A Deep News Headline Generation Model with REINFORCE Filter

Aobo Xu
School of Economics and Management
China University of Petroleum
Qingdao, China
upc20xab@s.upc.edu.cn

Ling Jian*
School of Economics and Management
China University of Petroleum
Qingdao, China
bebetter@upc.edu.cn

2023 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-8867-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/IJCNN54540.2023.10192007

***Abstract*—Generating accurate and concise headlines based on news content can help people filter out interesting content and improve the quality of life. News headline generation, as the sub-application area of text summarization, has many methodological commonalities but with higher requirements for generated text quality, which is more challenging. In this paper, we propose a new model to generate news headline, named RADGen (REINFORCE Aided Deep Generator for news headline), which combines a Transformer-based generator with a sentence-selecting filter based on REINFORCE algorithm. We perform experiments and assessments on Chinese news headline dataset, which achieve 25.71, 8.26 and 23.58 for f-score of ROUGE-1, ROUGE-2 and ROUGE-L respectively, to demonstrate the effectiveness of the proposed model. In addition, ablation experiments show the positive roll of reinforcement learning filter in news headline generation.**

I. Introduction

The dissemination of information through various media outlets, including social platforms and traditional news sources, results in a vast amount of daily news content being made available to the public. A brief but meaningful headline can help people filter out irrelevant information. Therefore, the generation of precise and informative headlines has the potential to enhance the quality of life for news readers. News headline generation, a typical application scenario of natural language processing (NLP), is a process of generating a concise sentence, which maintains the original meaning, for a given news text. It belongs to the sub-application area of summarization technology, where machine learning methods are widely used in. Text summarization techniques can be broadly categorized into two distinct paradigms: extractive [1] and abstractive [2] methods. The first approach involves identifying and selecting specific sentences or words from the original text as output, while the latter approach utilizes deep learning techniques and neural networks to understand the document and generate new phrases.

The generation of headlines, which are typically shorter in length than summaries, requires a more sophisticated model to effectively condense the information and capture the essential elements of the text. Three challenges hamper the news headline generation task, which are deemed necessary to be addressed in this paper. (i) One of the main difficulties is the representation of long documents for which the headline generation task is prone to low accuracy, including the presence of repetitive and inconsistent phrases. (ii) Additionally, the phenomenon known as "exposure bias" occurs in generation tasks, where ground truth is provided during the training phase but not during the testing phase. (iii) Another challenge frequently occurred in the scenarios of both summary and headline generation is the discrepancy between the objective function used during training and the evaluation metrics used to assess performance. Reinforcement learning has advantage in addressing the non-differentiable nature of evaluation metrics, by treating metrics as a part of reward for the agent [3]. By incorporating reinforcement learning, it is possible to mitigate the aforementioned challenges in the task of news headline generation.

Aiming to address the three challenges previously outlined, the proposed model has been designed with a focus on the following key aspects. For the first challenge, a reinforcement learning approach is employed to filter unimportant sentences and Transformer is utilized to extract features from long documents. We design reward based on ROGUE metrics, thereby the model tends to generate more accurate headline because repetition and incoherence of phrases often result in lower reward. For the second challenge, ground truth is only used for the calculation of reward during the training of reinforcement learning agent, thus alleviating the issue of "exposure bias" in the training process. For the last challenge, by designing reward according to metrics, the objective function and evaluation metrics can be aligned to some extent.

For the task of *News Headline Generation*, in this paper, we propose a novel deep reinforcement learning model, named RADGen (REINFORCE Aided Deep Generator for news headline). The model is combined with Transformer architecture to generate news headline and based on the classic policy gradient REINFORCE algorithm. The major contributions of this paper are as follows:

- We propose a new sentence-filtered model based on deep reinforcement learning algorithm, which aims to select importance sentences from long raw document inputs. And we combine it with a Transformer-based generator

This research was funded by the National Key R&D Program of China (No. 2021YFA1000100, 2021YFA1000102)
* Corresponding author

to generate news headlines.

