Contrastive Learning enhanced Author-Style Headline Generation

Hui Liu [*1], **Weidong Guo** [*†1], **Yige Chen** [2] **Xiangyang Li** [1]
[1]Platform and Content Group, Tencent
[2]College of Computer Science and Artificial Intelligence, Wenzhou University
[1]{pvopliu,weidongguo,xiangyangli}@tencent.com
[2]yigechen@wzu.edu.cn

Abstract

Headline generation is a task of generating an appropriate headline for a given article, which can be further used for machine-aided writing or enhancing the click-through ratio. Current works only use the article itself in the generation, but have not taken the writing style of headlines into consideration. In this paper, we propose a novel Seq2Seq model called CLH3G (**C**ontrastive **L**earning enhanced **H**istorical **H**eadlines based **H**eadline **G**eneration) which can use the historical headlines of the articles that the author wrote in the past to improve the headline generation of current articles. By taking historical headlines into account, we can integrate the stylistic features of the author into our model, and generate a headline not only appropriate for the article, but also consistent with the author's style. In order to efficiently learn the stylistic features of the author, we further introduce a contrastive learning based auxiliary task for the encoder of our model. Besides, we propose two methods to use the learned stylisic features to guide both the pointer and the decoder during the generation. Experimental results show that historical headlines of the same user can improve the headline generation significantly, and both the contrastive learning module and the two style features fusion methods can further boost the performance.

1 Introduction

Natural Language Generation tasks have achieved great success both in research and application, such as Neural Machine Translation (Bahdanau et al., 2014), Headline Generation (Jin et al., 2020; Ao et al., 2021) and so on. In many real-life reading scenarios, an attractive headline of the article can immediately grab the readers and then lead them to view the whole article. Thus, headline generation (HG) is becoming an important task and draw increasing attention nowadays, which aims to automatically generate the appropriate headline for a given aritcle.

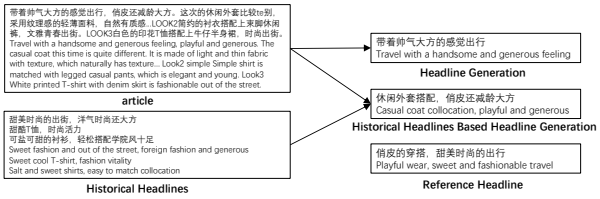

Figure 1: Comparison of general Headline Generation and Historical Headlines based Headline Generation.

Earlier research of HG (Dorr et al., 2003; Tan et al., 2017) mainly focus on generating a fluent and relevant headline for a given news article to alleviate the author's work in a machine-aided writing way. More recent works (Zhang et al., 2018; Xu et al., 2019; Jin et al., 2020) intend to generate attractive headlines for articles so as to get higher click-through ratio and further directly improve the profit of the online social media platforms. Furthermore, some works (Liu et al., 2020; Ao et al., 2021) try to generate keyphrase-aware and personalized headlines to meet the requirements of different application scenarios and further satisfy users' personal interests.

However, most of these existing methods only use the information of the article to generate the headline but ignore the author's historical headlines. In general, the headlines manually designed by the author are usually more suitable for the author's articles but the design style of the headline may be quite different from the writing style of the article and we can hardly obtain the author's headline style only through the content of the article. Therefore, when we integrate existing historical headlines into the HG model to learn the headline style of the author, such as grammar and syntax, the model can generate more appropriate headlines for machine-aided writing. For example, as shown in Figure 1, all of the historical headlines and the

[*]These authors contributed equally to this work.
[†]Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 5063 - 5072
December 7-11, 2022

reference headline are composed of two clauses. The generated headline of the historical headlines based HG model has the same syntax with the historical headlines, which makes it more likely to be accepted by the author and more attractive than the generated headline with distinctive syntax by the general HG model.

To our best knowledge, there is no corpus that contains both news articles and corresponding authorships to meet the requirement of our experiments. Therefore, in this paper, we build a new dataset named H3G(**H**istorical **H**eadlines Based **H**eadline **G**eneration) to explore the research of historical headlines based HG. We collect the H3G dataset from the online social media platform Tencent QQBrowser, which contains more than 380K news articles from more than 23K different authors. The detailed introduction of the H3G dataset is discussed in the Experiment section.

