An Improved Template Representation-based Transformer for Abstractive Text Summarization

Jiaming Sun, Yunli Wang, Zhoujun Li*
State Key Lab of Software Development Environment
Beihang University
Beijing, China
{jiamingsun, wangyunli, lizj}@buaa.edu.cn

***Abstract*—Text summarization plays an important role in various NLP applications. Using templates with generation methods is an effective way to address abstractive summarization. However, existing template-enhanced generation approaches use templates in a naive way and mainly adopt RNN-based Seq2Seq models, so they cannot make full use of valid information in the templates and suffer from templates' noise. To mitigate these problems, we propose a new abstractive summarization model called Summarization Transformer with Template-aware Representation (STTR), which uses a template-aware document encoding module and a document representation shifting loss to preserve the useful information and filter the noise of the template. The experiments on the Gigaword and LCSTS datasets show that our method outperforms baseline models and achieves a new state-of-the-art.**

I. Introduction

The goal of text summarization is to shorten an original article or paragraph while retaining important information. There are two main approaches to do this: extractive and abstractive summarization. Extractive summarization directly selects sentences from the original text as summary. Abstractive summarization rewrites and generates sentences by understanding the source input. We focus on abstractive summarization, which can produce summarizations more flexibly.

Due to the complexity of natural language and the redundant information present in texts, generating summaries based only on the source articles themselves presents a significant challenge. As a result, a great deal of works has tried to add external information for guiding the generation. The external information includes keywords [1], [2], topics [3], entities [4] and even summarization templates. Existing research shows that introducing templates to generation models is an efficient way to generate concise and coherent summaries and achieves the state-of-the-art performance [6].

For template-enhanced generation approaches, there are two subtasks to address: templates construction and summarization generation with constructed templates. Templates, largely created by experts with domain knowledge, are widely used in traditional summarization methods [5]. However, handcrafted templates cannot adapt to semantic changes in the original text. In recent years, summarization resources become more and more abundant. This makes it possible to construct better templates by retrieval methods. In this work, we only focus on making full use of templates and simply use the existing retrieval method to construct the templates.

For generation with constructed templates, existing methods can be divided into two types: separately encoding for template and document [7], and interacting encoding for template and document [6]. However, all of them do not make full use of the template and may introduce some noise to the decoder. Wang et al. [6] proposed BiSET, which uses bi-directional selective encoding and achieves the state-of-the-art in previous works. However, we think bi-directional interaction will introduce noise to the representation of the document because of the non-filtered information in template. And separately encoding for template and document does not select useful information. To mitigate these problems, we propose a novel method called Summarization Transformer with Template-aware Representation[1] (STTR), which includes two Transformer-based encoders and a Transformer-based decoder compared to bi-directional interaction. We only adopt a multi-head attention mechanism to get a template-aware encoding of the document, which selects effective semantics related to the document. It plays an important role in avoiding the noise of the template. Further, we assume that if the information selected by the attention mechanism is useful and noise-free, it will be close to the target representation and be more helpful to construct the target. So we further propose a document representation shifting loss to force the template-aware encoding of the document shift to the representation of the target. Finally, we use both template-aware encoding and the original representation of the document to generate the summaries.

Extensive experiments are conducted on Gigaword [22] and LCSTS [23] datasets, which are widely used abstractive summarization benchmarks. The results are evaluated by ROUGE [24] metric. Experiment results show that our method outperforms the existing baseline model and achieves a new state-of-the-art. The ablation test shows the effectiveness of STTR's components. We further conduct experiments and confirm that the similarity between the target representation and the selected information is positively correlated to the performance, we also test the robustness of our method by using different quality templates and experiment shows that our method can consistently outperform the baselines with

* Corresponding Author

[1]All our code is available at: https://github.com/SunJMMMM/STTR

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

different templates.

Overall, our contributions can be summarized as follows:

- A novel template-based abstractive summarization model, STTR, which is applied with template-aware document encoding module and document representation shifting loss.
- The robustness of the STTR and the effectiveness of the proposed components are verified through extensive experiments.
- Evaluation of the proposed model using benchmark datasets confirm its capacity to deliver new state-of-the-art performance.

II. Relate Work

A. Abstractive Summarization

There are two principal approaches to text summarization: extractive, and abstractive summarization. They both try to create a shorter version of a source document while retaining the key information. Unlike the extractive approach [9], [10], abstractive summarization aims to generate new content that is based on an understanding of the article, which is closer to human-based approaches. A number of abstractive methods have been proposed recently, all of which perform well. Rush et al. [11] proposed an RNN-based sequence-to-sequence model with an attention mechanism to generate the summaries. In [12], a copying mechanism was incorporated that directly copies words from the original text to solve the problem of out-of-vocabulary. In [13], a coverage mechanism was proposed that preserves the attention history to prevent the repetition of high-probability words. A CNN-based module is introduced in [14] to learn document representation and used to filter the redundant information in the encoder. Song et al. [15] finetuned the BERT [30] by setting seen and unseen words. This strengthens its ability to generate unknown words, enabling it to deliver good results on the Gigaword dataset.