- We provide empirical evidence to show the effectiveness of the proposed model on a Chinese news dataset. Our model achieves satisfied performance on ROUGE metrics compared with some classic models.
- We collected Chinese mainstream media news and selected high-quality news to construct the corpus, which has been released for the academic research. The news data is available at https://github.com/MyLove-XAB/RADGen.

II. RELATED WORK

A. Summarization Methods

Summarization is a task of condensing the core idea for a certain text. Most early research focused on extractive methods. Ranking words or sentences with scores and piecing them together into semantically complete summary is the common style. Some tend to calculate the importance of different words according to specially designed rules, such as term frequency–inverse document frequency (tf-idf) [4] and TextRank [5]. Some statistical models, such as Bayesian and Hidden Markov Model (HMM), are also applied to extract summary [6] [7].

With advancements in deep learning, end-to-end resolution of NLP tasks through the utilization of word embedding methods has become increasingly widespread [8]. Encoder-decoder structure [9] is the most widely used architecture for abstractive summary generation. Seq2seq is an acclaimed model using encoder-decoder structure in NLP text-to-text tasks [10], including machine translation and text generation, which often use long-short term memory network (LSTM) [11] or other neural networks structure to achieve input encoding and output decoding. And attention mechanism performs great power in the field of NLP, which also has explainable effect [12] [13]. Transformer [14], BERT [15], GPT [16] and their variants achieve great improvement in generation tasks. Pointer network [17] propose a coverage loss and is also widely used for text-to-text applications [18]. Some recent works try to combine extractive methods with neural networks, for example by treating summarization as a classification task and select sentence by deep neural network [19].

Reinforcement learning methods are recently used in NLP tasks. It has advantages in solving non-differentiable metrics, which can be commonly seen in NLP, especially in generation tasks, such as dialogue [20] [21] and summary [3]. Policy-gradient methods are mostly used in text generation tasks, which can easily connect the action space with the tokens predicted. In this case, reinforcement learning can be injected into corresponding tasks by focusing on the word-level [22] or sentence-level [23] text data.

B. Headline Generation Methods

Headline generation is a very popular area of NLP research, which can be seen as a summarization task within two sentences. Therefore, lots of summrization methods can be used in this area. Some headline generation models are still based on extractive methods [24], or view headline generation as a problem analogous to statistical machine translation [25]. Mostly use deep learning methods, generate headline abstractly [26], which apply summarization techniques into headline generation [27] [28]. The combination of extractive and abstractive also seems useful [29].

As mentioned before, reinforcement learning methods also have advantages while solving headline generation tasks. To integrate with reinforcement learning, some works try to construct the reward considering repetition [30]. In addition to generate brief news headline, PORLHG tried to considering the popularity of result [31]. Some even consider more during choosing actions, such as topic awareness [32], controlling style [33], attractiveness or faithfulness [31] and so on [34] [35]. Others try to apply headline generation into more practical application scenarios such as retail content [36]. In the field of headline generation, there still exists researching space by combining with reinforcement learning. Previous studies mostly focus on the final stage of word generation, whereas our approach also considers the model's input stage.

III. METHODOLOGY

The proposed RADGen model consists of two components: a Transformer headline generator and a reinforcement learning filter. The Transformer headline generator receive news contents as input and output the corresponding headlines. The reinforcement learning filter, on the other hand, makes decisions regarding which sentence should be filtered out and feeds the remained sentences into the well-trained Transformer headline generator to get a better headline. A brief structure of the proposed model to generate news headline is depicted as Fig. 1.