Besides, we propose a novel Contrastive Learning enhanced Historical Headlines based Headline Generation (CLH3G) model to extract and learn headline styles for HG. Inspired by the existing style transfer models (Lample et al., 2018; Dai et al., 2019), we represent the headline style of the historical headlines as a single vector. Such a design can not only reduce the computation cost of the historical headlines representation, but also facilitate the integration of historical headlines information on the decoder side of the HG model. Besides, two different methods are applied to guide the generation of the author-style headlines through the single headline vector. The first style vector fusion method can instruct the decoder to generate author-style target headline representation, and the other controls the generated words of the pointer-generator network. What's more, on the encoder side of Sequence-to-Sequence (Seq2Seq) model, we also use Contrastive Learning (CL) to distinguish headlines from different authors as an auxiliary task, which is consistent with and conduce to the extraction of the headline style.

Experimental results on automatic metrics ROUGE and BLEU and Human evaluation show that the historical headlines can greatly improve the effectiveness of HG compared with general HG models, and both of the two style vector fusion methods and Contrastive Learning based auxiliary task can also improve the performance. We also train a Contrastive learning classifier to distinguish headlines from different authors, and find our CLH3G model can generate more author-style headlines than the general HG models and other compared models.

To this end, our main contributions are summarized as follows:

- We propose a new HG paradigm namely Historical Headlines based HG to generate author-style headlines, which can be used for machine-aided writing and click-through ratio enhancing.
- We propose a novel model CLH3G, which utilizes two headline style vector fusion methods and contrastive learning to make full use of historical headlines.
- We construct a new Historical Headlines based HG dataset namely H3G and conducted abundant experiments on it. Experimental results show that the historical headlines are beneficial to headline generation, and both the two headline style vector fusion methods and Contrastive Learning can also improve the HG models.[1]

2 Related Work

Headline Generation focus on generating a suitable or attractive headline for a given article. We divide HG into three categories, namely general HG, style-based HG and adaptive HG.

The general HG models want to generate a fluent and suitable headline given an article. An early work (Dorr et al., 2003) uses linguistically-motivated heuristics to generate a matching headline. This method is very safe, because all words in the generated headline are selected from the original article. Then, some works (See et al., 2017; Gavrilov et al., 2019) use End-to-End neural networks to generate headlines. These methods achieve the state-of-the-art results and are very convenient for training and inference. Besides, (Tan et al., 2017) uses a coarse-to-fine model to generate headlines for long articles.

The style-based headline generation models aims to generate headline with specific styles. (Xu et al., 2019) uses Reinforcement Learning to generate sensational headlines to capture reader's interest. (Zhang et al., 2018) proposes dual-attention sequence-to-sequence model to generate question-style headlines, because they find question-style

[1]Our code is available at https://github.com/pvop/CLH3G

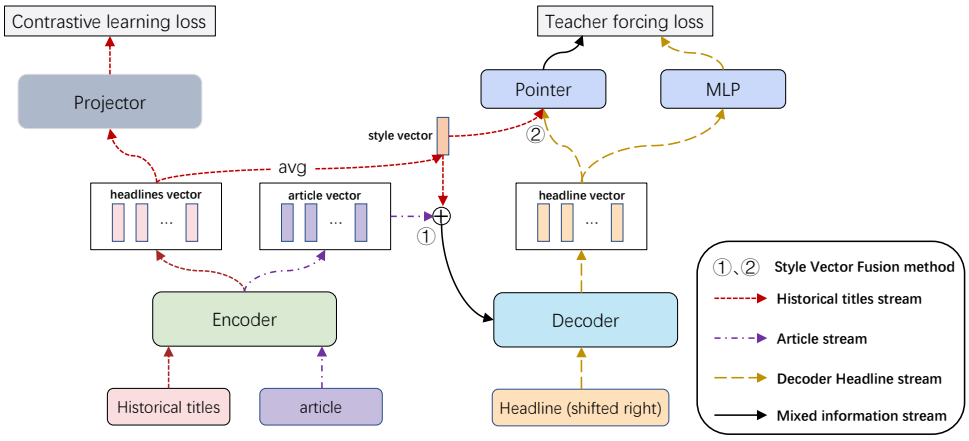

Figure 2: Our proposed Contrastive Learning enhanced Historical Headlines based Headline Generation Model.

headlines can get much higher click-through ratio. Besides, (Jin et al., 2020) uses parameter sharing scheme to generate general, humorous, romantic, click-baity headlines at the same time.