B. Template-based Summarization

Some works have attempted to introduce additional information to guide the generation of the summary, including template information. In [5], the summarization is guided by constructed templates. However, these time-consuming summaries are manually created and require domain knowledge. Template information is also added in [7] by retrieving similar summaries from the dataset and directly connecting them to the input. Wang et al. [6] proposed a merge strategy that consists of T2A and A2T mechanisms to soft-filter the source information. These template-based methods mostly use RNN to extract and merge template information. Our model, by contrast, uses the Transformer architecture and has an attention mechanism to extract and merge the key information.

C. The Transformer

The Transformer [8] is a sequence-to-sequence architecture, the strength of which is founded upon its self-attention mechanism. In comparison to RNN, it can capture deeper, finer-grained features and superior parallelism. It has been widely used as a powerful baseline in neural machine translation and has also been applied to text summarization. In [16], a constrictive attention approach is proposed that provides additional irrelevancy attention features, so that the model can better distinguish the input content. The attention mechanism is strengthened in [17] by constructing a learnable position bias and modelling the importance of each word at the decoding step. Cai et al. [18] obtained a document-level information through a CNN-based module which is added to the Transformer to enhance its ability for generating summaries. Unlike the above approaches, where original document information is used to optimize the transformer, our model looks at how to integrate template information into the Transformer architecture to achieve better results.

III. Problem Formulation

For a source document, $\mathbf{X} = \{x_1, x_2, \ldots, x_{L_x}\}$, there is a ground truth summary, $\mathbf{Y} = \{y_1, y_2, \ldots, y_{L_y}\}$ where both consist of a list of words. L_x and L_y are the lengths of the source document and the summary, respectively. In template-based summarization, another pair of texts, $\{\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}\}$, are retrieved from the training corpus, according to the similarities between $\mathbf{X}$ and $\widehat{\mathbf{X}}$. The corresponding summary of $\widehat{\mathbf{X}}$ (i.e. $\widehat{\mathbf{Y}}$) is considered the template summary of $\mathbf{X}$, which can be formulated as $\mathbf{T} = \{t_1, t_2, \ldots, t_{L_t}\}$. Given a tuple of $\{\mathbf{X}, \mathbf{T}\}$, STTR learns to estimate the conditional distribution of generating the summary, $\mathbf{Y}$. This is denoted by $P_\theta(\mathbf{Y}|\mathbf{X}, \mathbf{T})$, where θ is the parameter of STTR.

IV. Method

In this section, we describe our Summarization Transformer with Template-aware Representation (STTR) in detail. We begin with the multi-head attention mechanism. This is used in both of the Transformer and our proposed integration approach. Then, we look at the Transformer, which provides the baseline model for the STTR. Finally, we describe our template-aware document encoding module and document representation shifting loss.

A. Multi-head Attention Mechanism

To capture the representation of the input document, the Transformer uses a multi-head attention mechanism. This can capture long-term dependencies and more detailed features than RNN and CNN. The multi-head attention mechanism is also used in the STTR to extract the representation of the template and merge it with the source representation. It is based on Scaled Dot-Product Attention [8]:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \qquad (1)$$

where, $Q \in \mathbb{R}^{n \times d_k}$, $K \in \mathbb{R}^{m \times d_k}$ and $V \in \mathbb{R}^{m \times d_v}$ represent the query, key and value, respectively. $\mathbb{R}$ is real field. n and m are the length of the query and key/value sequences, respectively. d_k and d_v are the dimensions of the key and the value, respectively. Through the attention mechanism, we can