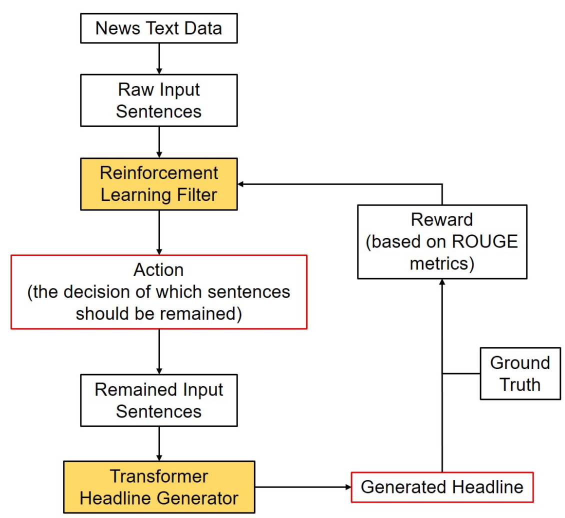

Fig. 1. Brief illustration of the proposed RADGen model. Both reinforcement learning filter and Transformer headline generator consist of neural network, where action and generated headline are the corresponding outputs.

A. Transformer Headline Generator

The proposed news headline generator takes news text as input and produces the corresponding headline as output. The generator adopts an encoder-decoder architecture as its main structure. To enhance the abstractive capabilities, the Transformer architecture is utilized as the key module for feature extraction. The Transformer architecture, which constitutes the core of the generator, employs a self-attention mechanism that calculates scores for each input word to indicate the variations in emphasis among the words. This enables the generator to extract and emphasize the most relevant information from the input text, resulting in the generation of a more informative and accurate headline.

The text representation for the ith input document, donated as doc_i, is achieved by concatenating each padded sentence, denoted as s_j, for $j = 0, 1, 2, ..., n$, where n is the number of sentences. For the input document doc_i, Transformer utilizes the multi-head attention (MHA) mechanism to calculate the attention scores as in [14]:

$$doc_i = concat([s_1, s_2, ..., s_j]) \tag{1}$$

$$atte(doc_i) = softmax(\frac{W^Q doc_i (W^K doc_i)^T}{\sqrt{d_k}}) W^V doc_i \tag{2}$$

$$MHA(doc_i) = concat(atte_1(doc_i), ..., atte_h(doc_i)) W^O \tag{3}$$

where $W^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}, W^K \in \mathbb{R}^{d_{\text{model}} \times d_k}, W^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ are the parameter matrix, h is the number of heads. The result is then connected with two linear transforms w.r.t feed forward network with ReLU as the activation function:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{4}$$

where W_1, W_2, b_1, b_2 are the weights and bias parameters of the two layers. Besides, the positional embedding, residual connection mechanism and layer normalization used in [14] are kept in our model.

The objective function aims to maximize the maximum-likelihood of the generated headline by minimizing the cross entropy loss at each decoding step, donate as L_{tr}:

$$L_{tr} = -\sum_{k=1}^{T} logp_\theta(y_k^* | y_1^*, y_2^*, ..., y_{k-1}^*, doc) \tag{5}$$

where $y^* = (y_1^*, y_2^*, ..., y_T^*)$ is the ground truth for input news. After the training process, the network parameters are saved and the well-trained generator is utilized as part of interactive environment for the subsequent implementation of reinforcement learning, which will be introducted in more detail in the following sections.

If we represent Transformer headline generator as THG, the whole process of predicting headline hl_i for news text doc_i can be expressed as following:

$$hl_i = THG(doc_i). \tag{6}$$

B. Reinforcement Learning Filter

It is a common occurrence for news text to contain sentences that are not deemed important. Inspired by the phenomenon, we design a sentence filter based on reinforcement learning to selectively retain or discard sentences, which indirectly impacts the final output generated by the Transformer-based generator. The design of state, action, policy, environment and reward, which are critical factors that impact the performance of the model, are discussed in the following.

State. The input for the sentence filter is the same document representation as the Transformer headline generator, as shown in (1). The representation is then embedded and fed into the policy network as the basis for action decision-making.

Action. The agent is tasked with making a binary decision for each sentence in the input document, where a decision of 1 corresponds to retaining the sentence as input for the generator and a decision of 0 corresponds to discarding the sentence. This decision-making process is formulated as a sequential binary classification task, which generates a series of binary decisions for news sentences sequentially. Therefore, the action space for an news text can be donated as $A = [a_1, a_2, ..., a_n], a_j \in \{0, 1\}$ for j in $\{0, 1, ..., n\}$.

