The Style-Content Duality of Attractiveness: Learning to Write Eye-Catching Headlines via Disentanglement

Mingzhe Li [1, 2,*], Xiuying Chen [1, 2,*], Min Yang [3], Shen Gao [1], Dongyan Zhao [1, 2], Rui Yan [4,5, †]

[1] Wangxuan Institute of Computer Technology, Peking University,Beijing,China
[2] Center for Data Science, AAIS, Peking University,Beijing,China
[3] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[4] Gaoling School of Artificial Intelligence, Renmin University of China
[5] Beijing Academy of Artificial Intelligence
{li_mingzhe,xy-chen,shengao,zhaody}@pku.edu.cn, min.yang@siat.ac.cn, ruiyan@ruc.edu.cn

Abstract

Eye-catching headlines function as the first device to trigger more clicks, bringing reciprocal effect between producers and viewers. Producers can obtain more traffic and profits, and readers can have access to outstanding articles. When generating attractive headlines, it is important to not only capture the attractive *content* but also follow an eye-catching written *style*. In this paper, we propose a Disentanglement-based Attractive Headline Generator (DAHG) that generates headline which captures the attractive content following the attractive style. Concretely, we first devise a disentanglement module to divide the style and content of an attractive prototype headline into latent spaces, with two auxiliary constraints to ensure the two spaces are indeed disentangled. The latent content information is then used to further polish the document representation and help capture the salient part. Finally, the generator takes the polished document as input to generate headline under the guidance of the attractive style. Extensive experiments on the public Kuaibao dataset show that DAHG achieves state-of-the-art performance. Human evaluation also demonstrates that DAHG triggers 22% more clicks than existing models.

Introduction

With the rapid growth of information spreading throughout the Internet, readers get drown in the sea of documents, and will only pay attention to those articles with attractive headlines that can catch their eyes at first sight. On one hand, generating headlines that can trigger high click-rate is especially important for different avenues and forms of media to compete for user's limited attention. On the other hand, only with the help of a good headline, can the outstanding article be discovered by readers.

To generate better headlines, we first analyze what makes the headlines attractive. By surveying hundreds of headlines of popular websites, we found that one important feature that influences the attractiveness of a headline is its ***content***. For example, when reporting the same event, the headline "Happy but not knowing danger: Children in India play on the poisonous foam beach" wins over 1000 page views, while the headline "Chennai beach was covered with white foam for four days in India" only has 387 readers. The popular headline highlights the fact that "the beach is poisonous and affects children", which will concern more people than "white foam". On the other hand, the ***style*** of the headline also has a huge impact on attractiveness. For example, the headline "Only two people scored thousand in the history of NBA Finals" attracts fewer people than the headline "How hard is it to get 1000 points in the NBA finals? Only two people in history!", due to its conversational style that makes readers feel the need to see the answer to this question.

Most of the recent researches regard the headline generation task merely as a typical summarization task (Shu et al. 2018). This is not sufficient because a good headline should not only capture the most relevant content of an article but also be attractive to the reader. However, attractive headline generation tasks were paid less attention by researchers. Xu et al. (2019) tackle this task by adversarial training, using an attractiveness score module to guide the summarization process. Jin et al. (2020) introduce a parameter sharing scheme to disentangle the attractive style from the attractive text. However, previous works neglect the fact that attractiveness is not just about *style*, but also about *content*.

Based on the above analysis, we propose a model named *Disentanglement-based Attractive Headline Generation* (DAHG), which learns to write attractive headlines from both style and content perspectives. These two attractiveness attributes are learned from an attractive prototype headline, *i.e.,* the headline of the document in the training dataset that is most similar to the input document. First, DAHG separates the attractive style and content of the prototype headline into latent spaces, with two auxiliary constraints to ensure the two spaces are indeed disentangled. Second, the learned attractive content space is utilized to iteratively polish the input document, emphasizing the parts in the document that are attractive. Finally, the decoder generates an attractive headline from the polished input document representation under the

[*] Equal contribution.

[†] Corresponding author: Rui Yan (ruiyan@ruc.edu.cn)

guidance of the separated attractive style space. Extensive experiments on the public Kuaibao dataset show that DAHG outperforms the summarization and headline generation baselines in terms of ROUGE metrics, BLEU metrics, and human evaluations by a large margin. Specifically, DAHG triggers 22% more clicks than the strongest baseline.

The major contributions of this paper are as follows: (1) We devise a disentanglement mechanism to divide the attractive content and style space from the attractive prototype headline. (2) We propose to generate an attractive headline with the help of disentangled content space under the style guidance. (3) Experimental results demonstrate that our model outperforms other baselines in terms of both automatic and human evaluations.

Related Work

Our research builds on previous works in three fields: text summarization, headline generation, and disentanglement.

Text Summarization. Headline generation is a task based on text summarization, where methods can be divided into two categories, extractive, and abstractive methods. Extractive models (Nallapati, Zhai, and Zhou 2017; Zhou et al. 2018; Zhang et al. 2018) directly select sentences from article as the summary. In contrast, abstractive models (Gao et al. 2019b; Chen et al. 2019b; Gao et al. 2020a,b; Li et al. 2020) generate a summary from scratch. A series of work relies on prototype text to assist summarization. Cao et al. (2018) chose the template most similar to the input as a soft template to generate summaries. Following this, Gao et al. (2019a) proposed to generate the summary with the pattern based on prototype editing. Our work differs from previous works in focusing on the attractiveness of the generated summary.