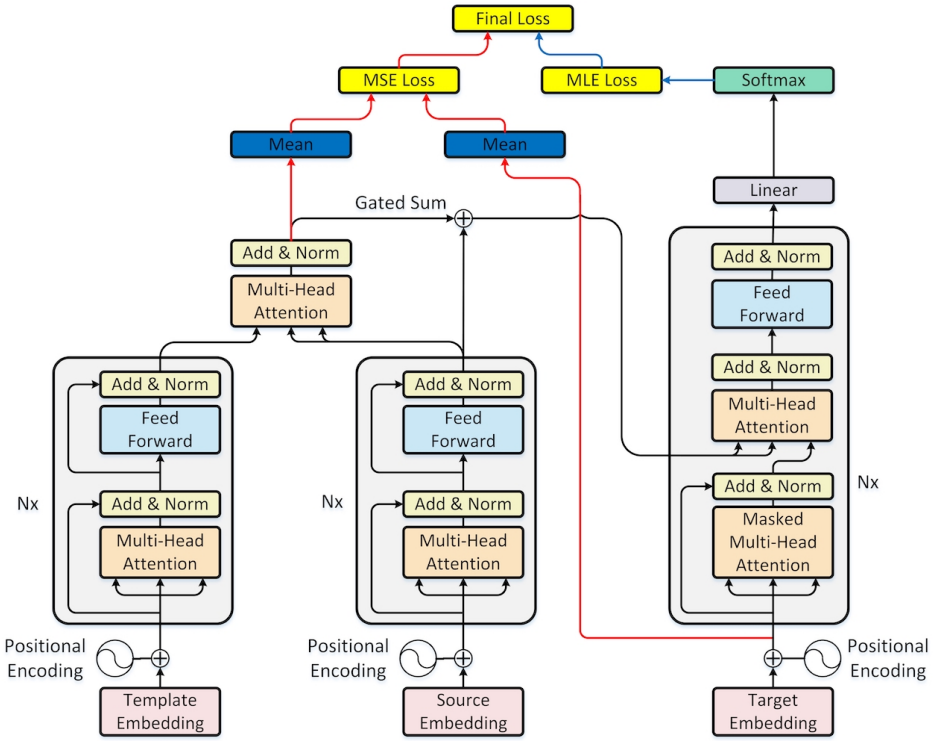

Fig. 1. An overview of our proposed STTR which has a similar structure with Transformer. We extend a template encoder(left) to extract template representation and merge it with source encoder(middle) through attention mechanism. An additional document representation shifting loss is added (red line) with the MLE loss (blue line) provided by decoder (right) as the final loss.

convert V into a new sequence, according to the correlation between Q and K.

Further, the multi-head attention mechanism concatenates multiple basic attentions with different parameters to reinforce its capability:

$$MultiHead(Q,K,V) = Contact(head_1, \ldots, head_h)W \quad (2)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (3)$$

where $Contact(\cdot)$ operation concatenates h basic attentions into a final value with the dimension of d_v; $W \in \mathbb{R}^{d_v \times d_{model}}$ represents the final linear projections and $W_i^Q \in \mathbb{R}^{d_k \times d_k}$, $W_i^K \in \mathbb{R}^{d_k \times d_k}$ and $W_i^V \in \mathbb{R}^{d_v \times d_v}$ are all learnable parameters.

B. Transformer Baseline Model

Input Due to the structure of the multi-head attention mechanism, the distance between any two words is equal, which leads to a lack of positional information. To solve this, the Transformer adds positional encoding by using a heuristic sine and cosine function:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \quad (4)$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \quad (5)$$

where, pos is the position of word in text; i is the dimension index of embedding, and the dimension of model is d_{model}.

The final input is the sum of token embedding and position embedding:

$$E_{input} = E^w_{input} + E^p_{input} \quad (6)$$

where $E^w_{input} = \{e_1, e_2, \ldots, e_{L_{input}}\}$ are the token embedding and $E^p_{input} = \{p_1, p_2, \ldots, p_{L_{input}}\}$ are the position embedding of each word in input text.

Encoder The encoder is used to extract the features of the input text and represent them as a vector. It is a stack of N layers, each composed of two sublayers: a multi-head self-attention layer and a fully-connected feed-forward network. Each layer is surrounded by an $AddNorm$ operation, which is a combination of residual connections and layer normalization.

The multi-head self-attention layer is the same as the multi-head attention detailed above, where $Q = K = V$. The output of the n-th multi-head self-attention layer of the source document can be defined as follows:

$$Z_s^n = AddNorm(MultiHead(H_s^{n-1}, H_s^{n-1}, H_s^{n-1})) \quad (7)$$

where, H_s^{n-1} is the output of the $n-1$ encoder layer. H_s^0 is the source document input:

$$H_s^0 = E_s \quad (8)$$

The feedforward network (FFN) sublayer of the n-th layer can be formalized as:

$$H_s^n = AddNorm(FFN(Z_s^n)) \quad (9)$$

$$FFN(Z_s^n) = relu(W_s^n Z_s^n + b_s^n) + b'^n_s \quad (10)$$

where, W_s^n, b_s^n, ${b'}_s^n$ are all learnable parameters. H_s^N is the final encoder output of the source document x.

Decoder The purpose of the decoder is to generate the summary from the encoder information. It is also a stack of N layers with three sublayers: a masked multi-head attention layer; a multi-head attention layer; and a feed forward layer.