Policy. The policy network, implemented as a deep neural network with LSTM as the hidden layer, followed by multi-layer perception (MLP), extracts features from the input sentences and outputs the probability of each sentence retained, as shown in (7) - (9). Sequentially output the decision for each sentence seems low efficient. Therefore, the policy network is designed to make simultaneous decisions for all sentences in a news article, which is achieved by setting the final layer with the same number units as the number of input sentences.

In our case, we assign sigmoid as the activation function for the final layer of the policy network, calculated as (9). Different from the commonly-used softmax function, sigmoid ensures that each output unit represents an independent retaining probability for the corresponding sentence position without the constraint of the sum of outputs equaling to 1. By deploying sigmoid activation function, we are able to convert the decision-making problem from a sequential binary-classification task to a multi-label classification task. It means that the number of retained sentences is uncertain and the actions can be sampled separately according to the probability p. Formally, the operational process of policy network is shown as follows:

$$x = LSTM(doc_i) \tag{7}$$

$$x = MLP(x) \tag{8}$$

$$p = sigmoid(x) = 1/(1 + e^{-x}) \tag{9}$$

Environment. We incorporate a well-trained transformer headline generator as a component of the environment, which is able to generate headlines in response to the agent's actions. Furthermore, we add a mechanism for computing reward as a feedback signal for different actions. How to calculate reward will be introduced later. With the intention of mitigating the problem of "exposure bias", the entire headlines are generated without utilizing the ground truth during this process.

Reward. ROUGE metrics count the number of overlapping units such as n-gram between the generated outputs and the reference headlines. We design the reward based on ROUGE metrics, by comparing the generated headline with the ground truth, donated as R_{rl},

$$R_{rl} = a * r_1 f + b * r_2 f + c * r_l f \tag{10}$$

where $r_1 f$, $r_2 f$, $r_l f$ are the f-score that calculated according to ROUGE-1, ROUGE-2 and ROUGE-L score as in [37], and a, b, $c \in [0, 1]$ are hyper parameters to show emphasis on different metrics.

In order to increase the robustness and accelerate the convergence of the algorithm, we incorporate a fixed baseline which is independent of the action taken [38]. The baseline, donated as R_{tr}, is calculated by utilizing the reward provided by the well-trained generator, which can be easily obtained prior to training the reinforcement learning filter. The difference between R_{rl} and R_{tr} serves as an indicator of the effectiveness of the agent's decision-making process.

Loss function. For the multi-label classification task, binary cross-entropy loss (BCE loss) is commonly used. The objective function for reinforcement learning filter can be designed by integrating BCE loss with reward, donated as L_{rl}.

$$BCE = -1/n \sum_{i}^{n} (t[i] * \log(o[i]) + (1 - t[i]) * \log(1 - o[i])) \tag{11}$$

$$L_{rl} = (R_{rl} - R_{tr}) * BCE \tag{12}$$

where o, t represent the outputs and targets.

According to the previous design, the reward R_{rl} is obtained at the moment of the whole headline being generated, satisfying the condition of REINFORCE algorithm [39]. We combine REINFORCE, a policy gradient algorithm, with deep learning to build our sentence-selecting agent. The policy network will be updated through the policy gradient, i.e.,

$$\theta = \theta + \alpha \left(R_{rl} - R_{tr}\right) \nabla_\theta \log \pi \left(A \mid doc, \theta\right). \tag{13}$$

C. Training Process

For the model training, we design a two-stage training process, depicted as Fig. 2. In the first stage, we adapt the teacher-forcing algorithm [40] to train the generator, by minimizing the cross entropy loss L_{tr}. After training, the generator is saved and used to calculate the reward baseline. The second stage leverages the well-trained Transformer headline generator from stage one and minimizes the loss function L_{rl} to train the reinforcement learning filter. The performance of the model is evaluated on a validation set. It is noteworthy that, aside from being used in the first stage to compute the cross-entropy loss, the ground truth is only used to calculate reward. The pseudo algorithm is presented in **Algorithm 1**.

