2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT) | 979-8-3503-9905-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICEICT57916.2023.10245936

Adaptive-Bert Network for Advertising Text Generation

1st Shuaiwei Yao
School of Information Engineering
China Jiliang University
Hangzhou, China
p21030854067@cjlu.edu.cn

2nd Xiaodong Pan
The 903rd Hospital of the People's Liberation Army of China
Hangzhou, China, 310013
35920117@qq.com

3rd Han Dong
Zhejiang Medical Products Service Center
Hangzhou, China
29400190@qq.com

4th Chunlei Kong
School of Information Engineering
China Jiliang University
Hangzhou, China
602349800@qq.com

5th Xiangping Wu*
School of Information Engineering
China Jiliang University
Hangzhou, China
xiangpingwu@cjlu.edu.cn

***Abstract*—Our system automatically generates tailored ad text based on the travel magazine advertisements layout and multiple travel blogs. The user inputs the layout of the travel magazine advertisement, which is then processed by a prediction-evaluation model (PEM). This model predicts the required number of text words and font size in the layout based on the input layout. Next, feed the data generated by the PEM and multiple tourism blogs into the text generation model, and ultimately produce text that meets the requirements and fits the designated text area in tourism magazines' advertisements. The generated text is then placed in the text area of the layout. It includes our new self-attention layer to ensure that the output text contains essential information. We conducted extensive experiments on real datasets to demonstrate the effectiveness of our proposed model. Using our model(Adaptive-Bert), users can create travel magazine advertising text that meets their requirements. Our method ensures that the generated text is of high quality and meets layout requirements.**

***Index Terms*—self-adaptive, summarizer, Bert**

I. Introduction

The generation of travel advertisement can be decomposed into two parts, the first part is to generate the layout of travel magazine, and the second part is to generate the text of travel magazine advertisement. In travel magazine advertisements, the text is very important, and the description of the text must be related to the scenic spots to be advertised in the magazine advertisement. We can summarize the text information in the travel blog and extract high-value information. To ensure the generated text is harmonious and aesthetically pleasing within the travel magazine advertisement layout, we first designed a prediction model to determine the number of words that can fit within a given font size. Then, using an iterative algorithm, we selected multiple groups of candidate words and font sizes, which were evaluated using our PEM. This model identified the candidate travel advertisement text with the best word count and font size, which was then pasted into the text area of the magazine advertisement layout. Finally, our text generation model produced the text that met all requirements.

*Corresponding author: Xiangping Wu, Associate Professor, School of Information Engineering, China Jiliang University, Hangzhou, China. E-mail: xiangpingwu@cjlu.edu.cn.

Overall, our enables users to quickly and easily create compelling travel magazine advertisements that are tailored to their specific layout and content requirements. Our contributions are threefold.

- We added an attention layer to the Bert model to ensure the output text contains important information.
- We developed a PEM to generate text that fits perfectly within the designated text region of the travel magazine advertisement layout.
- To prove the efficiency of our model, we carried out extensive experiments on actual datasets.

II. Related work

In recent years, pre-trained language models have achieved very good results in multiple tasks of NLP. Pretrained models typically leverage large unlabeled corpora to learn general language representations. Kuleshov et al. [1] and others respectively proposed a synonym replacement attack algorithm based on greedy search, which replaces as many words in sentences as synonyms. On this basis, Ren et al. [2] also proposed a new algorithm PWWA. This algorithm uses a scoring function when selecting keywords, not only considering the importance of words in the sentence, but also classifying the model with the original input and adversarial samples. The impact of the results was also taken into account. Li et al. [3] and others introduced Bert into the generation of adversarial sample algorithm based on synonym replacement. When the algorithm is looking for synonyms of words, it uses the Bert model to generate grammatical and semantic alternatives according to the context of keywords. But these replacement keywords may greatly change the meaning of the information we need to extract. In early researchers who regarded generating summaries as sequential classification models, Shen et al. [4] used conditional random fields to sequentially classify sentences in binary order. Cheng et al. [5] created a flexible framework comprising of two main components: a hierarchical document encoder and an attention-based extractor. This architecture enables the development of diverse summarization models capable of extracting either sentences or words. In simpler terms, they developed a structure that can be used to build different types of summarization models that can extract important information from a document. Sumarunner et al. [6] form abstract summaries by interpreting original documents and generating meaningful sentences in shorter versions. This approach involves the model's more semantic understanding of the document, writing summaries in a human-understandable form. Devlin et al. [7] proposed Transformer's Bidirectional Encoder Representation, a new language representation model using masked language modeling and "next sentence prediction" tasks on a corpus of 33 million words. trained. Xiao et al. [8] proposed an RNN-based ETS model that combines global and local contexts.