Like the encoder, to obtain a vector representation of the summary, the multi-head self-attention applies a mask matrix to prevent the future word from being unknown:

$$H_{ms}^n = AddNorm(MultiHead^*(H_{dl}^{n-1}, H_{dl}^{n-1}, H_{dl}^{n-1})) \quad (11)$$

where, H_{dl}^{n-1} is the output of the $n-1$ decoder layer and H_{dl}^0 is the target text input:

$$H_{dl}^0 = E_t \quad (12)$$

After this, cross-attention is applied between the encoder and decoder. Thus, the representation of the current decoding step can be transferred to the source representation:

$$H_d^n = AddNorm(MultiHead(H_{ms}^n, H_s^N, H_s^N)) \quad (13)$$

Then, FFN is applied to get the final representation:

$$H_{dl}^n = AddNorm(FFN(H_d^n)) \quad (14)$$

The probability of the generated word is calculated by using a linear layer and a softmax operation:

$$p(y_i|y_{<i}, x) = softmax(h_{i_{dl}}^N W_o) \quad (15)$$

where, $W_o \in \mathbb{R}^{d_{model} \times |V_t|}$, with $|V_t|$ being the size of the target vocabulary and $h_{i_{dl}}^n$ is the i-th element in $H_{dl}^N = \{h_{1_{dl}}^N, h_{2_{dl}}^N, \ldots, h_{L_{y\,dl}}^N\}$.

The generation loss is trained by minimizing the cross-entropy loss, which maximizes the probability of generating a ground truth summary:

$$\mathcal{L}_{mle} = -\sum_{i=1}^{I} logp(y_i^*|y_{<i}, x) \quad (16)$$

C. Template-aware Document Encoding

In our approach, we do not focus on the retrieval module and follow previous work [6] by retrieving the templates from training corpus. After that, the template information is integrated into the Transformer architecture and proposed STTR to make full use of the template summary.

As with the encoder of the source text, a template encoder with N layers is applied to extract the template features, as follows:

$$H_t^n = AddNorm(FFN(Z_t^n)) \quad (17)$$

$$Z_t^n = AddNorm(MultiHead(H_t^{n-1}, H_t^{n-1}, H_t^{n-1})) \quad (18)$$

where, H_t^0 is the template sentence input:

$$H_t^0 = E_t \quad (19)$$

To let the template guide the source by focusing on the key content, multi-head attention is applied between the source representation and target representation:

$$H_{ts} = AddNorm(MultiHead(H_t^N, H_s^N, H_s^N)) \quad (20)$$

In this way, a template-aware document representation can be obtained that is tightly related to the content of the template, thus achieving the goal of filtering out irrelevant information. However, the source document also contains content that is not in the template and that is useful for the target. We need the model to be able to select the original content or the transferred content.

To achieve soft selection of either the original document or the template-aware document, a gated sum mechanism can be applied:

$$g = \sigma(W_g[H_{ts}, H_s^N] + b_g) \quad (21)$$

$$S^N = g \odot H_s^N + (1-g) \odot H_{ts} \quad (22)$$

where W_g, b_g, are all learnable parameters. $\odot$ means the element-wise multiplication. σ is the sigmoid activation function. The final output of the encoder changing from H_s^N to S^N.

D. Losses of STTR

Document Representation Shifting Loss As the templates are summaries from other similar articles, there may be some content that is irrelevant to the target summary. Inspired by [19], which makes the sentence representation between source and target closer in neural machine translation, we propose minimizing the difference between the template-aware document and target representation to reduce the impact of the noisy information in the templates. A mean operation is applied to the target and template-guided document representation because this has proved effective at obtaining the sentence representation of a sequence [20], [21]. After this, a document representation shifting loss is applied to narrow the gap between the template and target:

$$\mathcal{L}_{mse} = \left\| \widehat{H}_{ts} - \widehat{E}_t \right\|^2 \quad (23)$$

where,$\widehat{\ }$is the mean operation. Through the document representation shifting loss, the template-aware document representation can be brought closer to the target space during training, thus filtering out the irrelevant information in the template.

Generation Loss Similiar to the Transformer baseline, the generation loss of STTR is trained by minimizing the cross-entropy loss, which maximizes the probability of generating a ground truth summary with the corresponding template t.

$$\mathcal{L}'_{mle} = -\sum_{i=1}^{I} logp(y_i^*|y_{<i}, x, t) \quad (24)$$

The final training loss is:

$$\mathcal{L} = \mathcal{L}'_{mle} + \mathcal{L}_{mse} \quad (25)$$

V. Experiments

A. Dataset and Evaluation Metrics

We conducted experiments on Gigaword and LCSTS datasets, which are widely used benchmark datasets for abstractive text summarization. Each sample in the Gigaword dataset is a pair of sentences that consists of the first sentence

of an article with its headline. The train/validate/test splits of the extracted corpus contained 3.8M/8K/2K instances. For a fair comparison, we used the version preprocessed by [6], which contains the source-target pairs and the retrieved templates[2]. The LCSTS dataset is a large Chinese short text summarization dataset collected from the microblogging website Sina Weibo. We followed Hu et al. [23] to preprocess the dataset, and the train/validate/test splits of the dataset are 2.4M/8K/0.7K.