IV. EXPERIMENTS

In this section, we conduct empirical experiments on Chinese news headline dataset, aiming to answer the following two research questions:

- Q1: Does the proposed RADGen model can effectively generate news headlines?
- Q2: Does reinforcement learning filter perform a positive role in the task?

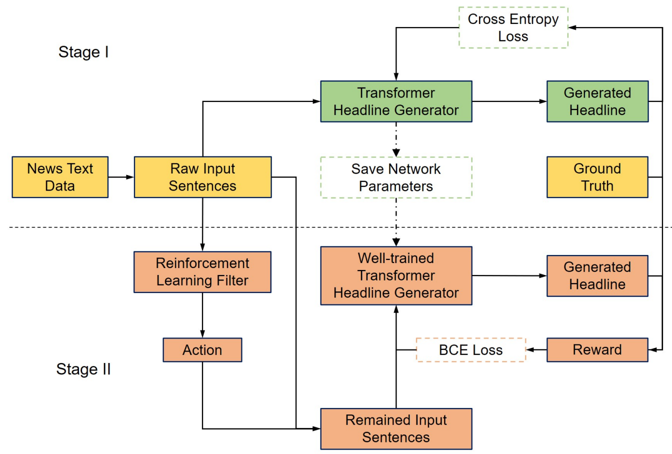

Fig. 2. Two-stage training pipeline. The first stage train a Transformer headline generator with loss function L_{tr}. The second stage utilizes REINFORCE algorithm to train a reinforcement learning filter with loss function L_{rl}.

Algorithm 1 Training Process of REINFORCE Aided Deep Generator for News Headline

Input: Preprocessed news documents doc

Parameters: Initialize θ_{tr} for Transformer headline generator, θ_{rl} of policy network, and set training epochs N_{ep}, batch size m, repeat times N_{rep}

Stage I:

1: Train generator with loss function (5) and update θ_{tr}
2: Save θ_{tr} and generate headlines hl with (6) for training news data
3: Calculate the baseline according to (10) for news data

Stage II:

4: **for** $i = 0$ to N_{ep} **do**
5: Sample m news data from training set
6: Get actions with policy network
7: Filter sentences according to the actions
8: Input news documents to generate headlines hl with well-trained generator θ_{tr}
9: Calculate reward for hl according to (10)
10: Train reinforcement learning filter with loss function as (12) and update θ_r according to (13)
11: Repeat 5 to 10 for N_{rep} times
12: Evaluate the model on the validation set
13: **end for**

A. Chinese News Headline Dataset

The dataset composed of news articles and their associated headlines if gathered from various prevalent Chinese news websites. Alongside the text data, we also obtain the number of readings, comments, likes and shares for each news article as supplementary information. In order to obtain a practical model for real-world applications, we emphasize the importance of utilizing high-quality data. To this end, we propose a metric that allows us to select high-quality news based on the number of comments and likes, as a way to ensure that the data used for training is of high quality. The quality score for a certain news document doc_i, donated as $score_i$, can be calculated as:

$$score_i = \lambda_r r_i^{norm} + \lambda_c c_i^{norm} + \lambda_l l_i^{norm} + \lambda_s s_i^{norm} \quad (14)$$

where $r_i^{norm}, c_i^{norm}, l_i^{norm} and s_i^{norm}$ is the normalization result of the number of readings, comments, likes and shares, for example, $r_i^{norm} = \frac{r_i - r_{\min}}{r_{\max} - r_{\min}}$, and $\lambda_r, \lambda_c, \lambda_l, \lambda_s$ are hyper parameters to balance the corresponding weights.

After ranking the data by the quality score, we collect 178942 news articles totally and make the data available to the public. Through random shuffling, the data is divided into three sets: the training set with a size of 163137, the validation set with a size of 7093, and the test set with a size of 7093. After preprocessing the collected data, we present all the statistic features in Table I. It can be observed that the length of headline is significantly shorter in comparison to the length of the news content. Consequently, a small increase in the number of correctly predicted words can lead a significant improvement in the metrics.