III. Model

A. Prediction-evaluation models

The user enters a set of data representing the data $N = \{n_1, n_2\}$ of the text area in the magazine layout, where n_1, n_2 represent the length and width of the text area, n_3, n_4 represent the predicted text quantity and font size, and then output through iterative algorithm a set of candidate data $\left\{M^1, M^2, \ldots, M^i\right\}$, where $M^i = \left\{n_1, n_2, m_1^i, m_2^i\right\}$ the final output of the model is $M^{best} = \left\{n_1, n_2, m_1^{best}, m_2^{best}, p\right\}$, m_1 indicates the total number of texts, m_2 indicates the font size, and p indicates the evaluation value. As shown in Fig. 1.

Algorithm 1 Find M algorithm

1: **Input:** length (n_1) and width (n_2) of the text area in the layout, predict number of text words n_3 and font size n_4
2: **Output:** a set of candidate data $\left\{M^1, M^2, \ldots, M^i\right\}$.
3: $n_g = \beta \bullet \frac{S(n_1,n_2)}{Size(m_2)}$
4: set $M = [\ ]$,T = True,
5: $m_x = [m_{x1}, m_{x2}, m_{x3}, m_{x4}]$
6: while T:
7: for n_4 in m_x:
8: $M^i = \left\{n_1, n_2, m_1^i, m_2^i\right\} \leftarrow RatioT\left(n_3\right)$
9: if n_3 not in $[n_g - \alpha, n_g + \alpha]$:
10: T = False:

Where, n_g is the number of words required in the text area of the travel magazine advertisement layout calculated by the formula, $S(*)$ is the area for calculating the area, and $Size(*)$ is the area occupied by one character for the calculation of font size m_2, because each The length of words is different, and there are multiple words in a sentence, so this method cannot be used to calculate the number of words, so we use two methods (formula calculation and prediction model) to constrain the number of words in the text, β is English the ratio value between words and characters, the text font size range value set by m_x, $RatioT(*)$ is the text word count scaling function. α is the maximum value scaled by the amount of text.

B. Text summary generative model

Let $D = \{sent_1, sent_2, sent_3, \ldots, sent_i\}$ input documents, extractive text generation is to assign a label $y_m \in (0, 1)$. Using the Bert pre-training model to generate extractive text requires it to output the token of each sentence. The Bert pre-training model is trained as a masked-language model, and the output vector is the token of the sentence. At the same time, Bert has special segmentation embeddings for each independent sentence, but it only has two labels to represent sentences in the extractive summary. Adjust the input and Bert's embedding to extract the summary, and use E_A and E_B to distinguish continuous sentences. When encoding each sentence, we place a [CLS] and [SEP] at the beginning and end of each sentence. This allows the model to use these two tokens to judge the independence of the sentence. The three embeddings in Bert are shown in Fig. 2. The modified text is then represented as token X, and each token X is assigned three embeddings: token embeddings, segmentation embedding, and position embedding. Wu sum these three embeddings to form a single input vector x_i, which is then sent to the transformer. Then through the attention layer, to calculate the attention weight of each sentence, pass through a decoder after passing through the SoftMax layer. Finally, output the summary text.

$$h^{l-1} = A\left(a^{l-1}\right) \tag{1}$$

$$\tilde{h}^l = LN\left(h^{l-1} + MHAtt\left(h^{l-1}\right)\right) \tag{2}$$

$$h^l = LN\left(\tilde{h}^l + FFN\left(\tilde{h}^l\right)\right) \tag{3}$$

Where $a^0 = PosEmb(X)$; X represents the sentence vector output by Bert; $PosEmb(*)$ represents the function of adding positional embedding to T; and indicates the position of each sentence. $A(*)$ represents the self-attention layer. $LN(*)$ is the layer normalization operation. $MHAtt(*)$ is multi-head attention processing. $FFN(*)$ represents the feed-forward neural network operation. The superscript l represents the number of stacked layers.

$$\hat{y}_i = \sigma\left(W_o h_i^l + b_o\right) \tag{4}$$

Where h_i^l represents the vector output in the sentence i layer l layer transformer.