Following the previous work [22], we employed the ROUGE presented in [24] as our evaluation metric. The ROUGE metric computes the overlapping lexical units between the generated summaries and the reference summaries where the official ROUGE script is applied. We used the full-length F1 scores of ROUGE-1, ROUGE-2 and ROUGE-L to evaluate the generated summaries, which corresponds to unigram, bi-gram and longest common subsequence overlaps.

B. Implementation Details

We implemented the experiment in PyTorch on 2 Tesla V100s. A byte-pair encoding algorithm [5] was used to segment the words for the Gigaword dataset, with the vocabulary size being about 15000. For the LCSTS dataset, we took the character-level sequence for training and evaluation. The vocabulary size was about 10000. The source embedding, target embedding, and linear sublayer are shared with a dimension size of 512. The STTR model used four attention heads, with the dimension of the feed-forward network being 1024. We set the layer number for the template encoder, source encoder and target decoder to 6. In training, the cross-entropy loss was used for maximum likelihood estimation (MLE) and label smoothing was introduced to reduce the likelihood of overfitting. For the consideration of memory and efficiency, we set the numerical values such as embedding sizes, the dimension of FFN and the number of attention heads as above. We used an Adam optimizer [31], where $\beta 1 = 0.9$, $\beta 2 = 0.98$ and $\varepsilon = 10^{-9}$. The learning rate was 10^{-4} and reduced to a half when the validation loss did not improve. The dropout rate was set to 0.1. For a fair comparison with previous work [6], we used beam search of size 5 to generate the summaries during decoding.

C. Main Results

Tables I and II show the main results of applying the proposed model on the Gigaword and LCSTS datasets. We compare with models based on the different structure including RNN, CNN and Transformer. We also compare it against methods that introduce external information, including templates, keywords and topics.

The baseline models for Gigaword include the following: **ABS+** [11], which is an RNN-based model enhanced by local attention and hand-extracted features. **Pointer-Generator** [13] is an RNN-based model with a pointer mechanism which enables the model to copy a word from the original text.

[2]Preprocessed data are available at: https://github.com/InitialBug/BiSET

TABLE I
EXPERIMENT RESULT ON GIGAWORD DATASET

Model	R-1	R-2	R-L
ABS+ [11]	29.76	11.88	26.96
Pointer-Generator [13]	35.98	15.99	33.33
Global [14]	36.30	18.00	33.80
RL-Topic-ConvS2s [3]	36.92	18.29	34.58
Re3 [7]	37.04	19.03	34.36
ContrastiveAttention [16]	38.72	19.09	35.82
ControlCopying [15]	39.08	20.47	36.69
BiSET [6]	39.11	19.78	36.87
STTR	**43.58**	**22.73**	**40.31**

TABLE II
EXPERIMENT RESULT ON LCSTS DATASET

Model	R-1	R-2	R-L
RNN context [23]	29.90	17.40	27.20
CopyNet [12]	34.40	21.60	31.30
DRGN [27]	37.00	24.20	34.20
Actor-Critic [28]	37.51	24.68	35.02
SuperAE [29]	39.20	26.00	36.20
Global [14]	39.40	26.90	36.50
Keyword [1]	40.90	28.30	38.20
ContrastiveAttention [16]	44.35	30.65	40.58
STTR	**45.21**	**33.59**	**42.44**

Further, it introduces a coverage mechanism to avoid the duplication of generated content. **Global** [14] uses a CNN-based module to capture a document representation, which is then used to filter the source input. **RL-Topic-ConvS2s** [3] is a convolutional sequence-to-sequence model trained through reinforcement learning that adds topic information into the model. **ContrastiveAttention** [16] is a Transformer-based model, which apply opponent attention to help model focus more on the relevant part. **ControlCopying** [15] treats the text summarization as a language modelling task and fine-tunes BERT by masking seen and unseen word separately. **Re3** [7] uses a retrieval module to retrieve templates and directly connect it with the source input. **BiSET** [6] further uses a template-to-article (T2A) and article-to-template (A2T) mechanism to enhance template integration.

The baseline models for LCSTS included: **RNN context** [23], which is an RNN-based model with attention mechanism. **CopyNet** [12] adds a copy mechanism into an attention-based sequence-to-sequence model. **DRGN** [27] is a conventional sequence-to-sequence model with a deep recurrent generative decoder that is used to improve the quality of summaries. **Actor-Critic** [28] uses a reinforcement approach during the training of model to overcome the typical problems encountered by teacher-forcing methods. **SuperAE** [29] uses an adversarial learning approach to supervise the representation of the source text. **Keyword** [1] uses a TextRank algorithm to extract the keyword from the article and add them into the model. The **Global** and **ContrastiveAttention** (mentioned above) were also used as baseline models for LCSTS.

