signals.

We evaluate the approach using the same three SHG tasks described in Jin *et al.* [2020]. Each task is assessed in terms of style control, content preservation, attraction, diversity, and fluency. Both automatic and human evaluations show that our model outperforms the state-of-the-art methods by all indicators. Further, a manual analysis of the intermediate outputs demonstrates interpretability and shows that the model performs well at both style generation and content insertion.

The contributions of this paper can be summarized as follows: 1) We propose a novel model called S-SHG, which consists of a **style generation component** and a **content insertion component**. 2) We inject intermediate supervised signals into the model, ensuring interpretability and the ability to decompose the headline into its stylistic attributes and substantive contents. 3) Experiments on three SHG tasks show that each component in our approach performs well on its sub-task and clarifies interpretability.

2 Related Work

Text Style Transfer The aim of this field is to change the style attributes of a piece of text while preserving its content. However, like SHG, it also suffers from a lack of training data that parallels the target style. Additionally, the models offer little explainability, making it difficult to separate style from content. To address the first concern, Shen *et al.* [2017] train a cross-aligned auto-encoder, with shared content but a separate style distribution. Hu *et al.*[2017] apply an attribute classifier and a VAE framework to guide the generator to produce sentences with a desired style. Similarly, Zhao *et al.* [2018] and John *et al.* [2019] use a regularized auto-encoder within an adversarial training framework. However, these methods require extensive hyper-parameter tuning and are not particular good at preserving the content. Denoised auto-encoding, another approach for generating style information, works through a self-learning-based reconstruction process [Shen *et al.*, 2020].

Many approaches based on adversarial training have been proposed to solve the second problem, which is to find a disentangled latent space irrelevant of the style [Shen *et al.*, 2017; Zhao *et al.*, 2018]. However, it is difficult to assure the quality of disentanglement with these strategies. They also lack interpretability. Several groups used cycled reinforcement learning [Xu *et al.*, 2018] to strengthen style attributes and fuse stylistic information through a style-specific decoder [Li *et al.*, 2020a; Li *et al.*, 2020b; Jin *et al.*, 2020]. Others subsequently added a mapping function to modify the latent representations [Mai *et al.*, 2020; Shang *et al.*, 2019]. The unique point of our approach is that it avoids the limitation of phrase boundaries and offers interpretable results at our stage-wise process.

3 Our Method

This section begins with a brief introduction to SHG task. We then introduce the overall structure of our proposed S-SHG. Finally, we outline the data construction methods, the training scheme, and the inference process of the proposed model.

3.1 Problem Formulation

The dataset for SHG includes two parts [Jin *et al.*, 2020]. One is a parallel headline generation dataset $X = \{\boldsymbol{x}(i), \boldsymbol{y}(i)\}_{i=1}^{N}$ consisting of pairs of news article $\boldsymbol{x}$ and their plain headlines $\boldsymbol{y}$ which have no specific style. The other is a non-parallel dataset $T = \{\boldsymbol{t}(i)\}_{i=1}^{L}$ consisting of sentences $\boldsymbol{t}$ that do carry stylistic information in the desired style s (e.g., clickbait). Overall, the goal of the task is to generate a stylistic headline $\boldsymbol{y}_s$ with style s given a source article $\boldsymbol{x}$, where the generation can be formulated as $P_s(\boldsymbol{y}_s|\boldsymbol{x})$.

3.2 Model Details

A challenging goal with our model is to explicitly show which part of sentence represents the "style" and which part summarizes the original content as the plain headline. Motivated by the phenomenon illustrated in Figure 1, we design a stage-wise process that essentially generates a stylistic pattern and then fills in the blanks with meaningful content.

Figure 2 provides a simple schematic of the process. The model comprises a shared encoder, a stylistic decoder and an insertion decoder, all of which are based on the Transformer architecture [Vaswani *et al.*, 2017]. The source article $\boldsymbol{x}$ are encoded into $\boldsymbol{z}$ as the inputs to the decoders. The intermediate output $\boldsymbol{y}^{inter}$ (for y) or $\boldsymbol{y}_s^{inter}$ (for y_s) are generated first by the *Stylistic Decoder*. These intermediate output contains the stylistic attributes and masked content-relevant placeholders. Next, the *Insertion Decoder* fills in the summarized content words, to produce either a plain headline y or our desired stylistic headline y_s.