Headline Generation. In recent years, text generation has made impressive progress (Li et al. 2019; Chan et al. 2019; Liu et al. 2020; Xie et al. 2020; Chan et al. 2020; Chen et al. 2021), and headline generation has become a research hotspot in Natural Language Processing. Most existing headline generation works solely focus on summarizing the document. Tan, Wan, and Xiao (2017) exploited generating headlines with hierarchical attention. Gavrilov, Kalaidin, and Malykh (2019) applied recent universal Transformer (Dehghani et al. 2018) architecture for headline generation. Attractive headline generation was paid less attention by researchers. Xu et al. (2019) trained a sensation scorer to judge whether a headline is attractive and then used the scorer to guide the headline generation by reinforcement learning. Jin et al. (2020) introduced a parameter sharing scheme to further extract attractive style from the text. However, to our best knowledge, no existing work considers the style-content duality of attractiveness.

Disentanglement. Disentangling neural networks' latent space has been explored in the computer vision domain, and researchers have successfully disentangled the features (such as rotation and color) of images (Chen et al. 2016; Higgins et al. 2017). Compared to the computer vision field, NLP tasks mainly focus on invariant representation learning. Disentangled representation learning is widely adopted in non-parallel text style transfer. For example, Fu et al. (2018) proposed an approach to train style-specific embedding. Some work also focused on disentangling syntax and semantic representations in text. Iyyer et al. (2018) proposed syntactically controlled paraphrase networks to produce a paraphrase of the sentence with the desired syntax given a sentence and a target syntactic form. Chen et al. (2019a) proposed a generative model to get better syntax and semantics representations by training with multiple losses that exploit aligned paraphrastic sentences and word-order information.

Existing works concentrate on learning the disentangled representation, and we take one step further to utilize this representation to generate attractive headlines.

Problem Formulation

Before presenting our approach for the attractive headline generation, we first introduce our notations and key concepts. For an input document $X^d = \{x_1^d, x_2^d, \dots, x_{m^d}^d\}$ which has m^d words, we assume there is a corresponding headline $Y^d = \{y_1^d, y_2^d, \dots, y_{n^d}^d\}$ which has n^d words. In our setting, the most similar document-headline pair to the current document-headline pair is retrieved from the training corpus as prototype pair. Retrieve details are introduced in §. The prototype document is defined as $X^r = \{x_1^r, x_2^r, \dots, x_{m^r}^r\}$, and prototype headline is $Y^r = \{y_1^r, y_2^r, \dots, y_{n^r}^r\}$.

For a given document X^d, our model first divides the latent representation of the retrieval attractive prototype headline Y^r into two parts: the attractive style space s, and attractive content space c. Then we use the content c to extract and polish document X^d and use the style s to guide attractive headline generation. The goal is to generate a headline $\hat{Y}^d$ that not only covers the salient part of input document X^d but also follows an attractive style.

Model

Overview

In this section, we propose our Disentanglement-based Attractive Headline Generation (DAHG) model, which can be divided into three parts as shown in Figure 1:

• **Disentanglement** contains (1) *Feature Extractor* module, which projects the prototype headline representation into latent style and content space, (2) *Style Space Constraint*, and (3) *Content Space Constraint* module, which ensure that the extracted style and content space does not mix with each other.

• **Highlight Polish** highlights the attractive part of the document guided by the attractive content space learned from the prototype headline.

• **Headline Generator** generates an attractive headline based on the polished document under the guidance of the attractive style.

Disentanglement

The disentanglement framework shown in Figure 2 consists of three components. The feature extractor serves as an autoencoder, with two constraints to facilitate disentanglement.

To begin with, we map a one-hot representation of each word in input document X^d and prototype document X^r into a high-dimensional vector space and employ a bi-directional

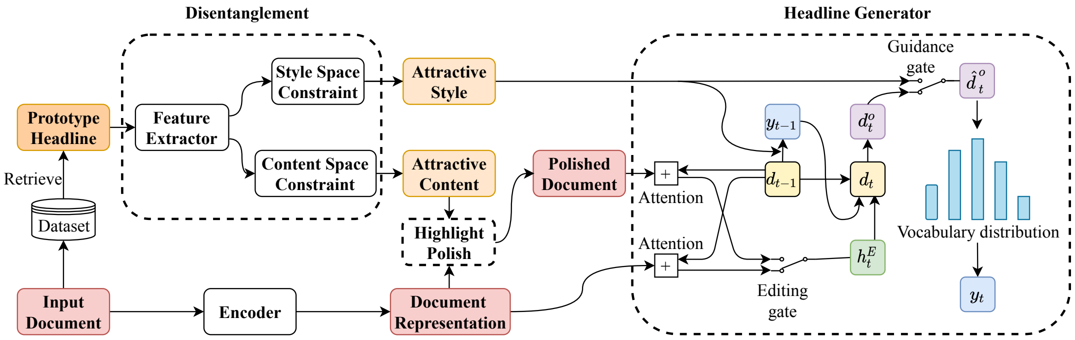


Figure 1: Overview of DAHG. We divide our model into three parts: (1) *Disentanglement* disentangles attractive style and attractive content space from prototype headline representation; (2) *Highlight Polish* highlights the attractive part in the input document with the help of attractive content space; (3) *Headline Generator* generates headline taking the polished document as input under the guidance of attractive style.

recurrent neural network (Bi-RNN$_\text{X}$) to model the temporal interactions between words. Then the encoder states of X^d and X^r is represented as $h_t^{X^d}$ and $h_t^{X^r}$, respectively. Following See, Liu, and Manning (2017), we choose Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) as the cell for Bi-RNN. We concatenate the last hidden state of the forward RNN $h_{m^r}^{X^r}$ and backward RNN $h_1^{X^r}$ and obtain the overall representation vector h^{X^r} for X^r.