Adaptive headline generation models want to generate different headlines for different scenarios. (Liu et al., 2020) proposes to generate different headlines with different keywords, which can be used to generate different headlines for different search queries in search engines. (Ao et al., 2021) uses the user impression logs of news to generate personalized headlines for different users to satisfy their different interests.

Contrastive Learning is very popular recently for representation learning. CL was first used for vision understanding in (Chen et al., 2020). Subsequently, CL is also used in Natural Language Generation, including Conditional Text Generation (Lee et al., 2020), Dialogue Generation (Cai et al., 2020), Report Generation (Yan et al., 2021) and text summarization (Liu and Liu, 2021). In this paper, we use CL like (Chen et al., 2020), whose framework includes a neural network encoder and a small neural network projection.

3 Model

Figure 2 shows our proposed Contrastive Learning enhanced H3G (CLH3G) model, which is an End-to-End Seq2Seq generation model. We will briefly introduce the entire model in Section 3.1 and discuss the encoder and the CL based auxiliary task in Section 3.2. Finally, the decoder and two headline style vector fusion methods are presented in Section 3.3.

3.1 Problem and Architecture

Given an article and k headlines from other articles written by the same author, our model will generate a headline which is most suitable for this article and consistent with the headline writing style of the author. Formally, the CLH3G model uses the article $A = [w_1^A, w_2^A, ..., w_a^A]$ of the author X and some historical headlines $T = [t_1, t_2, ..., t_k]$ of X to automatically generate a new headline $H = [w_1^H, w_2^H, ..., w_h^H]$, which is suitable for A and consistent with the headline writing style of X.

Compared with previous HG methods, our model put more emphasis on learning the style of the input historical headlines to improve the performance. Specifically, during encoding, we use a single vector like (Lample et al., 2018; Dai et al., 2019) to derive the style information from the input headlines, and adopt CL to further distinguish the style among different authors. The CL module will not bring overhead because it shares the same encoder with the original HG model. Besides, we fuse two different methods to integrate the style information into the decoder: the first one is designed to influence the representation of the generated headline, and the other will guide the pointer module to copy author-style headline words. In the rest of this section, we will introduce the CLH3G model in detail.

3.2 Encoder and Contrastive Learning based Auxiliary Task

Transformer Seq2Seq Model (Vaswani et al., 2017) has achieved remarkable success in Natural Language Generation. Transformer consists of a self-attention multi-head encoder and a self-attention

multi-head decoder. In order to enhance the semantic representation capability of the encoder, we use the pre-trained BERT-base model (Devlin et al., 2018) to initialize the parameters of the encoder, which can generate superior article representation and headline vectors to improve the effectiveness of HG models.

As shown in Figure 2, the encoder represents the article A as $H_A \in R^{a\times d}$, where d is the hidden size of BERT-base. For each headline t_i in T, we use the encoder outputs at $[CLS]$ as its headline representation, so all historical headlines T are represented as $H_T \in R^{k\times d}$. Subsequently, we average the historical representation H_T to obtain a single style vector $s_t \in R^{1\times d}$, and H_T is also used to compute the CL loss.