Finally, through $Control(K)$, the output is $Y = \{S_1, S_2, S_3, \ldots, S_K\}$, output the top K sentences of importance and form a coherent text, where K is the best output in the PEM text word count data.

IV. Experiments and results

A. Datasets

In our study, we utilized two sets of data. The first set was used to train and test our PEM. Since public datasets were not suitable for our purposes, we can only manually mark

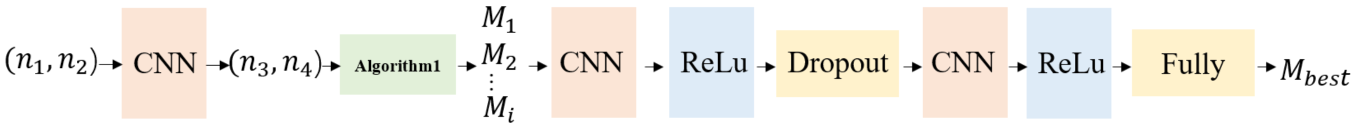

Fig. 1. Prediction-evaluation model.

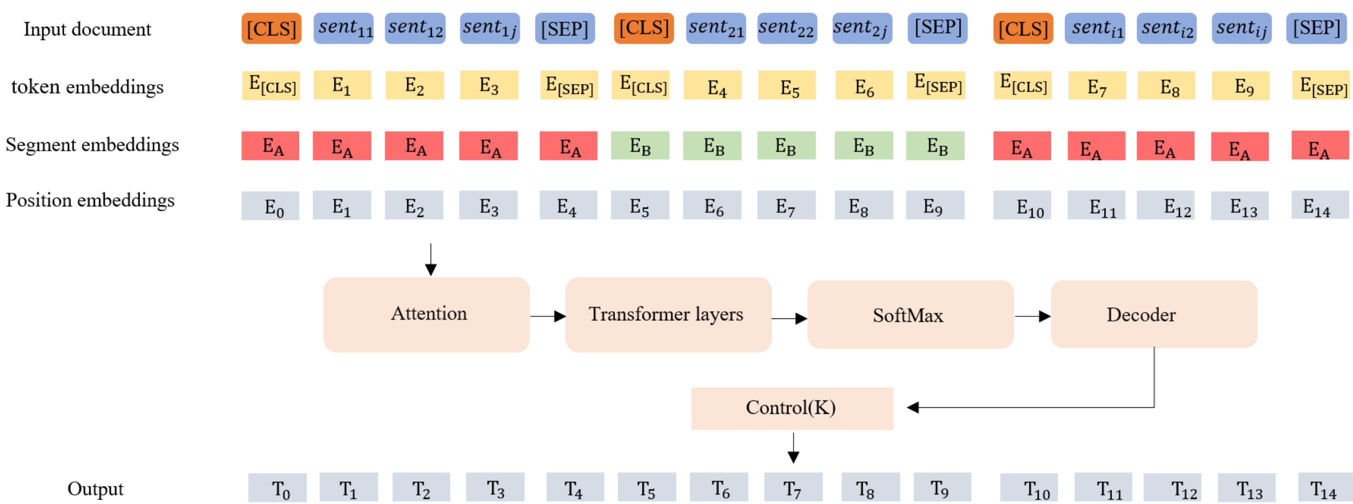

Fig. 2. This model is our text generative model.

the data by ourselves. So we created our own dataset by collecting numerous travel magazine advertisement layouts. We then manually inserted text into these layouts, varying the font sizes and text quantities to create visually appealing travel magazine ads. These ads were rated on a scale of attractiveness, and a total of 1230 ads were marked. We set 70 percent training data and 30 percent test data for our model development. The second set of data is used in the text summary generative model. Text summarization and abstract extraction are basically similar and can be performed with the same kind of public data sets and evaluation standards. They are DailyMail [9], NYT [10] public data sets.

TABLE I
SHOWS THE SIZE OF THE TRAIN AND TEST OF THE TWO PUBLIC DATASETS, AND SHOWS THE AVERAGE NUMBER OF WORDS AND SENTENCES IN THE SOURCE DOCUMENTS AND ABSTRACTS OF EACH DATASET.