Feature Extractor. We employ Variational AutoEncoder (VAE) as feature extractor, since VAE is appealing in explicitly modeling global properties such as syntactic, semantic, and discourse coherence (Li, Luong, and Jurafsky 2015; Yu et al. 2020). Concretely, feature extractor consists of two encoders and a decoder. The two feature encoders map the input prototype headline Y^r to an attractive content space c and an attractive style space s, and the decoder reconstructs the prototype headline $\hat{Y}^r$ from c and s.

Concretely, the two encoders compute two posterior distributions $q_\theta(c|Y^r)$ and $q_\theta(s|Y^r)$ given the prototype headline Y^r, corresponding to content space c and style space s, respectively. The latent representations c and s are obtained by sampling from $q_\theta(c|Y^r)$ and $q_\theta(s|Y^r)$. The reconstruction process can be formulated as $p_\theta(Y^r|[c;s])$, representing the probability of generating input Y^r conditioned on the combination of content c and style s. Herein θ represents the parameters of the above encoders and reconstruction decoder. Because of the intractable integral of the marginal likelihood $p_\theta(Y^r)$ (Kingma and Welling 2013), the posterior $q_\theta(c|Y^r)$ and $q_\theta(s|Y^r)$ are simulated by variational approximation $q_\phi(c|Y^r)$ and $q_\phi(s|Y^r)$, where ϕ is the parameters for q.

When learning the VAE, the objective is to maximize the variational lower bound of $\log p_\theta(Y^r)$:

$$\begin{aligned}\mathcal{L}_{VAE} = {} & \lambda_{KL_c}\text{KL}(q_\phi(c|Y^r)\|p_\theta(c)) \\ & + \lambda_{KL_s}\text{KL}(q_\phi(s|Y^r)\|p_\theta(s)) \\ & - \text{E}_{q_\phi(c|Y^r),q_\phi(s|Y^r)}[\text{log}p_\theta(Y^r|[c;s])], \end{aligned} \tag{1}$$

where the KL$(\cdot)$ denotes KL-divergence, the regularization for encouraging the approximated posterior $q_\phi(c|Y^r)$ and $q_\phi(s|Y^r)$ to be close to the prior $p_\theta(c)$ and prior $p_\theta(s)$, *i.e.,* standard Gaussian distribution. E$[\cdot]$ is the reconstruction loss conditioned on the approximation posterior $q_\phi(c|Y^r)$ and $q_\phi(s|Y^r)$.

Style Space Constraint. Overall, to ensure that the style information is stored in the style space s, while the content information is filtered, we first design a style space constraint applied on the VAE in feature extractor. Generally, we use a classifier to determine the style label of s, and a discriminator to identity which document s corresponds to. If the style space s can be successfully classified as attractive headlines, and the corresponding document cannot be distinguished, then we can safely draw the conclusion that s only contains style information.

To disentangle the style information, we first randomly select an attractive headline Y^a and an unattractive headline Y^n as the two candidates of the classifier. Then we use the same embedding matrix to map each word into a high-dimensional vector space, and use Bi-RNN$_\text{Y}$ to obtain the representation h^{Y^a} and h^{Y^n} for the title Y^a and Y^n, respectively. The last states of Bi-RNN$_\text{Y}$ are concatenated as the overall representation vector h^{Y^a} and h^{Y^n}. In this way, the classification result is obtained as:

$$C_s(Y^*) = \text{softmax}(W_{ss}[s; h^{Y^*}] + b_{ss}). \tag{2}$$

Since the classifier aims to maximize the probability of matching the randomly selected title Y^a with attractive style, the loss of the classifier network is defined as:

$$\mathcal{L}_{Cs} = -\log(C_s(Y^a)) - \log(1 - C_s(Y^n)). \tag{3}$$

As for filtering the content information from the style space s, we achieve this goal following the generative adversarial way (Goodfellow et al. 2014). Generally, we employ a discriminator to distinguish the document that corresponds to the prototype headline from two candidates, while the feature extractor is trained to encode the style space from which its adversary cannot predict the corresponding document. The positive sample is the prototype document X^r, and the negative sample is a random document, denoted by X^q. Similar to before, we use the same embedding matrix and Bi-RNN$_\text{X}$ to obtain the negative sample representation h^{X^q}. We build a two-way softmax discriminator on the style space s to predict

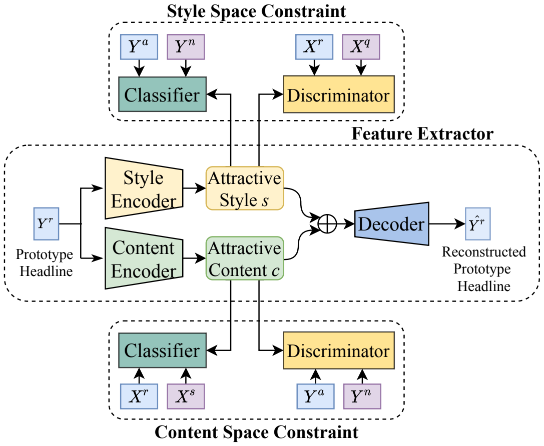


Figure 2: Framework of disentanglement module. Y^r is the input prototype headline and $\hat{Y}^r$ is the reconstructed prototype headline. X^r denotes the prototype document, X^s denotes the most similar document with X^r, and X^q denotes a random document. Y^a is a random attractive headline and Y^n is a random unattractive headline.

its corresponding document as:

$$D(X^*) = \text{softmax}(W_{sc}[s; h^{X^*}] + b_{sc}), \tag{4}$$

where h^{X^*} can be the ground truth prototype document representation h^{X^r} or the negative document representation h^{X^q}.

