BiSET: Bi-directional Selective Encoding with Template for Abstractive Summarization

Kai Wang
Sun Yat-sen University
wangk73@mail2.sysu.edu.cn

Xiaojun Quan[*]
Sun Yat-sen University
quanxj3@mail.sysu.edu.cn

Rui Wang
Alibaba Inc.
masi.wr@alibaba-inc.com

Abstract

The success of neural summarization models stems from the meticulous encodings of source articles. To overcome the impediments of limited and sometimes noisy training data, one promising direction is to make better use of the available training data by applying filters during summarization. In this paper, we propose a novel **Bi**-directional **S**elective **E**ncoding with **T**emplate (BiSET) model, which leverages template discovered from training data to softly select key information from each source article to guide its summarization process. Extensive experiments on a standard summarization dataset were conducted and the results show that the template-equipped BiSET model manages to improve the summarization performance significantly with a new state of the art.

1 Introduction

Abstractive summarization aims to shorten a source article or paragraph by rewriting while preserving the main idea. Due to the difficulties in rewriting long documents, a large body of research on this topic has focused on paragraph-level article summarization. Among them, sequence-to-sequence models have become the mainstream and some have achieved state-of-the-art performance (Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016). In general, the only available information for these models during decoding is simply the source article representations from the encoder and the generated words from the previous time steps (Nallapati et al., 2016; Gu et al., 2016; Lin et al., 2018), while the previous words are also generated based on the article representations. Since natural language text is complicated and verbose in nature, and training data is insufficient in size to help the models distinguish important article information from noise, sequence-to-

sequence models tend to deteriorate with the accumulation of word generation, e.g., they generate irrelevant and repeated words frequently (Koehn and Knowles, 2017).

Template-based summarization (Zhou and Hovy, 2004) is an effective approach to traditional abstractive summarization, in which a number of hard templates are manually created by domain experts, and key snippets are then extracted and populated into the templates to form the final summaries. The advantage of such approach is it can guarantee concise and coherent summaries in no need of any training data. However, it is unrealistic to create all the templates manually since this work requires considerable domain knowledge and is also labor-intensive. Fortunately, the summaries of some specific training articles can provide similar guidance to the summarization as hard templates. Accordingly, these summaries are referred to as soft templates, or templates for simplicity, in this paper.

Despite their potential in relieving the verbosity and insufficiency problems of natural language data, templates have not been exploited to full advantage. For example, Cao et al. (2018a) simply concatenated template encoding after the source article in their summarization work. To this end, we propose a Bi-directional Selective Encoding with Template (BiSET) model for abstractive sentence summarization. Our model involves a novel bi-directional selective layer with two gates to mutually select key information from an article and its template to assist with summary generation. Due to the limitations in obtaining handcrafted templates, we further propose a multi-stage process for automatic retrieval of high-quality templates from training corpus. Extensive experiments were conducted on the Gigaword dataset (Rush et al., 2015), a public dataset widely used for abstractive sentence summarization, and the results appear to

---

[*]Corresponding author.

be quite promising. Merely using the templates selected by our approach as the final summaries, our model can already achieve superior performance to some baseline models, demonstrating the effect of our templates. This may also indicate the availability of many quality templates in the corpus. Secondly, the template-equipped summarization model, BiSET, outperforms all the state-of-the-art models significantly. To evaluate the importance of the bi-directional selective layer and the two gates, we conducted an ablation study by discarding them respectively, and the results show that, while both of the gates are necessary, the template-to-article (T2A) gate tends to be more important than the article-to-template (A2T) gate. A human evaluation further validates the effectiveness of our model in generating informative, concise and readable summaries.

The contributions of this work include:

- We propose a novel bi-directional selective mechanism with two gates to mutually select important information from both article and template to assist with summary generation.

- We develop a Fast Rerank method to automatically select high-quality templates from training corpus.

- Empirical evaluations on the benchmark dataset show our model has achieved a new state of the art.

- The source code of this work has been released for future research.[1]

2 The Framework

Our framework includes three key modules: Retrieve, Fast Rerank, and BiSET. For each source article, Retrieve aims to return a few candidate templates from the training corpus. Then, the Fast Rerank module quickly identifies a best template from the candidates. Finally, BiSET mutually selects important information from the source article and the template to generate an enhanced article representation for summarization.