Contrastive learning is a self-supervised method that can learn knowledge from unlabeled data. Recently, CL has achieved great success in many fields (Chen et al., 2020; Liu and Liu, 2021). Same as (Chen et al., 2020), our CL module consists of a neural network base encoder and a small neural network projection. The CL encoder and the CLH3G encoder share parameters, so that we only need to compute the headlines representation once for both headline style vector and CL loss function to avoid additional overhead. The projection is a two-layer fully connected feed-forward network. Instead of explicitly constructing positive examples like most CL models, we regard the headline pairs belonging to the same author as positive samples, and the other headlines in the same batch belonging to negative samples. The loss function of the positive pair of examples (i, j) is defined as

$$L_{i,j} = -log\frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{I}_{[k\neq i]} exp(z_i, z_k)/\tau} \quad (1)$$

where $\mathbb{I}_{[k\neq i]}$ is an indicator function evaluated to 1 iff $k \neq i$ and τ is a temperature parameter.

In the H3G dataset, all train and test samples contain at least one historical headline. Certainly, the CLH3G model can generate the general headline only with article information, but we mainly want to explore the performance of the model when adding historical headlines. During the training phase, the target headline is also used in CL module but not in the computation for the single style vector. We randomly select two headlines from the input historical headlines and target headline for CL loss function. During the inference phase, we do not use the CL module and the target headline, so that the CL module will not affect the inference speed.

3.3 Decoder and Two Headline Style Vector Fusion Methods

As shown in Figure 2, the shifted right target headline H_{rh} is imputed into the decoder to generate the target headline representation matrix $D_H \in R^{h\times d}$. During the computation of decoder, our first headline style vector fusion method is simply concatenating the article representation H_A and the single style vector s_t to $H_{At} \in R^{(a+1)\times d}$. This concatenated result H_{At} can guide the decoder to represent the shifted right target headline to generate a new headline with the same style of headlines in T. There are many overlapping words in the headline and the corresponding article, so we use the pointer module same as (See et al., 2017) to solve the out-of-vocabulary (OOV) problem and improve the performance of generation models. Our second headline style vector fusion method is to add the single style vector to the pointer module. On the one hand, we use the style vector to select words of the input article for $i-th$ generated word as:

$$\begin{aligned}\alpha(i) = \text{softmax}(&w_{alpha1}^T[H_A : S_t]+ \\ &w_{alpha2}^T[d_H^i : s_t] + b_{alpha})\end{aligned} \quad (2)$$

where $S_t \in R^{a\times d}$ is the result of s_t repeating a times, and w_{alpha1}, w_{alpha2} and b_{alpha} are learnable parameters. We use the headline vector d_H^i to produce the vocabulary distribution of the $i-th$ generated word as:

$$P_{vocab}(i) = \text{softmax}(V^T d_H^i + b_{vocab}) \quad (3)$$

where V_T and b_{vocab} are learnable parameters. On the other hand, the generation probability $P_{gen}(i) \in [0, 1]$ is computed by H_A, d_H^i and the single style vector s_t as:

$$\begin{aligned}P_{gen}(i) = \sigma(&w_{gen1}^T h_A + w_{gen2}^T d_H^i + w_{gen3}^T s_t + \\ &b_{gen}),\ \text{where } h_A = \sum_{j=0}^{a-1} \alpha(i)_j * H_{A_j}\end{aligned} \quad (4)$$

where w_{gen1}, w_{gen2}, w_{gen3}, b_{gen} are learnable parameters. The final probability distribution of the generated word i as a certain word w is:

$$\begin{aligned}P_w(i) =&p_{gen}(i)P_{vocab_w}(i)+ \\ &(1 - p_{gen}(i)) \sum_{j:w_j=w} \alpha(i)_j\end{aligned} \quad (5)$$

Dataset	#author	#article	#article per author	avg article length	avg headline length
H3G	23726	384868	16	1291	27

Table 1: The statistics of H3G dataset.

The final probability distribution is used to compute the teacher forcing loss, and the final loss function is:

$$Loss = L_{teacher\_forcing} + \lambda L_{contra\_learning} \tag{6}$$

where λ is a hyperparameter.

The two headline style vector fusion methods take different ways to influence the final headline generation. For a new article, there are many reasonable headlines for the article from content to syntax. The first one is similar to informing the decoder the desired headline style in advance, allowing the decoder to have a more explicit generation direction. Besides, headlines with different author styles have different word preferences. The second one can guide the choice of words in the pointer network and whether to use pointer or generator.