Shared Encoder The encoder is denoted as $Enc(.)$. It can be shared since it plays a similar role to map input article x to latent representation z when generating both plain headline y and stylistic headline y_s. The mapping process can be represented as $\boldsymbol{z} = Enc(\boldsymbol{x})$.

Stylistic Decoder To better decompose the content and the style of text, our stylistic decoders generate $\boldsymbol{y}^{inter}$ and $\boldsymbol{y}_s^{inter}$. Following Jin *et al.* [2020], the decoder's parameters are shared between Dec^{sty} and Dec_s^{sty} except for the normalization layer and the attention mechanism. The two sets of parameters are denoted as $\{\alpha, \alpha_s\}$ and $\{W^q, W_s^q\}$. Thus, the intermediate outputs y^{inter} and y_s^{inter} are acquired as follows:

$$\begin{aligned} \boldsymbol{y}^{inter} &= Dec^{sty}(\boldsymbol{z}, \alpha, W^q) \\ \boldsymbol{y}_s^{inter} &= Dec_s^{sty}(\boldsymbol{z}, \alpha_s, W_s^q) \end{aligned} \tag{1}$$

where $Dec^{sty}(.)$ and $Dec_s^{sty}(.)$ denote the stylistic decoders for generating the plain headline y and the stylistic headline y_s, respectively. Hence, our model is capable of generating observable intermediate outputs y_s^{inter} and y^{inter} that are explainable.

Insertion Decoder The insertion decoder fills in the content-relevant placeholders with the correct words, which shares a similar scheme with Conditional Mask Language Model (CMLM). This task needs the bi-directional information around the masked words, which is not appropriate for a common auto-regressive Transformer decoder, as it only attend to the information before the target word. Hence, we apply a non-autoregressive Transformer decoder to predict the masked words simultaneously based on bi-directional information in y^{inter} and y_s^{inter}. Meanwhile, it can also drastically reduce the inference time.

Because similar functionality of the insertion decoder is needed to generate both the plain headline y and the desired stylistic headline y_s, we use one shared insertion decoder for both tasks. The outputs are $\boldsymbol{y} = Dec^{ins}(\boldsymbol{z}, \boldsymbol{y}^{inter})$ and $\boldsymbol{y}_s = Dec^{ins}(\boldsymbol{z}, \boldsymbol{y}_s^{inter})$, where $Dec^{ins}(.)$ denotes the shared insertion decoder.

In summary, the stylistic decoder learns to generate the intermediate outputs containing the style attribute and content-relevant placeholders. The insertion decoder receives the intermediate outputs and inserts words that appropriately preserve the original content into the placeholders.

3.3 Data Preparation

To have the model generate the intermediate outputs y^{inter} and y_s^{inter}, we need supervision signals for the stylistic decoder. These signals, denoted as m, must contain both stylistic attributes and content-relevant placeholders, which can be acquired by masking certain content-relevant words from $y(i)$ and $t(i)$. To do this, the training corpus X is modified to $\hat{X} = \{\boldsymbol{x}(i), \boldsymbol{m}_y(i), \boldsymbol{y}(i)\}_{i=1}^N$ that is agnostic to styles and T is modified into a non-parallel dataset $\hat{T} = \{\boldsymbol{t}(i), \boldsymbol{m}_t(i)\}_{i=1}^L$ of a specific style. $\boldsymbol{m}_y$ represent the masked plain headlines of y, while $\boldsymbol{m}_t$ are the masked sentences of $\boldsymbol{t}$.

The main objective is to create intermediate supervised signals (i.e., $\boldsymbol{m}_y$ and $\boldsymbol{m}_t$) in both the headline generation dataset and non-parallel stylistic dataset. The core idea is to keep the subset of words reflecting stylistic attribute of a sentence and mask the other content-relevant words.

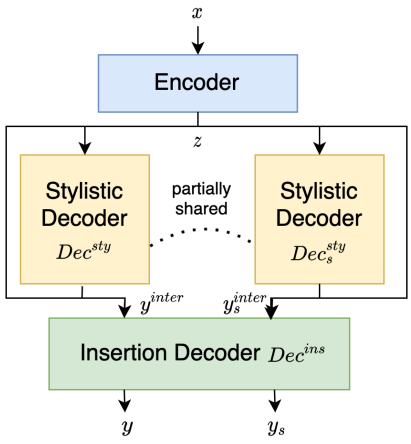

Figure 2: The architecture of S-SHG.