The training objective of the discriminator is to maximize the log-likelihood for correct classification, while the feature extractor aims to encode the style space where classifier cannot predict the correct label:

$$\mathcal{L}_S^d = -\log(D(X^r)) - \log(1 - D(X^q)), \tag{5}$$
$$\mathcal{L}_S^g = -\log(1 - D(X^r)). \tag{6}$$

In this way, the classifier ensures that the style space s stores the attractive style information and the discriminator guarantees that it does not contain content information.

Content Space Constraint. Similar to style space constraint, the purpose of content space constraint is to ensure that the content information of the prototype headline is stored in content space c and style information is not. Therefore, a classifier is utilized to predict which document the content space c matches, and a discriminator is employed to distinguish its style label from attractive or unattractive.

For classifier, the prototype document X^r and its most similar document X^s are provided as the two candidates, which improves the difficulty of the classifier. If the classifier can successfully identify the corresponding sample from extremely similar candidates, then we can say that they achieve good performance, and the content is successfully disentangled. The content-oriented loss to regularize the content space is obtained as:

$$C_c(X^*) = \text{softmax}(W_{cc}[c; h^{X^*}] + b_{cc}), \tag{7}$$
$$\mathcal{L}_{Cc} = -\log(C_c(X^r)) - \log(1 - C_c(X^s)), \tag{8}$$

where h^{X^*} can be the representation of X^r and X^s obtained by the same embedding martrix and Bi-RNN$_\text{X}$.

For discriminator, positive and negative samples are the random attractive title Y^a and unattractive title Y^n, respectively, which are selected in the same way as in style space constraint. The training objective of the feature extractor is to encode the content space from which its adversary cannot predict the corresponding style label as shown in Equation 10, while the discriminator is trained to maximize the log-likelihood for correct style label as Equation 11:

$$D(Y^*) = \text{softmax}(W_{cs}[c; h^{Y^*}] + b_{cs}), \tag{9}$$
$$\mathcal{L}_C^d = -\log(D(Y^a)) - \log(1 - D(Y^n)), \tag{10}$$
$$\mathcal{L}_C^g = -\log(1 - D(Y^a)). \tag{11}$$

Highlight Polish

Intuitively, the prototype attractive content space can help find attractive highlights in the document. Hence, following Chen et al. (2018), we design a highlight polish module, utilizing the disentangled attractive content representation c to polish the document representation.

Polish module consists of an RNN layer made up of Selective Recurrent Units (SRUs), a modified version of the original GRU. Generally speaking, SRU decides which part of the information should be updated based on both the polished document and the attractive content representation:

$$h_t^p = \text{SRU}(h_t^{X^d}, c), \tag{12}$$

where h_t^p denotes the t-th hidden state, and c is the attractive content space extracted from disentanglement module.

Headline Generator

The headline generator targets at generating a headline based on the polished document representation h_t^p following the extracted style space s. The final state of the input document representation h^{X^d} is employed as the initial state d_0 of the RNN decoder, and the procedure of t-th generation is calculated as:

$$d_t = \text{LSTM}_{\text{dec}}(d_{t-1}, [e(y_{t-1}); h_{t-1}^E]). \tag{13}$$

d_t is the hidden state of the t-th decoding step, $e(y_{t-1})$ is the embedding of last generated word y_{t-1}, and h_{t-1}^E is the context vector calculated by the standard attention mechanism (Bahdanau, Cho, and Bengio 2014) in Equation 17.

To take advantage of the original document hidden states $h_t^{X^d}$ and the polished document hidden states h_t^p, we combine them both into headline generation by a dynamic attention:

$$\delta_{it} = \frac{\exp(f(h_i^*, d_t))}{\sum_j \exp(f(h_j^*, d_t))}, \tag{14}$$
$$g_t^* = \sum\nolimits_i \delta_{it} h_i^*, \tag{15}$$

where h_i^* can be a polished document state h_i^p or an original document state $h_i^{X^d}$. The matching function f is designed as $f = h_i^* W_f d_t$ which is simple but efficient. When combining

$g_t^{X^d}$ and g_t^p, we use an "editing gate" γ, which is determined by the decoder state d_t:

$$\gamma = \sigma(W_g d_t + b_g), \tag{16}$$

$$h_t^E = \gamma g_t^{X^d} + (1-\gamma) g_t^p, \tag{17}$$

where σ denotes the sigmoid function. h_t^E is further concatenated with the decoder state d_t and fed into a linear layer to obtain the decoder output state d_t^o:

$$d_t^o = W_o[d_t; h_t^E] + b_o. \tag{18}$$

To incorporate the guidance of the extracted style representation s, we combine the decoder output state with s using a "guidance gate" γ_s:

$$\gamma_s = \sigma(W_g d_t + b_g), \tag{19}$$

$$\hat{d}_t^o = \gamma_s d_t^o + (1-\gamma_s) s. \tag{20}$$

In this way, the decoder can automatically decide the extent to which the attractive style representation is incorporated. Finally, we obtain the generated word distribution P_v:

$$P_v = \mathrm{softmax}(W_v \hat{d}_t^o + b_v). \tag{21}$$

The loss is the negative log likelihood of the target word y_t:

$$\mathcal{L}_{seq} = -\sum_{t=1}^{n^d} \log P_v(y_t). \tag{22}$$

To handle the out-of-vocabulary problem, we also equip our decoder with a pointer network (See, Liu, and Manning 2017). As our model is trained in an adversarial manner, we separate our model parameters into two parts:

(1) *discriminator module* contains the parameters of the discriminator in the style space constraint, which is optimized by $\mathcal{L}_D$ as:

$$\mathcal{L}_D = \mathcal{L}_S^d + \mathcal{L}_C^d. \tag{23}$$

(2) *generation module* consists of all other parameters in the model, optimized by $\mathcal{L}_G$ calculated as:

$$\mathcal{L}_G = \mathcal{L}_{VAE} + \mathcal{L}_{Cs} + \mathcal{L}_{Cc} + \mathcal{L}_S^g + \mathcal{L}_C^g + \mathcal{L}_{seq}. \tag{24}$$

Experimental Setup

Dataset

We use the public Tencent Kuaibao Chinese News dataset[1] proposed by Qin et al. (2018). The dataset contains 160,922 training samples, 1,000 validation samples, and 1,378 test samples. All these samples have more than 20 comments, which can be regarded as attractive cases (Xu et al. 2019). Since we use unattractive headlines to assist in the disentanglement of content and style, we collect 1,449 headlines with less than 20 comments from the same Kuaibao website as an unattractive headline dataset. The vocabulary size is set to 100k for speed purposes. Lucene[2] is employed to retrieve the prototype document-headline pair. Concretely, the document-headline pair in the training dataset that is most similar to the current document-headline pair is selected as prototype pair. The similarity of document-headline pairs is calculated by the scalar product of the TF-IDF vector (Luhn 1958) of each document.

Comparisons

We compare our proposed method against several baselines: (1) **Lead** (Nallapati, Zhai, and Zhou 2017; See, Liu, and Manning 2017) selects the first sentence of document as the summary. (2) **Proto** directly uses the retrieved headline as the generated one. (3) **PG** (See, Liu, and Manning 2017) is a sequence-to-sequence framework with attention mechanism and pointer network. (4)**R^{3}Sum** extends the seq2seq framework to jointly conduct template-aware summary generation (Cao et al. 2018). (5) **Unified** (Hsu et al. 2018) combines the strength of extractive and abstractive summarization. (6) **PESG** (Gao et al. 2019a) generates summarization through prototype editing. (7) **SAGCopy** (Xu et al. 2020) is a augmented Transformer with self-attention guided copy mechanism. (8) **GPG** generates headlines by "editing" pointed tokens instead of hard copying (Shen et al. 2019). (9) **Sensation** generates attractive headlines using reinforcement learning method (Xu et al. 2019). (10) **SLGen** (Zhang et al. 2020), encodes relational information of sentences and automatically learns the sentence graph for headline generation.

Evaluation Metrics

ROUGE: We evaluate models using standard full-length ROUGE F1 (Lin 2004) following previous works (Gao et al. 2019a; Xu et al. 2019). ROUGE-1, ROUGE-2, and ROUGE-L refer to the matches of unigram, bigrams, and the longest common subsequence, respectively.

BLEU: To evaluate our model more comprehensively, we also use the metric BLEU proposed by Papineni et al. (2002) which measures word overlap between the generated text and the ground-truth. We adopt BLEU-1∼4 and BLEU which adds a penalty for length.

Human evaluation: Schluter (2017) noted that only using the autometric to evaluate generated text can be misleading. Therefore, we also evaluate our model by human evaluation. We randomly sample 100 cases from the test set and ask three different educational-levels annotators to score the headlines generated by PESG, SAGCopy, Sensation, and our model DAHG. The statistical significance of observed differences is tested using a two-tailed paired t-test and is denoted using $^\blacktriangle$(or $^\blacktriangledown$) for strong (or weak) significance for $\alpha = 0.01$.

Implementation Details

Our experiments are implemented in Tensorflow (Abadi et al. 2016) on an NVIDIA GTX 1080 Ti GPU. Experiments are performed with a batch size of 64. We pad or cut the input document to 400 words and the prototype headline to 30 words. The maximum decode step is set to 30, and the minimum to 10 words. We initialize all of the parameters in the model using a Gaussian distribution. We choose Adam optimizer for training, and use dropout in the VAE encoder with keep probability as 0.8. For testing, we use beam search with size 4 to generate a better headline. It is well known that a straightforward VAE with RNN decoder always fails to encode meaningful information due to the vanishing latent variable problem (Bowman et al. 2015). Hence, we use the BOW loss along with KL annealing of 10,000 batches to achieve better performance. Readers can refer to (Zhao, Zhao, and Eskenazi 2017) for more details.

[1] `https://kuaibao.qq.com/`

[2] `https://lucene.apache.org`

	BLEU	BLEU-1	BLEU-2	BLEU-3	BLEU-4	R-1	R-2	R-L
extractive summarization								
Lead	4.82	13.07	4.89	3.27	2.58	19.70	7.98	17.41
Proto	6.62	22.72	7.14	4.09	2.89	22.22	6.90	19.61
abstractive summarization								
PG (See, Liu, and Manning 2017)	10.40	24.42	10.59	7.43	6.11	25.92	10.86	23.40
R^{3}Sum (Cao et al. 2018)	11.22	26.49	11.51	7.95	6.53	27.78	11.81	25.02
Unified (Hsu et al. 2018)	10.55	25.09	10.85	7.51	6.06	27.94	11.68	25.37
PESG (Gao et al. 2019a)	11.08	26.62	11.43	7.84	6.33	28.21	11.87	25.56
SAGCopy (Xu et al. 2020)	11.21	27.65	11.33	7.75	6.50	28.71	11.27	25.74
headline generation								
GPG (Shen et al. 2019)	10.51	24.48	11.00	7.60	5.96	26.12	11.61	23.84
Sensation (Xu et al. 2019)	10.70	26.18	10.89	7.54	6.10	26.28	10.84	23.94
SLGen (Zhang et al. 2020)	11.48	26.50	11.76	8.28	6.74	27.33	11.83	25.00
DAHG	**12.74**	**28.93**	**13.34**	**9.27**	**7.38**	**29.73**	**13.55**	**27.03**
+BERT	11.43	27.89	12.06	7.93	6.74	28.61	12.27	24.08

Table 1: BLEU and ROUGE scores comparison with baselines. All our ROUGE scores have a 95% confidence interval of at most ±0.22 as reported by the official ROUGE script.