Datasets	Docs	avg doc length		avg. summary length	
		words	sentences	word	sentences
DailyMail	196,961	653.33	29.33	54.65	3.86
NTY	96,834	800.04	35.55	45.54	2.44

B. Text generation evaluation criteria

There are certain differences in manual evaluation. This article uses Rouge, an evaluation index dedicated to text summarization tasks. We use two of the evaluation indexes $Rouge_{-N}$, $Rouge_{-L}$.

$$Rouge_{-N} = \frac{\sum\limits_{Seref_{summaries}} \sum\limits_{gram_N \in S} Count_{match}(gram_N)}{\sum\limits_{Seref_{summaries}} \sum\limits_{gram_N \in S} Count(gram_N)} \tag{5}$$

$$Rouge_{-L} = \frac{LCS(Y, S)}{len(s)} \tag{6}$$

where $LCS(Y, S)$ represents the longest common subsequence of Y (generate text) and S (true text).

C. Predictive evaluation model results

Our predictive evaluation model aims to predict a range of the word count for a generated text. While it is impossible to achieve an accurate word count due to the varying lengths of English words, we have observed that for a certain range of English words, the length of English sentences tends to be similar. This is because it is uncommon to have a large number of short words or long words in a single article. We have trained our model using a large number of articles, and the loss of the model is in Fig. 3.

TABLE II
ACCURACY RESULTS FOR THREE FORECAST-EVALUATION MODELS.

Model	LSTM	SVR	PME
Acc	0.93	0.91	0.96

From Table II, it can be seen that the PME model has the highest Acc. Because our data is simple and has no

Fig. 3. Is training and validation loss of the prediction evaluation model.

chronological order, we utilized both LSTM and SVR models as the basic models to design a simple PEM network. However, neither of them is as accurate as the CNN-based model (PME). This may be because our data consists of only a few simple and no time-ordered arrays, and it is not necessarily the case that a more complex model will yield better results.

D. Text summary generative model Results

In Table III, we used three indicators to evaluate our model, we can see that our model is in the DailyMail dataset $Rouge_{-L}$ is higher than other models, but it is only slightly higher in the $Rouge_{-2}$ evaluation standard, and it is the same in the NYT dataset. It can be seen that our model is more relevant in outputting summarized text and real text summaries.

TABLE III
RESULTS OF PUBLIC DATASETS DAILYMAIL AND NYT.

Datasets	Models	$Rouge_{-1}$	$Rouge_{-2}$	$Rouge_{-L}$
DailyMail	Nallapati [11]	35.4	13.3	32.6
	Lead-3 [6]	39.2	15.7	35.5
	Refresh [12]	41.0	18.8	37.7
	SqueezeBert [13]	42.51	19.56	38.92
	Ours	**43.2**	**19.8**	**39.88**
NYT	Nallapati	34.2	13.4	31.9
	Lead-3	39.1	15.8	34.9
	Refresh	40.0	18.2	36.6
	SqueezeBert	42.23	19.51	38.63
	Ours	**42.8**	**19.9**	**39.7**

E. Ablation experiment

Our model utilizes a self-attention layer after tokenizing the original document to calculate the importance of each sentence. This enables us to eliminate sentences with extremely low importance and some sentences with similar meanings, resulting in a more accurate summary. The vector output generated by the attention layer is then input to the transformer layer, which further improves the accuracy of our model. As shown in Fig. 4 and Fig. 5, the output of our model is relatively better than other models. Because the attention layer assigns an importance level to each sentence, which helps to extract more important text in subsequent steps.

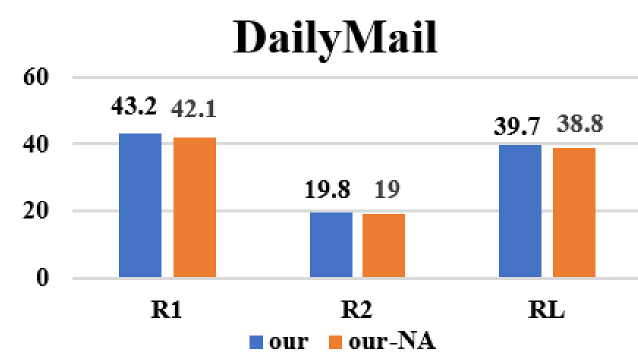

Fig. 4. Results of ablation experiments on public data DailyMail.