2.1 Retrieve

This module starts with a standard information retrieval library[2] to retrieve a small set of candidates for fine-grained filtering as Cao et al. (2018a). To do that, all non-alphabetic characters (e.g., dates)

---
[1] https://github.com/InitialBug/BiSET
[2] https://lucene.apache.org

are removed to eliminate their influence on article matching. The retrieval process starts by querying the training corpus with a source article to find a few (5 to 30) related articles, the summaries of which will be treated as candidate templates.

2.2 Fast Rerank

The above retrieval process is essentially based on superficial word matching and cannot measure the deep semantic relationship between two articles. Therefore, the Fast Rerank module is developed to identify a best template from the candidates based on their deep semantic relevance with the source article. We regard the candidate with highest relevance as the template. As illustrated in Figure 1, this module consists of a Convolution Encoder Block, a Similarity Matrix and a Pooling Layer.

Convolution Encoder Block. This block maps the input article and its candidate templates into high-level representations. The popular ways to this are either by using recurrent neural network (RNN) or a stack of convolutional neural network (CNN), while none of them are suitable for our problem. This is because a source article is usually much longer than a template, and both RNN and CNN may lead to semantic irrelevance after encodings. Instead, we implement a new convolution encoder block which includes a word embedding layer, a 1-D convolution followed by a non-linearity function, and residual connections (Gehring et al., 2017).

Formally, given word embeddings $\{e_i\}_{i=1}^{E} \in \mathbb{R}^d$ of an article, we use a 1-D convolution with kernel $\mathbf{k} \in \mathbb{R}^{2d \times kd}$ and bias $b_h \in \mathbb{R}^{2d}$ to extract the n-gram features:

$$h_i = \mathbf{k}[e_{i-k/2}, ..., e_{i+k/2}] + b_h \qquad (1)$$

where $h_i \in \mathbb{R}^{2d}$. We pad both sides of an article/template with zeros to keep fixed length. After that, we employ the gated linear unit (GLU) (Dauphin et al., 2017) as our activation function to control the proportion of information to pass through. GLU takes half the dimension of h_i as input and reduces the input dimension to d. Let $h_i = [h_i^1; h_i^2]$, where $h_i^1, h_i^2 \in \mathbb{R}^d$, we have:

$$r_i = GLU(h_i) = GLU([h_i^1; h_i^2]) = h_i^1 \otimes \sigma(h_i^2) \qquad (2)$$

where $r_i \in \mathbb{R}^d$, σ is the sigmoid function, and $\otimes$ means element-wise multiplication. To retain the original information, we add residual connections
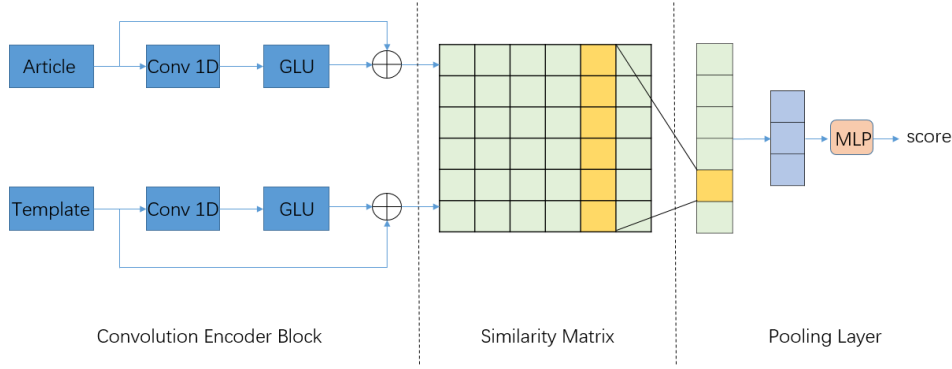
Figure 1: Overview of the Fast Rerank Module.

from the input of the convolution layer to the output of this block: $z_i = r_i + e_i$.

Similarity Matrix. The above encoder block generates a high-level representation for each source article/candidate template. Then, a similarity matrix $\mathcal{S} \in \mathbb{R}^{m \times n}$ is calculated for a given article representation, $\mathbf{S} \in \mathbb{R}^{m \times d}$, and a template representation, $\mathbf{T} \in \mathbb{R}^{n \times d}$:

$$s_{ij} = f(\mathbf{S}_i, \mathbf{T}_j) \qquad (3)$$

where f is the similarity function, and the common options for f include:

$$f(x,y) = \begin{cases} x^T y, & \text{dot product} \\ x^T W y, & \text{bilinear function} \\ \|x - y\|, & \text{Euclidean distance} \end{cases} \qquad (4)$$