Experimental Result

Overall Performance

We compare our model with the baselines in Table 1. Firstly, Proto outperforms Lead but underperforms other baselines, indicating that prototype information is helpful for our task, but directly taking it as input only leads to a small improvement. Secondly, abstractive methods outperform all extractive methods, demonstrating that Kuaibao is a dataset suitable for abstractive summarization. Finally, our model outperforms PESG by 14.99%, 5.39%, 14.15%, 5.75%, and outperforms SAGCopy by 13.65%, 3.55%, 20.23%, 5.01% in terms of BLEU, ROUGE-1, ROUGE-2, and ROUGE-L, respectively, which proves the superiority of our model. Besides, it can also be seen that with BERT augmented as the encoder of our model, the results are slightly below DAHG, which demonstrates that simply superimposing the pre-training module on our model does not help improve the effect.

For the human evaluation, we ask annotators to rate headlines generated by PESG, SAGCopy, Sensation, and our model DAHG according to fluency, consistency, and attractiveness. The rating score ranges from 1 to 3, with 3 being the best. Table 2 lists the average scores of each model, showing that DAHG outperforms other baseline models among all metrics. Besides, to directly compare the attractiveness of each model, we ask annotators to select the headline they are most likely to click among the four candidate headlines. The click-rate is also listed in Table 2, where our model wins 22% click among all baselines. This direct comparison shows that our headlines are more eye-catching and stand out above the rest. Note that, the generated headlines of DAHG are not clickbait, since they are generally faithful to the content of the documents as the consistency score shows. The kappa statistics are 0.49, 0.54, and 0.47 for fluency, consistency, and attractiveness, respectively, which indicates the moderate agreement between annotators. To verify the significance of these results, we also conduct the paired student t-test between our model and Sensation (the row with shaded background). We obtain a p-value of 5×9^{-9}, 3×10^{-7}, and 5×10^{-6} for fluency, consistency, and attractiveness.

	Flu	Con	Attr	Clr
PESG	2.07	2.14	1.70	0.11
SAGCopy	2.21	2.08	1.93	0.13
Sensation	2.02	1.97	2.19	0.27
DAHG	**2.59**$^{▲}$	**2.51**$^{▲}$	**2.47**$^{▲}$	**0.49**

Table 2: Fluency(Flu), consistency(Con), attractiveness(Attr) and click-rate(Clr) comparison by human evaluation.

Ablation Study

In order to verify the effect of each module in DAHG, we conduct ablation tests in Table 3. We first verify the effectiveness of separating style and content in DAHG-D, where we directly use the prototype headline to polish the input document. DAHG-C omits the process of polishing by attractive content representation, and DAHG-S does not use attractive style to guide the generation process. All ablation models perform worse than DAHG in terms of all metrics, demonstrating the preeminence of DAHG. Specifically, the polishing process contributes most to DAHG, and separating style and content is also important in achieving high performance.

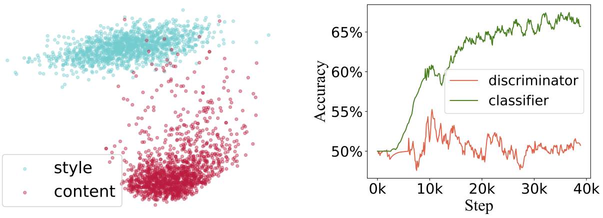
Analysis of Disentanglement

The vectors in the style and content spaces on test data are visualized in Figure 3(a). For visualization purpose, we reduce the dimension of the latent vector with t-SNE (Maaten and Hinton 2008). It can be observed that points representing same space located nearby in the 2D space while different ones. This suggests that the disentanglement module can disentangle the prototype headline into style and content space.

Besides, the accuracy curves of classifier and discriminator in style space constraint are shown in Figure 3(b). When the

	R-1	R-2	R-L
All	29.73	13.55	27.03
DAHG-D	27.69	12.01	25.32
DAHG-C	27.60	11.97	24.93
DAHG-S	27.82	12.43	25.32

Table 3: ROUGE scores of different ablation models of DAHG.



(a)

(b)

Figure 3: (a) Visualization result for disentanglement. (b) Accuracy curve of classifier and discriminator in style space constraint.

training begins, the accuracy of the discriminator fluctuates from time to time, which verifies the adversarial training. After several steps, the training converges, and the accuracy of the discriminator stays around 50%, which means that the content information is successfully excluded from style space. The discriminator cannot perform better than a random guess. As for the accuracy curve of the classifier, the accuracy score of choosing the style label is on a steady upward curve and finally reaches convergence along the training progresses, which proves that the style information is contained in the style space. The accuracy curves in content space constraint are similar, and are omitted due to the limited space.