Extracting Style-Agnostic Content It is difficult to pinpoint the signals for a specific style. However, in a way, we can say that the content is the words that convey key information about meaning, irrespective of style. For the most part, these elements will be *nouns*, *conjunctions*, *pronouns*, and *orientation phrases*. For this task, we enlist the help of Stanford CoreNLP [1] to analyze the POS in the target sentences. We then mask these words with the placeholders $[\text{mask}]_{\text{noun}}$, $[\text{mask}]_{\text{conj}}$, $[\text{mask}]_{\text{pron}}$, $[\text{mask}]_{\text{orient}}$. Note that a uniform masking probability of 0.2 is applied empirically, which means that not all the content-relevant words are necessarily need to be masked. This is to make it easier for the stylistic decoder to acquire the basic ability to generate language. This masking strategy enables the model to not only generate more fluent sequences that preserve content but also to extract the stylistic attributes from an unsupervised stylistic corpus with the first component.

In Figure 3, $\boldsymbol{m}_t$ and $\boldsymbol{m}_y$ show an example of the intermediate supervised signals. The model first generates intermediate output given the supervised signals $\boldsymbol{m}_y$ and $\boldsymbol{m}_t$, e.g., "$[\text{mask}]_{\text{noun}}$ in northeast $[\text{mask}]_{\text{noun}}$ kills at least 29". This phrase includes both style-relevant structures and placeholders indicating that content words, such as words "Floods" and "China", should be placed in these positions.

3.4 Model Training

As Figure 3 shows, the end-to-end training process of S-SHG, is conducted on our two constructed datasets $\hat{X}$ and $\hat{T}$ based on the multi-task framework proposed in Jin *et al.* [2020], where a Denoised Auto-Encoding (DAE) task and a seq2seq task are conducted simultaneously to capture style information and summarize, respectively.

We take a further step to bring our S-SHG into this training process. In this way, the learning of these two tasks are further decomposed by the stylistic generation component and the content insertion component.

[1] https://stanfordnlp.github.io/CoreNLP/

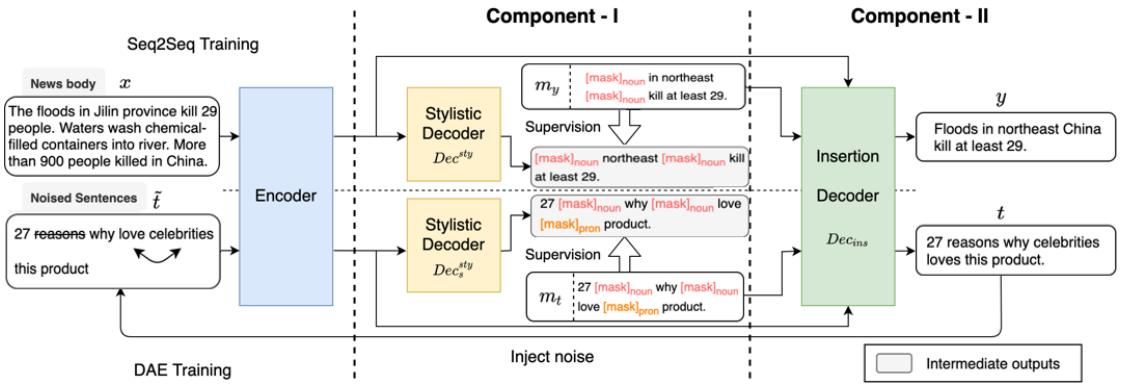

Figure 3: The training scheme of our proposed stage-wise model for stylistic headline generation. The gray panel represents the intermediate output produced by the stylistic decoder.

Incorporating the intermediate supervised signals m_y and m_t, we modify the original objectives $P(\boldsymbol{y}|\boldsymbol{x})$ and $P(\boldsymbol{t}|\tilde{\boldsymbol{t}})$ into the following form:

$$P(\boldsymbol{y}|\boldsymbol{x}) = P_{ins}(\boldsymbol{y}|\boldsymbol{m}_y, \boldsymbol{x})P_{sty}(\boldsymbol{m}_y|\boldsymbol{x}) \tag{2}$$

$$P(\boldsymbol{t}|\tilde{\boldsymbol{t}}) = P_{ins}(\boldsymbol{t}|\boldsymbol{m}_t, \tilde{\boldsymbol{t}})P_{sty}(\boldsymbol{m}_t|\tilde{\boldsymbol{t}}) \tag{3}$$

where Eq. (2) shows the objective for the supervised headline generation task (i.e., $Task_y$), and Eq. (3) is for the unsupervised DAE training (i.e. $Task_t$). The term $\tilde{t}$ is converted by injecting noise into the sentence t. P_{sty} and P_{ins} represent the distributions from the stylistic and insertion decoder respectively.