Most previous work uses dot product or bilinear function (Chen et al., 2016) for the similarity, yet we find the family of Euclidean distance perform much better for our task. Therefore, we define the similarity function as:

$$f(x,y) = exp(-\|x - y\|^2) \qquad (5)$$

Pooling Layer. This layer is intended to filter out unnecessary information in the matrix $\mathcal{S}$. Before applying such pooling operations as max-pooling and k-max pooling (Kalchbrenner et al., 2014) over the similarity matrix, we note there are repeated words in the source article, which we only want to count once. For this reason, we first identify some salient weights from $\mathcal{S}$:

$$q = max_{column}(\mathcal{S}) \qquad (6)$$

where max_{column} is a column-wise maximum function. We then apply k-max pooling over q to

select k most important weights, $p \in \mathbb{R}^k$. Finally, we apply a two-layer feed-forward network to output a similarity score for the source article and the candidate template:

$$p = k\text{-}max(q) \qquad (7)$$
$$a = ReLU(\mathbf{W}_a p + b_1) \qquad (8)$$
$$s = \sigma(\mathbf{W}_s a + b_2) \qquad (9)$$

2.3 Traditional Methodologies

In this section, we explore three traditional approaches to taking advantage of the templates for summarization. They share the same encoder and decoder layers, but own different interaction layers for combination of a source article and template. The encoder layer uses a standard bi-directional RNN (BiRNN) to separately encode the source article and the template into hidden states h_i^s and h_j^t.

Concatenation. This approach directly concatenates the hidden state, $\{h_i^t\}_{i=1}^N$, of a template after the article representation, $\{h_i^s\}_{i=1}^M$, to form a new article representation, $\{z_i^s\}_{i=1}^{M+N}$. This approach is similar to R^3Sum (Cao et al., 2018a) but uses our Fast Rerank and summary generation modules.

Concatenation+Self-Attention. This approach adds a multi-head self-attention (Vaswani et al., 2017) layer with 4 heads on the basis of the above direct concatenation.

DCN Attention. Initially introduced for machine reading comprehension (Seo et al., 2017), this interaction approach is employed here to create template-aware article representations. First, we compute a similarity matrix, $\mathcal{S} \in \mathbb{R}^{m \times n}$, for each pair of article and template words by $s_{ij} = \mathbf{W}_0[h_i^s; h_j^t; h_i^s \otimes h_j^t]$, where ';' is the concatenation operation. We then normalize each row and col-
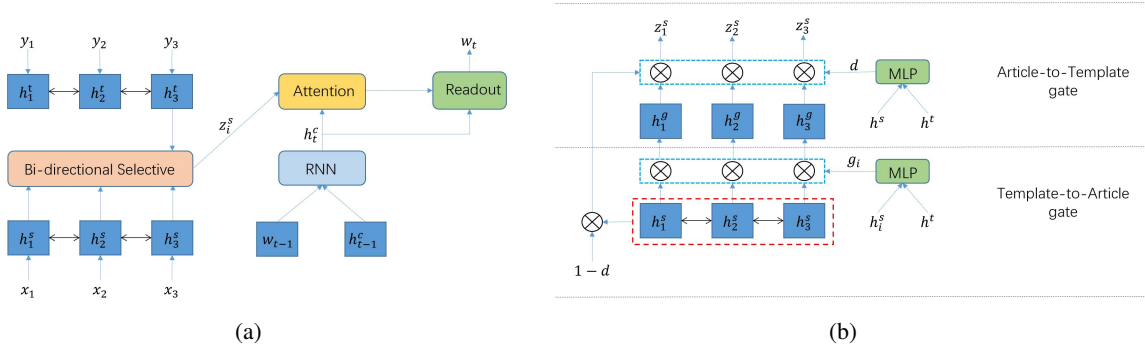
Figure 2: The structure of the proposed model: (a) the Bi-Directional Selective Encoding with Template model (BiSET) and (b) the bi-directional selective layer.

umn of $\mathcal{S}$ by softmax, giving rise to two new matrices $\overline{\mathcal{S}}$ and $\overline{\overline{\mathcal{S}}}$. After that, the Dynamic Coattention Network (DCN) attention is applied to compute the bi-directional attention: $\mathbf{A} = \overline{\mathcal{S}} \cdot h^t$ and $\mathbf{B} = \overline{\mathcal{S}} \cdot \overline{\overline{\mathcal{S}}}^T \cdot h^s$, where $\mathbf{A}$ denotes article-to-template attention and $\mathbf{B}$ is template-to-article attention. Finally, we obtain the template-aware article representation $\{z_i^s\}_{i=1}^M$:

$$z_i^s = [h_i^s; \mathbf{A}_i; h_i^s \otimes \mathbf{A}_i; h_i^s \otimes \mathbf{B}_i] \quad (10)$$