Analysis of Sampling Effect

We finally investigate the influence of sampling different latent variables in the attractive style and content space. As shown in Table 4, for a single document, we choose one similar prototype headline and one randomly-selected prototype headline as the input of DAHG, and examine the quality of the generated headline. Different prototype headlines lead to different sampling results in the style and content space, and correspondingly, the more proper the prototype is, the better the sampling result and the headline are. Case 1 in Table 4 is a good sampling case, where prototype 1 is selected as the most similar one. We can see that the style and content of the generated headline match the document well, emphasizing on "be contained by students". As for case 2, the prototype focuses on gender information, hence leads to a bad sampling result, emphasizing "assistant is a girl", which does not cover the necessary information of a proper headline.

We also compare our model with several baselines in Table 4. Most baselines can generate fluent headlines in this case. However, they miss the attractive style and can include unattractive content. The headline generated by PESG is a plain statement. Sensation generates more attractive headline, but is not faithful to the document, and includes unnecessary content. While for our model in case 1, DAHG captures the keywords "contained" and "police" that not only cover the events in the document but also draw mass attention.

Document: I believe everyone is familiar with Xu Xiaodong in the picture above. A fight between modern boxing and traditional Taiji made him become a social focus. Today, another containment incident happened, which made Xu a little nervous. At 16:25 this afternoon, Xu was contained by seven Taiji students during the live broadcast. When talking about the situation at that time, Xu said: "Seven students won't let me go. I'm afraid." When asked what he thought of when he was surrounded, Xu said, "I worried about what will happen to my assistant? My assistant is a girl, I cannot fight back. So I call the police.". Although Xu is sometimes reckless, his original intention is admirable. He did not try to fight against the martial, instead, he only meant to fight against the cheaters in the traditional martial arts.
Reference headline:Emergencies! Xu Xiaodong is contained by seven Taiji disciples.
Case 1: ***Prototype 1:*** Xu Xiaodong behaves beyond the bottom line and has been despised by the whole traditional martial arts. ***DAHG 1:*** Xu Xiaodong is contained by Taiji disciples and calls the police! Saying that he might have accidents.
Case 2: ***Prototype 2:*** What are the most beautiful girls like in a boy's eyes? ***DAHG 2:*** Xu Xiaodong: My assistant is a girl. I cannot fight-back.
PESG: Xu Xiaodong: a modern fight vs. Traditional Taiji makes him the focus of the society.
SAGCopy: Xu Xiaodong: I am a fighting madman, and the person I fear most is myself.
Sensation: Xu Xiaodong fights against all Taiji students in the Wulin. My favorite is mine.

Table 4: Case study to verify the influence of sampling on style and content space. The text in blue denotes attractive information related to the prototype headline, and text in red denotes information that is related to the prototype but not proper for a headline.

Conclusion

In this paper, we propose a Disentanglement-based Attractive Headline Generator (DAHG) to generate an attractive headline. Our model is built on the fact that the attractiveness of the headline comes from both style and content aspects. Given the prototype document-headline pair, DAHG disentangles the attractive content and style space from the prototype attractive headline. The headline generator generates attractive headlines under the guidance of both. Our model achieves state-of-the-art results in terms of ROUGE scores and human evaluations by a large margin. In near future, we aim to bring the model online.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments. This work was supported by the National Key Research and Development Program of China (No. 2020YFB1406702), the National Science Foundation of China (NSFC No. 61876196 and No. 61672058), Beijing Outstanding Young Scientist Program (No. BJJWZYJH012019100020098). Rui Yan is supported as a young fellow at Beijing Academy of Artificial Intelligence (BAAI).

Ethics Statement

The motivation of our paper is to generate attractive headline for news, which is useful for both readers and writers. On one hand, generating headlines that can trigger high click-rate is especially important for different avenues and forms of media to compete for user's limited attention. On the other hand, only with the help of a good headline, the outstanding article can be discovered by readers.

The generated headlines sometimes can sometimes be cilckbaits, which cause users to fall into information garbage. However, theoretically, all headline generation modules face the same problem. In the model design, we maintain the consistency of the headline's and the document's content information as much as possible, so as to ensure that the generated headline is faithful to the content of the document to a large extent. In the future, we will continue to explore how to enhance the consistency of content in the headline generation.

References

Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 265–283.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A. M.; Jozefowicz, R.; and Bengio, S. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349* .

Cao, Z.; Li, W.; Li, S.; and Wei, F. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 152–161.

Chan, Z.; Chen, X.; Wang, Y.; Li, J.; Zhang, Z.; Gai, K.; Zhao, D.; and Yan, R. 2019. Stick to the Facts: Learning towards a Fidelity-oriented E-Commerce Product Description Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4960–4969.

Chan, Z.; Zhang, Y.; Chen, X.; Gao, S.; Zhang, Z.; Zhao, D.; and Yan, R. 2020. Selection and Generation: Learning towards Multi-Product Advertisement Post Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3818–3829.

Chen, M.; Tang, Q.; Wiseman, S.; and Gimpel, K. 2019a. A multi-task approach for disentangling syntax and semantics in sentence representations. *arXiv preprint arXiv:1904.01173* .

Chen, X.; Chan, Z.; Gao, S.; Yu, M.-H.; Zhao, D.; and Yan, R. 2019b. Learning towards Abstractive Timeline Summarization. In *IJCAI*, 4939–4945.

Chen, X.; Cui, Z.; Zhang, J.; Wei, C.; Cui, J.; Wang, B.; Zhao, D.; and Yan, R. 2021. Reasoning in Dialog: Improving Response Generation by Context Reading Comprehension. In *AAAI*.

Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; and Abbeel, P. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, 2172–2180.