Given this decomposition, S-SHG captures stylistic information with the first component, then learns summarized content with the second component. Details are shown as follows:

Component - I: Style Generation

For the training of the style generation component (w.r.t. stylistic decoder), we have two intermediate signals m_y and m_t as supervision, minimizing the cross entropy loss as follows:

$$\begin{aligned} \mathcal{L}_{sty} &= \mathcal{L}_{sty}^{Task_y} + \mathcal{L}_{sty}^{Task_t} \\ \mathcal{L}_{sty}^{Task_y} &= \mathbb{E}_{(\boldsymbol{x},\boldsymbol{m_y})\sim\hat{X}}[-\log p(\boldsymbol{m_y}|\boldsymbol{x})] \\ \mathcal{L}_{sty}^{Task_t} &= \mathbb{E}_{(\tilde{\boldsymbol{t}},\boldsymbol{m_t})\sim\hat{T}}[-\log p(\boldsymbol{m_t}|\tilde{\boldsymbol{t}})] \end{aligned} \tag{4}$$

Component - II: Content Insertion

For the training of the content insertion component (w.r.t., the insertion decoder), we use the original target y and t as supervision. As the task of this component is equal to a conditional mask language model (CMLM) task, we optimize it with cross entropy loss over the masked tokens in m_y and m_t:

$$\begin{aligned} \mathcal{L}_{ins} &= \mathcal{L}_{ins}^{Task_y} + \mathcal{L}_{ins}^{Task_t} \\ \mathcal{L}_{ins}^{Task_y} &= \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{m_y})\sim\hat{X}}[-\log p(\boldsymbol{y}|\boldsymbol{m_y}, \boldsymbol{x})] \\ \mathcal{L}_{ins}^{Task_t} &= \mathbb{E}_{(\tilde{\boldsymbol{t}},\boldsymbol{t},\boldsymbol{m_t})\sim\hat{T}}[-\log p(\boldsymbol{t}|\boldsymbol{m_t}, \tilde{\boldsymbol{t}})] \end{aligned} \tag{5}$$

The whole model is trained by jointly minimizing the style generation component training loss $\mathcal{L}_{sty}$ and the content insertion component training loss $\mathcal{L}_{ins}$ via the proposed stage-wise framework. The total loss becomes:

$$\mathcal{L} = \lambda\mathcal{L}_{sty} + (1-\lambda)\mathcal{L}_{ins} \tag{6}$$

where λ is a trade-off parameter, which is set as 0.5 in experiments.

3.5 Inference

Given a news article x, the stylistic headline $\boldsymbol{y}_s$ is generated by the following inference process:

$$\boldsymbol{y}_s = Dec^{ins}(Dec_s^{sty}(Enc(\boldsymbol{x})), Enc(\boldsymbol{x})) \tag{7}$$

Meanwhile, due to the intrinsic structure of our S-SHG model , we can also acquire a plain headline without any specific styles:

$$\boldsymbol{y} = Dec^{ins}(Dec^{sty}(Enc(\boldsymbol{x})), Enc(\boldsymbol{x})) \tag{8}$$

4 Experiments and Results

Experimental Settings We use the datasets published by Jin *et al.* [2020], which consist of a parallel dataset CNN-NYT, and three non-parallel stylistic corpora related to clickbait, humor, and romance.

The CNN-NYT dataset comprises article-headline pairs from the New York Times (NYT) and the CNN datasets. These data are cleaned by filtering out sentences of less than four tokens, leaving 134,443 training samples, 3,000 validation samples, and 3,000 test samples. Without article-headline pairs, the three stylistic corpora are made up of style-specific sentences. Specifically, the humorous and romantic corpora are collected from the novels in the corresponding genres of BookCorpus [Zhu *et al.*, 2015]. The clickbait corpus is assembled from the Examiner SpamClickBait News dataset[2]. Each consists of $500,000$ sentences.