2.4 BiSET

Inspired by the research in machine reading comprehension (Seo et al., 2017) and selective mechanism (Zhou et al., 2017), we propose a novel **Bi**-directional **S**elective **E**ncoding with **T**emplate (BiSET) model for abstractive sentence summarization. The core idea behind BiSET is to involve templates to assist with article representation and summary generation. As shown in Figure 2, BiSET contains two selective gates: Template-to-Article (T2A) gate and Article-to-Template (A2T) gate. The role of T2A is to use a template to filter the source article representation:

$$g_i = \sigma(\mathbf{W}_{sh} h_i^s + \mathbf{W}_{th} h^t + b_s) \quad (11)$$
$$h_i^g = h_i^s \otimes g_i \quad (12)$$

where h^t is the concatenation of the last forward hidden state, $\overrightarrow{h_n^t}$, and the first backward hidden state, $\overleftarrow{h_1^t}$, of the template.

On the other hand, the purpose of A2T is to control the proportion of h^g in the final article representation. We assume the source article is credible and use its representation h^s together with h^t to calculate a confidence degree, where h^s is obtained in a similar way as h^t. The confidence de-

gree d is computed by:

$$d = \sigma((h^s)^T \mathbf{W}_d h^t + b_d) \quad (13)$$

The final source article representation is calculated as the weighted sum of h_i^s and h_i^g:

$$z_i^s = d h_i^g + (1-d) h_i^s \quad (14)$$

which allows a flexible manner for template incorporation and helps to resist errors when low-quality templates are given.

The decoder layer. This layer includes an ordinary RNN decoder (Luong et al., 2015). At each time step t, the decoder reads the word w_{t-1} and hidden state h_{t-1}^c generated in the previous step, and gives a new hidden state for the current step:

$$h_t^c = RNN(w_{t-1}, h_{t-1}^c) \quad (15)$$

where the hidden state is initialized with the original source article representation, h^s. We then compute the attention between h_t^c and the final article representation z^s to obtain a context vector c_t:

$$\varepsilon_{t,i} = (z_i^s)^T \mathbf{W}_c h_t^c \quad (16)$$
$$\alpha_{t,i} = \frac{exp(\varepsilon_{t,i})}{\sum_{i=1}^M exp(\varepsilon_{t,i})} \quad (17)$$
$$c_t = \sum_{i=1}^M \alpha_{t,i} z_i^s \quad (18)$$

After that, a simple concatenation layer is used to combine the hidden state h_t^c and the context vector c_t into a new hidden state h_t^a:

$$h_t^a = tanh(\mathbf{W}_{ha}[c_t; h_t^c]) \quad (19)$$

which will be mapped to a new representation of

vocabulary size and fed through a softmax layer to output the target word distribution:

$$p(w_t|w_1, ..., w_{t-1}) = softmax(\mathbf{W}_p h_t^a) \quad (20)$$

2.5 Training

The Retrieve module involves an unsupervised process with traditional indexing and retrieval techniques. For Fast Rerank, since there is no ground truth available, we use ROUGE-1[3] (Lin and Hovy, 2003) to evaluate the saliency of a candidate template with respect to the gold summary of current source article. Therefore, the loss function is defined as:

$$L_r(\theta) = -\frac{1}{N} \sum_{i=1}^{N} [s^* \log s + (1 - s^*) \log(1 - s)]$$
$$(21)$$

where s is a score predicted by Equation 9, and N is the product of the training set size, D, and the number of retrieved templates for each article.

For the BiSET module, the loss function is chosen as the negative log-likelihood between the generated summary, w, and the true summary, w^*:

$$L_w(\theta) = -\frac{1}{D} \sum_{i=1}^{D} \sum_{j=1}^{L} \log p(w_j^{*(i)}|w_{j-1}^{(i)}, x^{(i)}, y^{(i)})$$
$$(22)$$

where L is the length of the true summary, θ contains all the trainable variables, and x and y denote the source article and the template, respectively.

3 Experiments

In this section, we introduce our evaluations on a standard dataset.

3.1 Dataset and Implementation

The dataset used for evaluation is Annotated English Gigaword (Napoles et al., 2012), a parallel corpus formed by pairing the first sentence of an article with its headline. For a fair comparison, we use the version preprocessed by Rush et al. (2015)[4] as previous work.