TABLE I
DATA STATISTICS OF CHINESE NEWS DATASET

data	avg.sentence	avg.token
text	19.70	494.41
headline	1.46	14.96

Note: avg.sentence is the average number of sentences for news text or headlines; avg.token is the average number of tokens for news text or headlines after preprocessing.

B. Details

For the experiments, the word-based approach is adopted for the reason that in the case of Chinese news, words contain more relevant to latent knowledge of documents than Chinese characters do. During the data preprocessing phase, we perform tokenization on the input documents, followed by the removal of stop words and punctuations. This helps to create a more refined and structured representation of the text data, enabling more effective analysis and processing. As mentioned before, we encode each sentences and then concatenate them, which make it more convenient for sentence selecting.

For the training of the Transformer headline generator, batch updates are performed by traversing the training set. The baseline is computed only once using the well-trained generator. During training the reinforcement learning filter, similar batch update method is utilized, in which a set of b_{rl} news articles are randomly selected to update the policy network. This process is repeated N_{rep} times for each training step, which is analogous to PPO algorithm training style [41]. Ultimately, the model is chosen based on its performance on validation set. The whole training process can be sped up through parallel computation. Some hyper parameters setting is presented in Table II.

TABLE II
SOME PARAMETERS' SETTING

Symbol	Value	Description
b_{tr}	256	Batch size of Transformer generator
b_{rl}	256	Batch size of reinforcement learning filter
n_{layer}	2	Number of Transformer layers
n_{head}	4	Number of heads in Transformer
$h_L STM$	512	Hidden units of LSTM layer
N_{rep}	5	Repeat times for each step
l	960	Padded length of each news document

C. Results

In the experimental evaluation, we apply RADGen model to generate headlines according to the news text in Chinese news headline dataset, and present the performance using the f-score metrics for ROUGE-1, ROUGE-2, and ROUGE-L. Furthermore, we compare the performance of our model with that of classic headline generation models through comparison and ablation experiments.

Comparison experiments. To answer Q1, we carries out a comparative analysis of our model (RADGen) with other models in the field of headline generation. This involves a comparison with three different models: (i) a seq2seq model that utilizes LSTM neural networks as the hidden layer, (ii) an extractive model that selects a sentence randomly from the news text to form the headline and (iii) a TextRank model that utilizing TextRank algorithm to identify the most important sentence as headline.

As presented in Table III, the experimental results indicate that the proposed model outperforms other models on ROUGE-1 and ROUGE-L metrics and achieves comparable results on ROUGE-2.

TABLE III
COMPARISON EXPERIMENTS RESULTS

model	ROUGE-1	ROUGE-2	ROUGE-L
LSTM seq2seq	7.47	0.68	6.88
extractive model	12.33	4.63	11.19
TextRank model	15.32	**8.58**	15.57
RADGen model	**25.71**	8.26	**23.58**

Ablation experiments. To answer Q2, we also conduct ablation experiments to show the effect of reinforcement learning filter. Specifically, we compare the performance of the model with and without filter, in order to figure out the extent to which the filter contributes to the overall performance of the model. The result, as presented in Fig. 3, shows that each metric has a positive improvement, by using the news input processed by the filter.

Generation results. The performance during the training process, as evaluated on the validation set, is illustrated in Fig. 4. While evaluating the performance, we compare the model in two different ways of choosing actions. One approach is sampling action by the probabilities generated from the policy network. The other one involves deterministically selecting the action with the highest probability for each sentence,

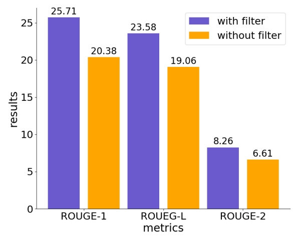

Fig. 3. Ablation experiment results. The model with filter performs better on all three metrics.

referred to as greedy action selection. The result shows that both approaches achieve comparable performance, while the latter one appears to be more stable. In Table IV, we present a comparison of generated news headline compared with their respective ground truth. The proposed RADGen model appears to have a successful impact on enhancing the quality of news headlines, despite some inaccuracies in the generation of numeric character strings.