Baselines We compare S-SHG against the following four well-known baselines:

MASS-Stylistic [Jin *et al.*, 2020], initialized with MASS model [Song *et al.*, 2019] and then sequentially fine-tuned on the CNN-NYT dataset via seq2seq training followed by DAE training on the stylistic dataset.

Generative Style Transformer (GST) [Sudhakar *et al.*, 2019], which replaces the style words of the source sentence

[2]https://www.kaggle.com/therohk/examine-the-examiner

Styles	Model	BLEU(↑)	ROUGE-1(↑)	ROUGE-2(↑)	ROUGE-L(↑)	ACC-BERT(↑)	ACC-fastText(↑)	D-1(↑)	D-2(↑)	D-3(↑)	PPL(↓)
Humor	MASS-Stylistic	2.87	19.04	4.82	16.51	50.83	51.03	0.14	0.65	0.91	40.04
	GST	6.90	17.91	4.51	15.92	57.18	50.23	0.21	0.78	0.93	68.86
	ST	3.91	20.34	5.51	18.52	53.44	56.18	0.19	0.76	0.94	77.07
	TitleStylist	8.83	26.81	9.12	23.92	50.83	51.03	0.22	0.71	0.89	25.95
	S-SHG	**8.92**	**27.26**	**9.34**	**24.57**	**64.32**	**65.87**	0.20	0.72	0.84	**25.35**
	TitleStylist-F	10.33	27.71	9.82	25.04	NA	NA	0.23	0.72	0.89	**25.24**
	S-SHG-F	**10.87**	**28.23**	**10.54**	**25.42**	NA	NA	0.22	0.73	0.91	25.50
Romance	MASS-Stylistic	2.67	18.01	4.42	15.61	53.22	52.97	0.13	0.63	0.90	42.15
	GST	4.02	22.32	7.31	20.34	50.08	50.46	0.20	0.73	0.90	62.75
	ST	3.80	20.61	5.80	18.72	54.93	54.32	0.19	0.75	0.94	83.78
	TitleStylist	8.64	26.31	8.92	23.30	51.16	51.33	0.21	0.70	0.89	24.71
	S-SHG	**8.78**	**26.82**	**9.23**	**23.74**	**62.15**	**69.93**	0.18	0.71	0.88	**24.46**
	TitleStylist-F	10.11	27.12	9.61	24.40	NA	NA	0.22	0.70	0.88	**23.20**
	S-SHG-F	**10.84**	**29.13**	**10.34**	**25.92**	NA	NA	0.21	0.73	0.90	24.67
Clickbait	MASS-Stylistic	1.62	16.21	3.50	13.42	53.98	52.08	0.17	0.69	0.93	83.15
	GST	5.30	22.90	6.51	20.72	58.27	56.68	0.21	0.76	0.93	45.17
	ST	4.50	23.52	7.13	23.20	55.23	57.18	0.20	0.76	0.92	93.97
	TitleStylist	8.82	27.21	9.20	24.52	56.03	59.07	0.26	0.79	0.94	32.02
	S-SHG	**8.94**	**27.73**	**9.31**	**24.74**	**69.43**	**63.85**	0.23	0.78	0.95	**31.82**
	TitleStylist-F	10.32	27.82	9.70	25.03	NA	NA	0.23	0.73	0.90	25.88
	S-SHG-F	**11.45**	**28.43**	**10.74**	**25.72**	NA	NA	0.21	0.74	0.91	**25.34**

Table 1: The automatic evaluation for all metrics: style control (ACC), content preservation (ROUGE-N/BLEU), global diversity (Distinct-N), and language perplexity (PPL). S-SHG and TitleStylist both generate two outputs during the inference phase: style-specific headlines and factual headlines. These factual alternative headlines are denoted with a "-F". Note that the factual headlines do not involve with stylistic evaluation, so the corresponding evaluations for these elements are labeled "NA".

with the other style words retrieved from target sentences. Because GST lacks a headline generation module, we fine-tuned MASS on the CNN-NYT dataset and use GST to translate its generated plain headlines into the target style.

Style Transformer (ST) [Dai *et al.*, 2019], a text style transformer that incorporates style information through a reconstruction task. The stylistic headline generation is accomplished by using the same way as GST.