4 Experiments

4.1 Dataset

We collect our H3G dataset from an online social media platform Tencent QQBrowser. Some platform accounts are shared by more than one authors and publish a large number of articles every day. So we select the accounts who published 3-60 articles within two months in 2021. Finally, we get more than 380K different articles of more than 23K different authors, and the statistics of the H3G dataset are shown in Table 1. We randomly divide the H3G dataset into training set, validation set and test set. The validation set and test set contain 500 and 2000 samples respectively, and the rest of the articles are used as the training set. For these three sets, we search the historical headlines of the same author from the headlines in the training set, which avoids the answers leakage of the validation set and the test set. In this paper, we do not consider the time when the article was published, so the historical headlines are all headlines within two months from the same author, excluding the target headline.

4.2 Baselines

We select two competitive models as our basic baseline models, namely general HG and merge H3G.

- General HG model uses transformer architecture and BERT-base to initialize the encoder parameters as our CLH3G. Different from our CLH3G model, the general HG model only use the original article to generate the corresponding headline. The general HG model is used to verify the effectiveness of historical headlines for headline generation.

- Compared with General HG model, the merge H3G model concatenates historical headlines and the article as the input of encoder, which is a very simple method to utilize the historical headlines and can be used to verify the effectiveness of our proposed CLH3G model.

Besides, we also implement two strong baseline models, namely AddFuse HG, StackFuse HG from (Liu et al., 2020).

- The AddFuse HG model concatenates all historical headlines into a sentence as the input of the encoder to get $H_{headlines}$. $H_{headline}$ and H_A are used to compute headline-filtered article H_{FA} through the multi-head self-attention sub-layer. Finally, the target headline is generated by H_{FA} instead of H_A.

- Based on the AddFuse HG model, the StackFuse HG model performs a multi-head attention on H_{FA} and H_A one by one in each block of the decoder, so each decoder stack is composed of four sub-layers.

4.3 Implementation and Hyperparameters

We set the maximum article length and target headline length as 512 and 32 for all models. The length of concatenated headlines in the AddFuse H3G model and the StackFush H3G model is 256. The length of each historical headline in the CLH3G model is 32. In order to be consistent with the real online applications, the number of historical headlines is random chosen from 1 to $min(K, \#(articles\ of\ the\ author) - 1)$, where K is a hyperparameter. The encoder and the decoder of all transformer-based models have the same architecture hyperparameters as BERT-base. The parameters of all models are trained by Adafactor optimizer (Shazeer and Stern, 2018), which can save the storage and converge faster. At the same time, we set batch size and dropout of all models to 96 and 0.1, respectively. We train all the models 50K steps and then test on the validation set every 500 steps. We finally report the results of the test set in the best step of the validation set. During

Model	Rouge-1	ROUGE-2	ROUGE-L	BLEU
General HG	42.39	29.29	40.60	22.48
Merge H3G	42.48	29.29	40.42	22.81
AddFuse H3G	43.64	30.28	41.59	23.69
StackFuse H3G	43.80	30.43	41.77	23.76
CLH3G	**44.15**	**30.77**	**42.12**	**24.13**

Table 2: Rouge and BLEU scores of different Headline Generation Models.

inference, we also use beam search with length penalty to generate more fluent headlines. We set the beam size and length penalty of all models to 4 and 1.5, respectively.

4.4 Experimental Results on ROUGE and BLEU

We use ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002) as metrics to automatically evaluate the quality of generated headlines of all baseline models and our CLH3G model. For all historical headlines based HG models, we set the maximum number of historical headlines K to 10, and the λ of our CLH3G model to 0.1. As shown in Table 2, most of the H3G models are better than the general HG model, which demonstrates the effectiveness of historical headlines in headlines generation. The merge H3G model has some similar results with the general HG model, because we only truncate the article to keep the input length of the merge H3G model as 512, and the long historical headlines causes the loss of the article information. Besides, the two strong baseline models AddFuse HG model and StackFuse HG model achieve excellent results compared with the general HG model and the merge H3G model. There are two main reasons for this: (1) these two models can obtain additional historical headlines information than the general HG model; (2) compared with the merge H3G model, the information of the original article will not be lost when using historical headlines. Compared with the AddFuse H3G model, the StackFuse H3G model uses the original article representations H_A incrementally and perform better results. Finally, our CLH3G model achieves the best results for all metrics, which demonstrates our CLH3G can extract and utilize the information of historical headlines effectively compared with other baseline models. Besides, the complexity per layer of the self-attention model is $O(n^2 \cdot d)$, and our CLH3G model represents all historical headlines one by one, while the AddFuse H3G model and the StackFuse model represents all concatenated historical headlines at the same time, so our CLH3G model is more efficient than AddFuse H3G model and StackFuse model.