To provide further insights into the proposed model, we record the actions chosen by the reinforcement learning agent, serving as an indication of the retention of news data, as Fig. 5. The results from the RADGen model suggest that it places a higher weight on the retention of earlier sentences in the news data, as evident from the actions chosen by the reinforcement learning agent. This aligns with the typical news writing convention of presenting the main idea in the beginning and providing additional details later on.

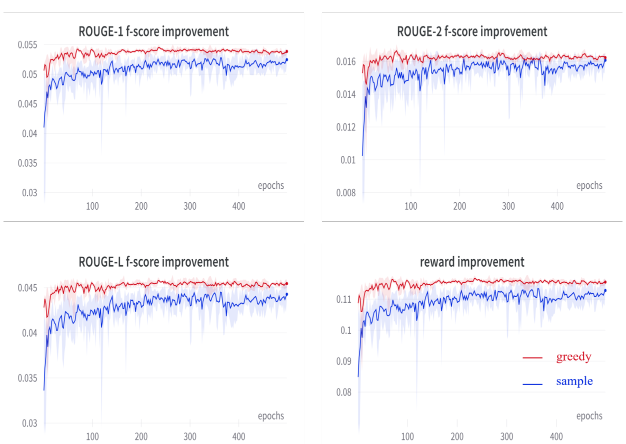

Fig. 4. Performance on the validation set. The figure shows the difference between 4 metrics and their baseline and is presented by the average performance across 5 replicated runs.

V. CONCLUSION

In this study, we present a REINFORCE Aided Deep Generator for news headline through the integration of a deep encoder-decoder generator and reinforcement learning techniques. The proposed RADGen model utilizes Transformer as the headline generator which also serves as the environment of the reinforcement learning, and employs a policy network

TABLE IV
RADGEN MODEL GENERATION RESULTS

news document	ground truth	model result
21只个股今日获机构买入型评级……为机构首次关注，涉及会畅通讯、江阴银行等6只个股。	中国平安、中公教育等股今日获机构买入型评级	中公教育等股今日获机构买入型评级
据wind统计……拟减持金额约1.64亿元。新通联重要股东拟减持金额2976万元。	5家上市公司抛出逾11亿元股东减持计划	9家上市公司抛出逾11亿元股东减持计划
盘后数据显示，龙虎榜中……净流出金额分别是1.57亿元、5998万元、3800万元。	机构今日买入这14股，抛售紫光股份1.57亿元	机构今日买入这14股，抛售紫光股份1.14亿元

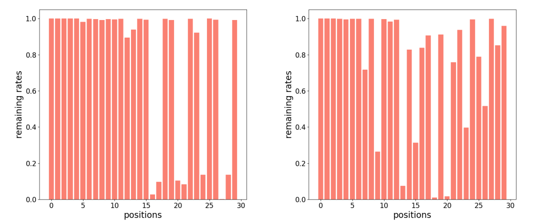

Fig. 5. Action distribution. The position distribution of the retained sentences by greedily choosing actions (left) and sampling actions according to the probabilities (right).

to select the most relevant sentences as input for the generator using the REINFORCE algorithm. Additionally, we construct a high-quality Chinese news headline dataset and release it for academic research. Experimental studies demonstrate the superiority of the proposed model via ROUGE metrics.

In terms of future research, focusing on the better exploration of action space could potentially yield promising results. And it may be worthwhile to investigate the integration of news headline generation with practical application tasks such as fake news detection.

REFERENCES

[1] B. Dorr, D. Zajic, and R. Schwartz, “Hedge trimmer: A parse-and-trim approach to headline generation,” in *North American Chapter of the Association for Computational Linguistics*, 2003.

[2] S. Chopra, M. Auli, and A. M. Rush, “Abstractive sentence summarization with attentive recurrent neural networks,” in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 93–98, 2016.