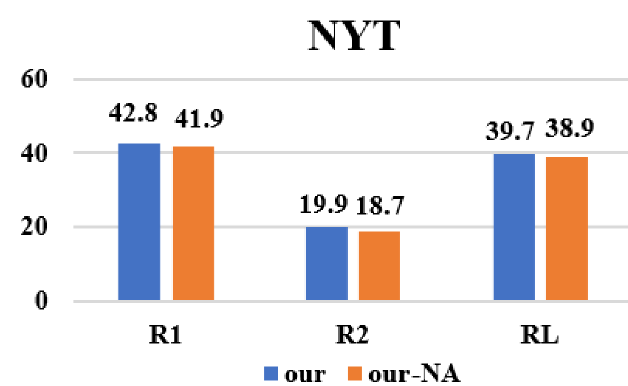

Fig. 5. Results of ablation experiments on public data NYT.

F. Final display result

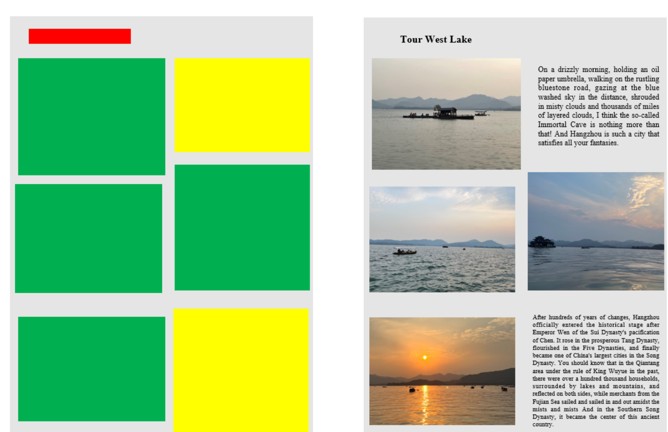

Fig. 6. The final generated text is posted in the layout's renderings, and the images in tourism magazine advertisements are provided, not generated.

The user provides the layout diagram on the left as shown in Fig. 6, where different colors represent different regions. Our model generates suitable text and attaches it to the corresponding text area in the layout diagram. To verify the aesthetic quality of the generated text attachment, the user provides pictures, which are then pasted in the corresponding positions in the layout.

V. Conclusion and Future Work

We present a model that can generate travel magazine texts that are adaptive to user-input travel magazine layouts. Our model surpasses other models in performance based on extensive experimental results on two public datasets. Furthermore, we generate text that conforms to the layout of travel magazines. The self-attention layer in our model calculates the importance of each sentence after tokenizing the original document, which results in higher accuracy of the generated text. As future work, we aim to implement the ability for users to input travel magazine layouts and scenic spot photos, and automatically generate travel magazines that conform to the travel magazine layout and are relevant to the provided scenic spot photos.

Acknowledgment

This research was supported by the Basic Public Welfare Research Program of "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant 2023C01231.

References

[1] V. Kuleshov, S. Thakoor, T. Lau, and S. Ermon, "Adversarial examples for natural language classification problems," 2018.

[2] S. Ren, Y. Deng, K. He, and W. Che, "Generating natural language adversarial examples through probability weighted word saliency," in Proceedings of the 57th annual meeting of the association for computational linguistics, 2019, pp. 1085-1097.

[3] L. Li, R. Ma, Q. Guo, X. Xue, and X. Qiu, "Bert-attack: Adversarial attack against bert using bert," arXiv preprint arXiv:2004.09984, 2020.

[4] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen, "Document summarization using conditional random fields," in IJCAI, 2007, pp. 2862-2867.

[5] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," arXiv preprint arXiv:1603.07252, 2016.

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

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[8] W. Xiao and G. Carenini, "Extractive summarization of long documents by combining global and local context," arXiv preprint arXiv:1909.08089, 2019.

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

[10] G. Durrett, T. Berg-Kirkpatrick, and D. Klein, "Learning-based single-document summarization with compression and anaphoricity constraints," arXiv preprint arXiv:1603.08887, 2016.

[11] R. Nallapati, B. Zhou, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," arXiv preprint arXiv:1602.06023, 2016.

[12] S. Narayan, S. B. Cohen, and M. Lapata, "Ranking sentences for extractive summarization with reinforcement learning," arXiv preprint arXiv:1802.08636, 2018.

[13] S. Abdel-Salam and A. Rafea, "Performance study on extractive text summarization using BERT models," Information, vol. 13, p. 67, 2022.