As can be seen in Tables I and II, the proposed model

outperformed the baseline models across both datasets. In the Gigaword dataset, STTR performed better than **ControlCopying**, which is a Transformer-based model using BERT as additional information and achieves SOAT without template information. The results show that it is more useful to introduce templates than other external information. Additionally, our model improved the performance by 4.47 for ROUGE-1, 2.95 for ROUGE-2 and 3.44 for ROUGE-L scores compared with the template-based summarization model **BiSET**. The results show the advantages of using a Transformer architecture in template-based abstractive summarization and the effectiveness of the proposed methods. In the Chinese LCSTS dataset, our model outperforms the best model **ContrastiveAttention** by 0.86 for ROUGE-1, 2.94 for ROUGE-2 and 1.86 for ROUGE-L, which is also a Transformer-based model. These results show that our proposed methods are equally effective in different languages and performs better at the sentence level(i.e. R-2, R-L).

D. Ablation Study

We also undertook an ablation study using the Gigaword dataset to assess the performance gain acquired by adopting the proposed integration approach and loss. We compared the integrated approach with a simple connection method and the previous work, BiSET [6] to verify that the proposed method is suitable for a transformer architecture.

In Table III, the first row is the Transformer baseline, which does not integrate the template information. The results show that the Transformer baseline has a competitive result compared to previous RNN-based models. In the second to fourth line, we added template information into the Transformer and tested three different merge approaches. In the Connection approach there was a simple connection between the source document and the template representation after applying the encoder. BiSET involved merging the source and template by using a T2A and A2T mechanism, as proposed in the original BiSET paper [6]. 'Attention' refers to the proposed approach, which uses multi-head attention to get the template-guided document representation and then use gated sum to soft select the key information. The results show that the performance improved, indicating that the introduction of the template had a positive effect. The Connection approach performed better than BiSET, which implies that BiSET is not suitable for the Transformer architecture. The results of the proposed Attention method (in the fourth line) are significantly better, underscoring the effectiveness of the proposed method.

TABLE III
ABLATION STUDY ON GIGAWORD

Method	R-1	R-2	R-L
Transformer	37.03	17.94	34.41
+BiSET	41.39	21.03	38.30
+Connection	42.39	21.46	39.02
+Attention (our approach)	43.47	22.46	39.80
+Attention +MSE Loss (our full approach)	**43.58**	**22.73**	**40.31**

TABLE IV
SIMILARITY ANALYSIS

Model	Gigaword		LCSTS	
	Sim(%)	ROUGE-L	Sim(%)	ROUGE-L
STTR w/o loss	48.00%	39.80	51.20%	42.13
+MSE Loss	63.16%	40.31	80.20%	42.44

Finally, we looked at using the proposed loss technique. Here, the results were even better. In particular, the loss improved the performance significantly at the sentence level (i.e. R-L). This means that having a similar template-aware document and target at the sentence-level can improve the overall quality of the generated text.

E. Sentence-level Similarity Analysis

We also examined the impact of sentence-level similarity on the ROUGE score. Following the previous work [26], we used cosine similarity to measure the distance between two vectors. In detail, we first applied a mean operation to the template-aware document representation and target representation. We then calculated the cosine score of these two vectors:

$$sim = cosine(\widehat{H}_{ts}, \widehat{E}_t) \tag{26}$$

Since the cosine similarity ranges in $[-1, 1]$, we normalize the results to $[0, 1]$.

It can be seen from Table IV that, after adding the loss, the similarity between the two representations increased. Moreover, there is a positive correlation between the similarity and the ROUGE-L score, further confirming that the proposed method is effective.

F. Robustness

We further evaluate the robustness of our proposed model with different qualities of templates on ROUGE-2 metric. To do that, we first briefly describe the Fast Rerank module in previous work [6] and our experiment settings. After that, we calculated the rouge metric of the template itself and the scores of the two models which are STTR and BiSET under different templates.

The retrieval of templates is divided into two stages. First, a standard information retrieval library[3] is used to retrieve a small number of candidate article-summary pairs, according to their similarity. After that, a Fast Rerank module is used to estimate the correlation between the retrieved summaries and the query article. The summary with the highest score is used as the template. It is composed of a Convolution Encoder Block, a Similarity Matrix and a Pooling Layer. This makes it possible to estimate the correlation between a query article and retrieved summaries quickly and accurately. In detail, we set the rerank number of template candidates to 5, 10, 20 and 30, which results in different quality of templates. Further, we constructed a random-selected template which is totally irrelevant with source document.