[3] R. Paulus, C. Xiong, and R. Socher, “A deep reinforced model for abstractive summarization,” in *International Conference on Learning Representations*, 2018.

[4] H. Christian, M. P. Agus, and D. Suhartono, “Single document automatic text summarization using term frequency-inverse document frequency (tf-idf),” *ComTech: Computer, Mathematics and Engineering Applications*, vol. 7, no. 4, pp. 285–294, 2016.

[5] R. Mihalcea and P. Tarau, “Textrank: Bringing order into text,” in *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411, 2004.

[6] J. Kupiec, J. Pedersen, and F. Chen, “A trainable document summarizer,” in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 68–73, 1995.

[7] J. M. Conroy and D. P. O'leary, "Text summarization via hidden markov models," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 406–407, 2001.

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Computer Science*, 2013.

[9] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *Computer Science*, 2014.

[10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Computer Science*, 2014.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[16] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

[17] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," *Advances in neural information processing systems*, vol. 28, 2015.

[18] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, 2017.

[19] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[20] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao, "Deep reinforcement learning for dialogue generation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1202, 2016.

[21] X. Li, Y.-N. Chen, L. Li, J. Gao, and A. Celikyilmaz, "End-to-end task-completion neural dialogue systems," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 733–743, 2017.

[22] T. Zhang, M. Huang, and L. Zhao, "Learning structured representation for text classification via reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[23] Y.-C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 675–686, 2018.

[24] S. Xu, S. Yang, and F. Lau, "Keyword extraction and headline generation using novel word features," in *Twenty-fourth Aaai Conference on Artificial Intelligence*, 2010.

[25] M. Banko, V. O. Mittal, and M. J. Witbrock, "Headline generation based on statistical translation," in *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 318–325, 2000.

[26] S.-Q. Shen, Y.-K. Lin, C.-C. Tu, Y. Zhao, Z.-Y. Liu, M.-S. Sun, *et al.*, "Recent advances on neural headline generation," *Journal of computer science and technology*, vol. 32, no. 4, pp. 768–784, 2017.

[27] J. Tan, X. Wan, and J. Xiao, "From neural sentence summarization to headline generation: A coarse-to-fine approach.," in *IJCAI*, vol. 17, pp. 4109–4115, 2017.

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

[29] Y.-C. Tseng, M.-H. Yang, Y.-C. Fan, W.-C. Peng, and C.-C. Hung, "Template-based headline generator for multiple documents," *IEEE Access*, vol. 10, pp. 46330–46341, 2022.

[30] P. Xu and P. Fung, "A novel repetition normalized adversarial reward for headline generation," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7325–7329, 2019.

[31] Y.-Z. Song, H.-H. Shuai, S.-L. Yeh, Y.-L. Wu, L.-W. Ku, and W.-C. Peng, "Attractive or faithful? popularity-reinforced learning for inspired headline generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 8910–8917, 2020.

[32] L. Wang, J. Yao, Y. Tao, L. Zhong, W. Liu, and Q. Du, "A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization," in *International Joint Conference on Artificial Intelligence*, 2018.

[33] D. Jin, Z. Jin, J. T. Zhou, L. Orii, and P. Szolovits, "Hooks in the headline: Learning to generate headlines with controlled styles," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5082–5093, 2020.

[34] P. Dognin, I. Padhi, I. Melnyk, and P. Das, "Regen: Reinforcement learning for text and knowledge base generation using pretrained language models," in *Conference on Empirical Methods in Natural Language Processing*, 2021.

[35] P. Xu, C.-S. Wu, A. Madotto, and P. Fung, "Clickbait? sensational headline generation with auto-tuned reinforcement learning," 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)*, pp. 3065–3075, 2019.

[36] Y. S. Kanungo, S. Negi, and A. Rajan, "Ad headline generation using self-critical masked language model," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pp. 263–271, 2021.

[37] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, pp. 74–81, 2004.

[38] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[39] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992.

[40] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[41] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR, abs/1707.06347*, 2017.