TitleStylist [Jin *et al.*, 2020], the seminal model, trained on supervised headline generation and unsupervised DAE.

Note that all the models are trained on the processed datasets mentioned above.

4.1 Quantitative Results

Style Control Style transfer accuracy is often used to evaluate how well an approach controls style. The idea is to apply a high-performance style classifier to evaluate the probability that a sentence adheres to a particular style. The accuracy metric (ACC) is therefore defined as the percentage of the generated sentences assigned to be the target style by the classifier. To evaluate the accuracy, we use two types of classifiers based on BERT [Dai *et al.*, 2019] and fastText [Joulin *et al.*, 2017]. Both classifiers are trained on the combined plain headline/stylistic training sets, and both achieve good performance on the test sets of all the stylistic datasets[3]. The two ACC columns in Table 1 show the accuracy for all the models. Using the fastText classifier, S-SHG surpasses the rest of the baselines, by between 10.0% and 36.2%. Similar trends are reflected by ACC-BERT metric.

Analysis on Global Diversity Although S-SHG returned better ACC metrics, we want to explore the possibility that this model was simply generating repeat, unique templates. Hence, to measure whether the model is learning diverse stylistic attributes, we implement a diversity metric. Diversity is an important criterion in text generation. Here, we define **global diversity** as the overall diversity of the model outputs, measured by automatic calculation following the Distinct-N metric [Li *et al.*, 2016] across the entire set of outputs. The results are reported in Table 1. S-SHG has a competitive Distinct-N score, which means that it is not generating repeat text templates. Additionally, we also define **local diversity** by the dynamism and richness of the content of a headline sentence. More details on this are provided in Section 4.2.

Content Preservation We use the standard metrics BLEU [Papineni *et al.*, 2002] and ROUGE [Lin, 2004] to measure how well the models expressed the given content. Higher scores mean better preservation. The leftmost block of Table 1 shows the results for all models. Unfortunately, we fail to reproduce the good results for TitleStylist reported in the original paper [Jin *et al.*, 2020], using their open-source code[4]. We report our results as they stand. The results demonstrate that S-SHG preserves more content information than the other baseline models while generating both style-specific headlines and factual headlines.

Language Perplexity We use perplexity (PPL) to measure the fluency of the generated sentences. A lower perplexity indicates that generated sentences are more fluent. We fine-tune OpenAI-GPT [Radford *et al.*, 2018] with our training headlines[5] to calculate the PPL of the generated headlines. The results in Table 1 show that S-SHG is comparable with TitleStylist and surpasses the other baseline models by a significant margin.

4.2 Human Evaluation

To more comprehensively assess our model, we also conduct human evaluations in addition to quantitative calculations. To this end, we randomly sample 20 news abstracts from each test set, and ask ten judges to rate the outputs generated by each

[3]BERT-based classifier achieves 99.6%, 99.5%, and 96.1% while fastText attains 99.6%, 94.2%, and 99.8%, respectively.

[4]https://github.com/jind11/TitleStylist

[5]PPL on the development set is 34.049.

	Example #1	Example #2	Example #3
Abstract	Vessel capsizes in Bahamas; 30 reported dead. U.S. coast guard-joins rescue effort and drops food, rafts.	The cease - fire comes on the 8th day of violence between Gaza and Israel. CNN has multiple crews around the region, bringing you the latest information.	Aaron Hernandez expected to go on trial in 2015. He has pleaded not guilty to three first - degree murder charges. Some of his closest associates are also facing charges.
Plain Headline	30 dead, dozens rescued after haitians' boat capsizes off Bahamas.	Latest key developments in the Gaza - Israel conflict.	Aaron Hernandez case : who 's who.
MASS-Stylistic	Vessel capsizes Bahamas injures 30 reported dead and U.S. coast guard joins rescue effort drops food rafts.	The latest cease fire on the 8th day of violence between Gaza and Israel has multiple crews around you.	Aaron Hernandez expected to go on trial 2015 as he has pleaded not guilty to three first degree murder charges.
GST	30 dead dozens rescued Haitians vs boat capsizes off Bahamas.	Developments in the Gaza.	Aaron Hernandez case who forecast weekend who.
ST	Cruise ship capsizes off Bahamas your at least 30 dead.	How to make a cease free fire scotty Gaza to Gaza.	What one Republic's next for Aaron Hernandez.
Title Stylist	30 dead after boat capsizes in Bahamas.	What you need to know about the Gaza cease - fire.	Aaron Hernandez pleads not guilty to murder.
S-SHG	At least 30 dead after boat capsizes off Bahamas. y_s^{inter}: At least 30 dead $[mask]_{conj}$ boat capsizes off $[mask]_{noun}$.	How to make a cease-fire the Gaza conflict. y_s^{inter}: How to make a $[mask]_{noun}$ - fire the Gaza $[mask]_{noun}$.	Whats next for Aaron Hernandez. y_s^{inter}: Whats next for $[mask]_{noun}$ $[mask]_{noun}$.