4.5 Experiment results on headline style

To study the style relationship between the generated headlines and the historical headlines, we use BERT-base and Contrastive Learning to train a classifier to distinguish headlines from different authors. The setting of the classification model is same as the contrastive learning based auxiliary task in our CLH3G model. The samples in the training set is a set of headlines of the same author, and we randomly select two of them as the positive samples to train the contrastive learning classifier. The two headlines of the negative sample in the validation set and the test set are randomly selected from different authors. We train the contrastive learning based classifier for 50K steps and obtain the best model in the validation set according to accuracy. The contrastive learning classifier will output a score within $[-1, 1]$, and the higher the score is, the greater the possibility that the two samples belong to the same author. We make the generated headline and all the historical headlines to build the evaluation samples one by one and report the accuracy and the average classification score. We name the original author headline and the historical headlines as Reference, which will get the highest accuracy and average score in theory.

The classification accuracy and the average scores are shown in Table 3. The Reference gets the highest accuracy and average score compared with other HG and H3G models. The general HG model obtains the worst accuracy and average score, which is consistent with its performance on ROUGE and BLEU, because it can only generate headlines aimlessly without the information of historical headlines. We notice that the merge H3G model achieves the best accuracy and average score besides Reference. This may be because the merge H3G model exploits the whole historical headlines and a small portion of the article to generate a new headline, and the missing information of the article makes the model relies more on historical headlines. Compared with the general HG model, the AddFuse H3G model and the StackFuse H3G model get better results, which is also consistent with its performance on ROUGE and BLEU. Our CLH3G model get approximate results compared with the merge H3G model, and is better than

Model	Accuracy	Average Score
General HG	86.09	48.61
Merge H3G	90.18	54.76
AddFuse H3G	89.93	52.51
StackFuse H3G	89.41	52.26
CLH3G	90.13	54.10
Reference	**92.22**	**58.21**

Table 3: The contrastive learning classification results and Human Evaluation results of different Headline Generation Models.

the AddFuse H3G model and the StackFuse H3G model. The results of the classification accuracy and average score can reflect the effectiveness of using historical headlines. Our contrastive learning module and two headline vector fusion methods are both beneficial to learn the style of historical headlines, resulting better accuracy and average score. The best results on ROUGE and BLEU of our CLH3G model prove that our CLH3G model can utilize and fuse the article and the historical headlines effectively at the same time.

4.6 Human Evaluation

Besides, we also apply Human Evaluation to verify the generated headine style. We randomly sampled 50 news from the test set and asked three annotators to rerank the five generated headline and the reference headline, while the ranked first get 6 points, and the ranked last get 1 point. Besides, The similar headlines will get the same ranked points, resulting the relatively high scores for all models. We use three criteria namely fluency, relevance and attraction as (Jin et al., 2020).

The results is shown in the Table 4. Similarly with the results on Rouge, BLEU and the CL based classification, the general HG get the worst results, and the reference get the best results. The historical headlines based models get significantly better results than the general HG model on fluency and relevance. The historical headlines can guide the generation of target headline syntax, resulting better fluency. Meanwhile, the better relevance is because these historical headlines based models have less factual consistency errors than the general HG model. Finally, our CLH3G model get the best results on all three aspects except the Reference headlines.

Model	Fluency	Relevance	Attraction
General HG	4.44	4.84	5.14
Merge H3G	5.18	5.14	4.96
AddFuse H3G	4.74	5.16	**5.28**
StackFuse H3G	4.82	5.14	5.26
CLH3G	**5.2**	**5.18**	**5.28**
Reference	**5.8**	**5.52**	**5.78**

Table 4: Human evaluation results on fluency, relevance and attraction.