[3]https://lucene.apache.org

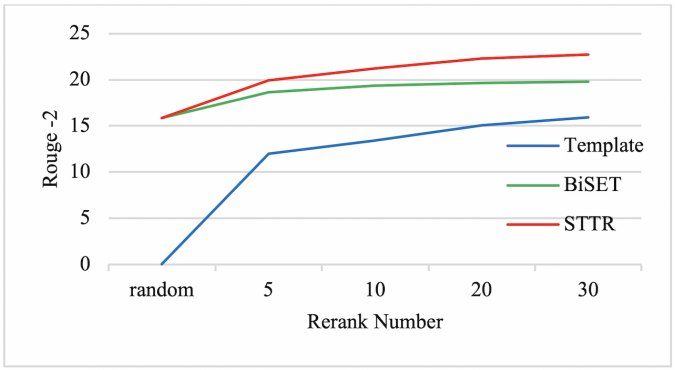

Fig. 2. The ROUGE-2 score across different quality of templates

In Fig. 2, the blue line represents the template score, which increased with the reranked candidate number, and the random-selected templates are zero. It can be seen by comparing the blue and red lines that STTR always performed better than templates with different qualities. This suggests that the model can learn and select useful information from the template to help generate the summary. Even if the abstract is randomly selected, the model can still produce relatively normal results, showing that the model does not completely rely on the template. It can be seen from the red and blue lines that the score of BiSET grows slowly with the improvement of the template quality. However, STTR can always extract useful information from the template and further improve the quality of the generated abstract.

G. Case Study

Table V shows the example of generated summaries on both datasets. Additionally, we added the results of Transformer baseline and STTR without proposed loss. The key information of reference, Transformer baseline and template are colored in red, blue and green, respectively. From the example, we can conclude that:

1) The introducing of template information can result in better summaries. For example, in the first article, the STTR without proposed loss generates the correct number of died people which is 'four', as mentioned in the template while the Transformer generates the wrong number 'two'. In the second article, the templates-guided models all pay attention to the correct point, 'payment', while the Transformer baseline pay attention to the wrong point, which is 'settlement business'. It can be seen that a proper template can guide the model without deviating from the main idea of the article and not making factual errors to some extent.

2) Soft select attention can avoid noisy information in the template. For example, in the first article, the STTR without loss changes the 'boiler blast' in the template to 'bomb blast', which is the true fact in reference. In the second article, the 'market share' in the template is not shown in the summaries. This indicates that the model can avoid irrelevant information while extracting useful information. It further proves that the model does not completely rely on template information.

TABLE V
CASE STUDY ON GIGAWORD AND LCSTS

Source:	a woman street cleaner and her three young daughters were killed saturday when a bomb in a metal container exploded in bangladesh, police said.
Ref:	mother three daughters die in in bangladesh blast.
Template:	four die in boiler blast in bangladesh.
Transformer:	two killed in bomb blast in bangladesh.
STTR w/o loss:	four die in bomb blast in bangladesh.
STTR:	woman street cleaner three daughters die in bangladesh blast.
Source:	人民币在全球支付货币排名已由2012年1月份的第20位攀升至今年5月份的第13位，人民币支付额稳步增长，市场份额升至0.84%的新高。此前央行发布的《中国货币政策执行报告》显示，一季度银行累计办理跨境贸易人民币结算业务10039.2亿元，同比增长72.3%。 The ranking of RMB in the global payment currency has climbed from 20th in January 2012 to 13th in May this year. RMB payment has steadily increased and its market share has risen to a new high of 0.84%. The "China Monetary Policy Implementation Report" previously issued by the central bank showed that in the first quarter, banks handled RMB 10.032 billion in cross-border trade RMB settlement business, a year-on-year increase of 72.3%.
Ref:	人民币全球支付排名提升助力国际化前行。 Improvement of RMB global payment ranking helps internationalization.
Template:	人民币作为全球支付货币的市场占有率创新高。 Market share of RMB as a global payment currency hits record high.
Transformer:	一季度人民币结算业务同比增长3.7%。 RMB settlement business in the first quarter increased by 3.7% year-on-year.
STTR w/o loss:	人民币支付额升至0.84%。 RMB payment rose to 0.84%.
STTR:	人民币全球支付货币排名攀升。 Ranking of RMB global payment currencies climbs.

3) Adding the proposed loss can make the generate summaries more cosine. As in the first article, the STTR detail describes people who are 'woman street cleaner and three daughters' and remove the word 'bomb' which makes the structure of generated summary more closer to the reference. Also, in the second article, the generate summary of STTR mentioned the 'ranking' in the reference which does not appear in the other two models.

VI. CONCLUSION

In this paper, we proposed a Summarization Transformer with Template-aware Representation (STTR) which introduced template information into the transformer architecture. We extend a template encoder to extract template information and merge it into the architecture through an attention mechanism. Additionally, we proposed a document representation shifting loss to make the representation between them closer. The experiment result shows our proposed model achieves state-of-the-art on Gigaword and LCSTS datasets. The further experiment shows the effectiveness of our integration approach and proposed loss.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (Grant Nos.U1636211, 61672081, 61370126), the Beijing Advanced Innovation Center for Imaging Technology (Grant No.BAICIT-2016001), and the Fund of the State Key Laboratory of Software Development Environment (Grant No.SKLSDE-2019ZX-17).