Table 2: The clickbaity headlines generated by different models. Blue parts represent the unreasonable phrase. Red parts represent the stylistic attributes that S-SHG captures. y_s^{inter} represents the intermediate outputs.

Styles	Models	Relevance	Attraction	Fluency	Diversity
Humor	MASS-Stylistic∗	0.10	-0.13	-0.01	-0.10
	GST∗	0.10	-0.38	-0.30	0.15
	ST∗	-0.49	-0.08	-0.40	-0.23
	TitleStylist∗	0.06	0.18	0.20	0.02
	S-SHG	**0.23**	**0.41**	**0.51**	**0.16**
Romance	MASS-Stylistic∗	**0.27**	-0.22	0.06	-0.08
	GST∗	-0.09	-0.20	-0.32	-0.04
	ST∗	-0.41	-0.12	-0.30	-0.07
	TitleStylist∗	-0.01	0.20	0.09	-0.04
	S-SHG	0.24	**0.34**	**0.47**	**0.23**
Clickbait	MASS-Stylistic	**0.28**	-0.25	-0.02	0.23
	GST ∗	-0.48	-0.11	-0.32	-0.21
	ST∗	-0.12	-0.13	-0.14	-0.3
	TitleStylist∗	0.10	0.01	0.17	-0.31
	S-SHG	0.22	**0.53**	**0.31**	**0.58**

Table 3: Human evaluation. Models with $*$ are significantly different from S-SHG (using a pairwise t-test; $p < 0.001$).

of MASS-Stylistic, GST, ST, and TitleStylist, plus our S-SHG. The judges make their assessments against the following four criteria: 1) *Relevance*—how semantically relevant the headline is to the news article. 2) *Attractiveness*—how appealing they feel the headline is. 3) *Fluency*—how comprehensive and easy-to-read the headline is. 4) *Diversity* —using the principles of the local diversity, discussed in Section 4.1. Concretely, a diverse headline is a single non-trivial sentence at the lexical level [Zhu *et al.*, 2018]. Following Liu and Lapata [2019], we use the Best-Worst Scaling method [Kiritchenko and Mohammad, 2017], asking the judges to choose the best examples of each criteria from the entire set of outputs (randomly mixed so judges do not know which model produce which headline). The final score of each model is calculated as the percentage of the times it is chosen as best minus the percentage of times it is chosen as the worst. Hence, final scores, shown in Table 3, range from -1 (worst) to 1 (best).

The results of this evaluation mainly conform to the quantitative results, from which we can conclude that S-SHG generate the most reflective, appealing, fluent, and diverse headlines. Interestingly, the MASS-Stylistic model receives slightly more votes for relevance for the Romance and Clickbait datasets than the proposed S-SHG. However, we can clearly see that sentences generated by MASS-Stylistic are too redundant to be a headline, through the examples included in Table 2. Apart from this, our method still has the second best Relevance score. This and the automatic evaluation score metric prove the effectiveness of our model.

4.3 Qualitative Results and Findings

To further investigate the style factors and performance of S-SHG, we sample several headlines generated by the models from the "clickbait" dataset, as shown in Table 2, and inspect the results manually. We draw several valuable conclusions that the community might benefit from.

Interpretable Generation As mentioned, the outputs generated by MASS-Stylistic are too long to be regarded as headlines and too incomprehensible to be of any use. The results of the GST are deemed the least fluent while the ST repeats its outputs. By contrast, S-SHG generates semantically appropriate, largely grammatically correct headlines that are understandable and appealing. Further, S-SHG produces intermediate outputs, which can be examined to reveal the elements of the headline relevant to style vs content. Take the rightmost column in Table 2 as example, 'Whats next for' are the stylistic attributes, while the content placeholders '$[mask]_{conj}$' and '$[mask]_{noun}$' have been generated properly and inserted as 'after' and 'Bahamas'. This makes the process of SHG interpretable and understandable.