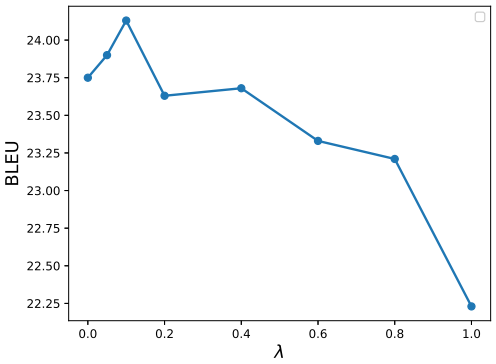

Figure 3: Results of CLH3G with different values of Contrastive Learning coefficient λ.

4.7 Experimental Results with different values of hyperparameter λ

In order to study the influence of contrastive learning module, we train our CLH3G model with different contrastive learning coefficient λ. We report the BLEU results of the experiments in Figure 3. The different values of λ have a great impact on the final results of our CLH3G model, and a clear conclusion can be drawn from the results. When λ is smaller than 0.1, the larger of λ, the better of the performance. And when λ is bigger than 0.1, the smaller of λ, the better of the performance. The best result is achieved when λ is 0.1. We will analyze the reasons of this experiment results. Firstly, when λ is very small, contrastive learning module has little positive impact on the whole model, so that the results are getting better. Then, with the increasing of λ, contrastive learning module has too much impact on the whole model, which disturbs the training of headline generation, so the results are getting worse.

4.8 Incremental Experiments

To further demonstrate the effectiveness of contractive learning module and the two headline vector fusion methods in our CLH3G model, we conduct incremental experiments and report the results in Table 4. As shown in Table 6, the concat and

Historical Headlines: 1. Digital currency has been accepted by the world, and the Chinese market is about to explode! (数字货币已被世界公认，中国市场即将爆发！) 2. The central bank takes strong action, and RMB is about to rise! (人民币崛起 央行强势出手！) 3. Digital RMB develops from "point" to "surface", and has entered thousands of families! (数字人民币由“点”到“面”，走进千家万户！)
Reference headline: The central bank is launching digital RMB, indicating the full outbreak of digital currency! (央行数字人民币落地，迎来数字货币全面爆发！)
Generated headline by general HG model: Daofu Chen, Financial Research Institute of the State Council: China should give better play to the cross-border payment system based on blockchain (国金融研所陈道富：中国宜更好发挥基于区块链等的跨境支付体系的作用)
Generated headline by CLH3G model: The central bank is launching digital RMB, striving to become the first player in the field! (央行数字人民币即将落地，争做第一个吃螃蟹的人！)

Table 5: An Example of generated headlines through general HG model and CLH3G model.

Model	Rouge-1	Rouge-2	Rouge-L	BLEU
General HG	42.39	29.29	40.60	22.48
+ Concat Style Vector	43.63	30.31	41.79	23.90
+ Concat Style Vector + CL	43.66	30.39	41.45	23.74
+ Pointer Style Vector	43.09	29.93	41.25	23.20
+ Pointer Style Vector + CL	43.55	30.24	41.82	23.35
+ Two fusion Methods	43.37	30.32	41.58	23.75
+ Two fusion Methods + CL	**44.15**	**30.77**	**42.12**	**24.13**

Table 6: Incremental Experiment of CLH3G model.

pointer headlines style vector fusion methods both can improve the performance of Headline Generation, because they use additional historical headlines. In addition, the concat fusion method can get better results compared with the pointer fusion method, which proves that informing the decoder the desired headline style in advance is more effective than guiding the choice of words in pointer network. It may also be due to that there are few overlapping words between different headlines of the same author, and their headline patterns and style are consistent instead. We also add contrastive learning based auxiliary task to the concat fusion method and the pointer fusion method, respectively. The performance of the concat fusion method using CL based auxiliary task is slightly improved in ROUGE-1 and ROUGE-2, while the performance in ROUGE-L and BLEU is reduced, which shows that CL has little effect on the concat fusion method. The pointer fusion method with the CL based auxiliary task greatly improves the effectiveness of all metrics, which proves that the pointer is more dependent on the headline style. Furthermore, when we use the two fusion methods at the same time, the results is somewhere in between using a single method, because the relatively worse headline vector will mislead the choice of words in the pointer. For our CLH3G model, the better headline vector leads to the best results, which demonstrates our contrastive learning module can extract better headline style vector for H3G models.