Chen, X.; Gao, S.; Tao, C.; Song, Y.; Zhao, D.; and Yan, R. 2018. Iterative Document Representation Learning Towards Summarization with Polishing. *arXiv preprint arXiv:1809.10324* .

Cho, K.; van Merrienboer, B.; Çaglar Gülçehre; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*.

Dehghani, M.; Gouws, S.; Vinyals, O.; Uszkoreit, J.; and Kaiser, Ł. 2018. Universal transformers. *arXiv preprint arXiv:1807.03819* .

Fu, Z.; Tan, X.; Peng, N.; Zhao, D.; and Yan, R. 2018. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Gao, S.; Chen, X.; Li, P.; Chan, Z.; Zhao, D.; and Yan, R. 2019a. How to Write Summaries with Patterns? Learning towards Abstractive Summarization through Prototype Editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3732–3742.

Gao, S.; Chen, X.; Li, P.; Ren, Z.; Bing, L.; Zhao, D.; and Yan, R. 2019b. Abstractive text summarization by incorporating reader comments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6399–6406.

Gao, S.; Chen, X.; Ren, Z.; Zhao, D.; and Yan, R. 2020a. From Standard Summarization to New Tasks and Beyond: Summarization with Manifold Information. In *IJCAI*.

Gao, S.; Chen, X.; Ren, Z.; Zhao, D.; and Yan, R. 2020b. From Standard Summarization to New Tasks and Beyond: Summarization with Manifold Information .

Gavrilov, D.; Kalaidin, P.; and Malykh, V. 2019. Self-Attentive Model for Headline Generation. In *European Conference on Information Retrieval*, 87–93. Springer.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; and Lerchner, A. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *ICLR* 2(5): 6.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Hsu, W.-T.; Lin, C.-K.; Lee, M.-Y.; Min, K.; Tang, J.; and Sun, M. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. *arXiv preprint arXiv:1805.06266* .

Iyyer, M.; Wieting, J.; Gimpel, K.; and Zettlemoyer, L. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059* .

Jin, D.; Jin, Z.; Zhou, J. T.; Orii, L.; and Szolovits, P. 2020. Hooks in the Headline: Learning to Generate Headlines with Controlled Styles. *arXiv preprint arXiv:2004.01980* .

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* .

Li, J.; Bing, L.; Qiu, L.; Chen, D.; Zhao, D.; and Yan, R. 2019. Learning to write stories with thematic consistency and wording novelty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1715–1722.

Li, J.; Luong, M.-T.; and Jurafsky, D. 2015. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057* .

Li, M.; Chen, X.; Gao, S.; Chan, Z.; Zhao, D.; and Yan, R. 2020. VMSMO: Learning to Generate Multimodal Summary for Video-based News Articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9360–9369.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Liu, D.; Li, J.; Yu, M.-H.; Huang, Z.; Liu, G.; Zhao, D.; and Yan, R. 2020. A Character-Centric Neural Model for Automated Story Generation. In *AAAI*, 1725–1732.

Luhn, H. P. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development* 2(2): 159–165.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9(Nov): 2579–2605.

Nallapati, R.; Zhai, F.; and Zhou, B. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 311–318. Association for Computational Linguistics.

Qin, L.; Liu, L.; Bi, V.; Wang, Y.; Liu, X.; Hu, Z.; Zhao, H.; and Shi, S. 2018. Automatic article commenting: the task and dataset. *arXiv preprint arXiv:1805.03668* .

Schluter, N. 2017. The limits of automatic summarisation according to rouge. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 41–45.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368* .

Shen, X.; Zhao, Y.; Su, H.; and Klakow, D. 2019. Improving Latent Alignment in Text Summarization by Generalizing the Pointer Generator. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3753–3764.

Shu, K.; Wang, S.; Le, T.; Lee, D.; and Liu, H. 2018. Deep headline generation for clickbait detection. In *2018 IEEE International Conference on Data Mining (ICDM)*, 467–476. IEEE.

Tan, J.; Wan, X.; and Xiao, J. 2017. From Neural Sentence Summarization to Headline Generation: A Coarse-to-Fine Approach. In *IJCAI*, 4109–4115.

Xie, P.; Cui, Z.; Chen, X.; Hu, X.; Cui, J.; and Wang, B. 2020. Infusing Sequential Information into Conditional Masked Translation Model with Self-Review Mechanism. In *COLING*, 15–25.

Xu, P.; Wu, C.-S.; Madotto, A.; and Fung, P. 2019. Clickbait? Sensational Headline Generation with Auto-tuned Reinforcement Learning. *arXiv preprint arXiv:1909.03582* .

Xu, S.; Li, H.; Yuan, P.; Wu, Y.; He, X.; and Zhou, B. 2020. Self-Attention Guided Copy Mechanism for Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1355–1362.

Yu, M.-H.; Li, J.; Liu, D.; Zhao, D.; Yan, R.; Tang, B.; and Zhang, H. 2020. Draft and edit: Automatic storytelling through multi-pass hierarchical conditional variational autoencoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 1741–1748.

Zhang, R.; Guo, J.; Fan, Y.; Lan, Y.; and Cheng, X. 2020. Structure Learning for Headline Generation. In *AAAI*, 9555–9562.

Zhang, X.; Lapata, M.; Wei, F.; and Zhou, M. 2018. Neural latent extractive document summarization. *arXiv preprint arXiv:1808.07187* .

Zhao, T.; Zhao, R.; and Eskenazi, M. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960* .

Zhou, Q.; Yang, N.; Wei, F.; Huang, S.; Zhou, M.; and Zhao, T. 2018. Neural document summarization by jointly learning to score and select sentences. *arXiv preprint arXiv:1807.02305* .