5 Conclusion

The paper presents an end-to-end stage-wise SHG model containing the style generation component and the content insertion component, where the former generates stylistic-relevant intermediate outputs and the latter inserts the summarized content. Experimental results on three stylistic datasets have shown that our stage-wise model is fully interpretable and capable of generating high-quality stylistic headlines.

Acknowledgements

We appreciate the helpful discussions with Boxing Chen. We also thank all the anonymous reviewers for their insightful suggestions.

References

[Chopra *et al.*, 2016] Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, 2016.

[Dai *et al.*, 2019] Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. Style transformer: Unpaired text style transfer without disentangled latent representation. In *ACL*, 2019.

[Hu *et al.*, 2017] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *ICML*, 2017.

[Jin *et al.*, 2020] Di Jin, Zhijing Jin, Joey Tianyi Zhou, Lisa Orii, and Peter Szolovits. Hooks in the headline: Learning to generate headlines with controlled styles. In *ACL*, 2020.

[John *et al.*, 2019] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for non-parallel text style transfer. In *ACL*, 2019.

[Joulin *et al.*, 2017] Armand Joulin, Edouard Grave, and Piotr Bojanowski Tomas Mikolov. Bag of tricks for efficient text classification. *EACL 2017*, page 427, 2017.

[Kiritchenko and Mohammad, 2017] Svetlana Kiritchenko and Saif M. Mohammad. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. *ArXiv*, abs/1712.01765, 2017.

[Krishna *et al.*, 2020] Kalpesh Krishna, John Wieting, and Mohit Iyyer. Reformulating unsupervised style transfer as paraphrase generation. *ArXiv*, abs/2010.05700, 2020.

[Li *et al.*, 2016] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B. Dolan. A diversity-promoting objective function for neural conversation models. In *NAACL*, 2016.

[Li *et al.*, 2018] Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *NAACL*, 2018.

[Li *et al.*, 2020a] Xiao Li, Guanyi Chen, Chenghua Lin, and Ruizhe Li. Dgst: a dual-generator network for text style transfer. *arXiv e-prints*, pages arXiv–2010, 2020.

[Li *et al.*, 2020b] Xiao Li, Guanyi Chen, Chenghua Lin, and Ruizhe Li. A dual-generator network for text style transfer applications. *ArXiv*, abs/2010.14557, 2020.

[Lin, 2004] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[Liu and Lapata, 2019] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. *ArXiv*, abs/1908.08345, 2019.

[Mai *et al.*, 2020] Florian Mai, Nikolaos Pappas, Ivan Montero, Noah A. Smith, James Henderson Idiap Research Institute, Epfl, University of Washington, and Allen Institute for Artificial Intelligence. Plug and play autoencoders for conditional text generation. In *EMNLP*, 2020.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *Technical report, OpenAI*, 2018.

[Shang *et al.*, 2019] Mingyue Shang, Piji Li, Zhenxin Fu, Lidong Bing, Dongyan Zhao, Shuming Shi, and Rui Yan. Semi-supervised text style transfer: Cross projection in latent space. In *EMNLP*, 2019.

[Shen *et al.*, 2017] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S Jaakkola. Style transfer from non-parallel text by cross-alignment. In *NIPS*, 2017.

[Shen *et al.*, 2020] Tianxiao Shen, Jonas Mueller, Regina Barzilay, and Tommi Jaakkola. Educating text autoencoders: Latent representation guidance via denoising. In *International Conference on Machine Learning*, pages 8719–8729. PMLR, 2020.

[Song *et al.*, 2019] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936. PMLR, 2019.

[Sudhakar *et al.*, 2019] Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. "transforming" delete, retrieve, generate approach for controlled text style transfer. In *EMNLP*, 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[Xu *et al.*, 2018] Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *ACL (1)*, 2018.

[Zhao *et al.*, 2018] Junbo Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M Rush, and Yann LeCun. Adversarially regularized autoencoders. In *ICML*, 2018.

[Zhu *et al.*, 2015] Yukun Zhu, Ryan Kiros, Richard S Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, 2015.

[Zhu *et al.*, 2018] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. *arXiv preprint arXiv:1802.01886*, 2018.