REFERENCES

[1] Wang, Qianlong, and Jiangtao Ren, "Abstractive Summarization with Keyword and Generated Word Attention," in 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1-8. IEEE.

[2] Li, Chenliang, Weiran Xu, Si Li, and Sheng Gao, "Guiding generation for abstractive text summarization based on key information guide network," in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 2018, pp. 55-60.

[3] Wang, Li, Junlin Yao, Yunzhe Tao, Li Zhong, Wei Liu, and Qiang Du, "A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization," in Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 4453-4460.

[4] Sharma, E., Huang, L., Hu, Z., & Wang, L, "An Entity-Driven Framework for Abstractive Summarization," 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), 2019, pp. 3271-3282.

[5] Zhou, Liang, and Eduard Hovy, "Template-Filtered Headline Summarization," in Text Summarization Branches Out, 2004, pp. 56-60.

[6] Wang, Kai, Xiaojun Quan, and Rui Wang, "BiSET: Bi-directional Selective Encoding with Template for Abstractive Summarization," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 2153-2162.

[7] Cao, Ziqiang, et al, "Retrieve, rerank and rewrite: Soft template based neural summarization," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 152-161.

[8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I, "Attention is all you need," in Advances in neural information processing systems, 2017, pp. 5998-6008.

[9] Nallapati, Ramesh, Feifei Zhai, and Bowen Zhou, "SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents," in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 3075-3081.

[10] Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., & Zhao, T, "Neural Document Summarization by Jointly Learning to Score and Select Sentences," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 654-663.

[11] Rush, Alexander M., Sumit Chopra, and Jason Weston, "A Neural Attention Model for Abstractive Sentence Summarization," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 379-389.

[12] Gu, J., Lu, Z., Li, H., & Li, V. O, "Incorporating Copying Mechanism in Sequence-to-Sequence Learning," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 1631-164.

[13] See, Abigail, Peter J. Liu, and Christopher 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), 2017, pp. 1073-1083.

[14] Lin, J., Sun, X., Ma, S., & Su, Q, "Global Encoding for Abstractive Summarization," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2018, pp. 163-169.

[15] Song, K.; Wang, B.; Feng, Z.; Ren, L.; and Liu, F, "Controlling the Amount of Verbatim Copying in Abstractive Summarization," in Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020.

[16] Duan, X., Yu, H., Yin, M., Zhang, M., Luo, W., & Zhang, Y, "Contrastive Attention Mechanism for Abstractive Sentence Summarization," 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), 2019, pp. 3035-3044.

[17] You, Y., Jia, W., Liu, T., & Yang, W, "Improving abstractive document summarization with salient information modeling," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 2132-2141.

[18] Cai, Tian, et al. "Improving Transformer with Sequential Context Representations for Abstractive Text Summarization," in CCF International Conference on Natural Language Processing and Chinese Computing. Springer, Cham, 2019, pp. 512-524.

[19] Yang, M., Wang, R., Chen, K., Utiyama, M., Sumita, E., Zhang, M., & Zhao, T, "Sentence-Level Agreement for Neural Machine Translation." Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 3076-3082.

[20] Le, Quoc, and Tomas Mikolov, "Distributed representations of sentences and documents," in International conference on machine learning, 2014, pp. 1188-1196.

[21] Mikolov, T., Chen, K., Corrado, G., & Dean, J, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[22] Nallapati, R., Zhou, B., dos Santos, C., Gulçehre, Ç., & Xiang, B, "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond," in Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, 2016, pp. 280-290.

[23] Hu, Baotian, Qingcai Chen, and Fangze Zhu, "LCSTS: A Large Scale Chinese Short Text Summarization Dataset," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1967-1972.

[24] Lin, Chin-Yew, and Eduard Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, 2003, pp. 150-157.

[25] Sennrich, Rico, Barry Haddow, and Alexandra Birch, "Neural Machine Translation of Rare Words with Subword Units," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 1715-1725.

[26] Lapata, Mirella, and Regina Barzilay, "Automatic evaluation of text coherence: Models and representations," IJCAI. Vol. 5, 2005, pp. 1085-1090.

[27] Li, P., Lam, W., Bing, L., & Wang, Z, "Deep Recurrent Generative Decoder for Abstractive Text Summarization," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 2091-2100.

[28] Li, Piji, Lidong Bing, and Wai Lam, "Actor-critic based training framework for abstractive summarization," arXiv preprint arXiv:1803.11070, 2018.

[29] S. Ma, X. Sun, J. Lin, and H. Wang, "Autoencoder as assistant supervisor: Improving text representation for chinese social media text summarization," meeting of the association for computational linguistics, vol. 2, pp. 725–731, 2018.

[30] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K, "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), 2019, pp. 4171-4186.

[31] Kingma, Diederik P., and Jimmy Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.