4.9 Case Study

We display an example of generated headlines by general HG model and CLH3G model in Table 5. Both the historical headlines and the generated headline by CLH3G model are exclamatory sentences. Besides, the generated headline by CLH3G is more informative and attractive than the generated headline by general HG model.

5 Conclusion

In this paper, we discuss the effectiveness of Historical Headlines for Headline Generation, and aim to generate headlines not only appropriate for the given news articles, but consistent with the author's style. We build a large Historical Headlines based Headline Generation dataset, and propose a novel model CLH3G to integrate the historical headlines effectively, which contains a contrastive learning based auxiliary task and two headline style vector fusion methods. Experimental results show the effectiveness of historical headlines for headline generation and the exceptional performance of both the CL based auxiliary task and the two headline style vector fusion methods of our CLH3G model.

Limitations

This paper introduces a new headline generation task, which use historical headlines to generate article headlines, and also proposes a novel model called CLH3G for this task. CLH3G uses two headline style vector fusion methods to make full use of historical headlines. However, those two style vector fusion methods is difficult to applied into pretrained Sequence to Sequence model including T5, Mass (Song et al., 2019; Raffel et al., 2019) directly, because those two methods will change the whole architecture of pretrained model, resulting slightly worse results compared with original pretrained model. As result, the integration of CLH3G and pretrained Sequence to Sequence models requires abundant H3G data to achieve comparable results.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments.

References

Xiang Ao, Xiting Wang, Ling Luo, Ying Qiao, Qing He, and Xing Xie. 2021. Pens: A dataset and generic framework for personalized news headline generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 82–92.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Hengyi Cai, Hongshen Chen, Yonghao Song, Zhuoye Ding, Yongjun Bao, Weipeng Yan, and Xiaofang Zhao. 2020. Group-wise contrastive learning for neural dialogue generation. *arXiv preprint arXiv:2009.07543*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

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*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. Technical report, MARYLAND UNIV COLLEGE PARK INST FOR ADVANCED COMPUTER STUDIES.

Daniil Gavrilov, Pavel Kalaidin, and Valentin Malykh. 2019. Self-attentive model for headline generation. In *European Conference on Information Retrieval*, pages 87–93. Springer.

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*.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. 2020. Contrastive learning with adversarial perturbations for conditional text generation. *arXiv preprint arXiv:2012.07280*.

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

Dayiheng Liu, Yeyun Gong, Jie Fu, Wei Liu, Yu Yan, Bo Shao, Daxin Jiang, Jiancheng Lv, and Nan Duan. 2020. Diverse, controllable, and keyphrase-aware: A corpus and method for news multi-headline generation. *arXiv preprint arXiv:2004.03875*.

Yixin Liu and Pengfei Liu. 2021. Simcls: A simple framework for contrastive learning of abstractive summarization. *arXiv preprint arXiv:2106.01890*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. From neural sentence summarization to headline generation: A coarse-to-fine approach. In *IJCAI*, volume 17, pages 4109–4115.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Peng Xu, Chien-Sheng Wu, Andrea Madotto, and Pascale Fung. 2019. Clickbait? sensational headline generation with auto-tuned reinforcement learning. *arXiv preprint arXiv:1909.03582*.

An Yan, Zexue He, Xing Lu, Jiang Du, Eric Chang, Amilcare Gentili, Julian McAuley, and Chun-Nan Hsu. 2021. Weakly supervised contrastive learning for chest x-ray report generation. *arXiv preprint arXiv:2109.12242*.

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, Huanhuan Cao, and Xueqi Cheng. 2018. Question headline generation for news articles. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 617–626.