During training, both the Fast Rerank and BiSET modules have a batch size of 64 with the Adam optimizer (Kingma and Ba, 2015). We also apply grad clipping (Pascanu et al., 2013) with a

range of [-5,5]. The differences of the two modules in settings are listed below.

Fast Rerank. We set the size of word embeddings to 300, the convolution encoder block number to 1, and the kernel size of CNN to 3. The weights are shared between the article and template encoders. The k of k-max pooling is set to 10. L2 weight decay with $\lambda = 3 \times 10^{-6}$ is performed over all trainable variables. The initial learning rate is 0.001 and multiplied by 0.1 every 10K steps. Dropout between layers is applied.

BiSET. A two-layer BiLSTM is used as the encoder, and another two-layer LSTM as the decoder. The sizes of word embeddings and LSTM hidden states are both set to 500. We only apply dropout in the LSTM stack with a rate of 0.3. The learning rate is set to 0.001 for the first 50K steps and halved every 10K steps. Beam search with size 5 is applied to search for optimal answers.

3.2 Evaluation Metrics

Following previous work (Nallapati et al., 2016; Zhou et al., 2017; Cao et al., 2018a), we use the standard F1 scores of ROUGE-1, ROUGE-2 and ROUGE-L (Lin and Hovy, 2003) to evaluate the selected templates and generated summaries, where the official ROUGE script[5] is applied. We employ the normalized discounted cumulative gain (NDCG) (Järvelin and Kekäläinen, 2002) from information retrieval to evaluate the Fast Rerank module.

4 Results and Analysis

In this section, we report our experimental results with thorough analysis and discussions.

4.1 Performance of Retrieve

The Retrieve module is intended to narrow down the search range for a best template. We evaluated this module by considering three types of templates: (a) **Random** means a randomly selected summary from the training corpus; (b) **Retrieve-top** is the highest-ranked summary by Retrieve; (c) **N-Optimal** means among the N top search results, the template is specified as the summary with largest ROUGE score with gold summary.

As the results show in Table 1, randomly selected templates are totally irrelevant and unhelpful. When they are replaced by the Retrieve-top

---

[3]We also tried ROUGE-2 and ROUGE-L, but ROUGE-1 shows to be more suitable.

[4]https://github.com/harvardnlp/sent-summary

[5]The ROUGE evaluation option: -m -n 2 -w 1.2
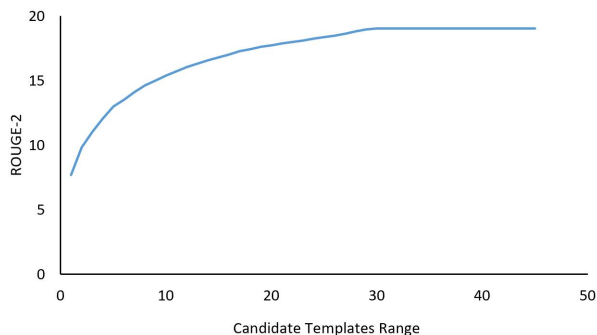
Figure 3: Quality of candidate templates under different ranges.

Figure 4: Quality of rankings given by Fast Rerank.

templates, the results improve apparently, demonstrating the relatedness of top-ranked summaries to gold summaries. Furthermore, when the N-Optimal templates are used, additional improvements can be observed as N grows. This trend is also confirmed by Figure 3, in which the ROUGE scores increase before 30 and stabilize afterwards. These results suggest that the ranges given by Retrieve indeed help to find quality templates.

Type	ROUGE-1	ROUGE-2	ROUGE-L
Random	2.58	0.00	2.48
Retrieve-top	23.46	7.67	20.94
5-Optimal	32.69	11.74	28.71
10-Optimal	35.90	13.32	31.42
15-Optimal	37.82	16.79	34.08
20-Optimal	38.92	17.72	34.94
30-Optimal	40.49	19.01	36.10

Table 1: Performance of different types of templates.

4.2 Fast Rerank

As mentioned before, the role of Fast Rerank is to re-rank the initial search results and return a best template for summarization. To examine the effect of this module, we studied its ranking quality under different ranges as in Section 4.1. The original rankings by Retrieve are presented for comparison with the NDCG metric. We regard the ROUGE-2 score of each candidate template with the reference summary as the ground truth. As shown in Figure 4, Fast Rerank consistently provides enhanced rankings over the original.

4.3 Interaction Approaches

In Section 2.3, we also explored three alternative approaches to integrating an article with its template. The results are shown in Table 2, from which we can note that none of these approaches help yield satisfactory performance. Even though DCN
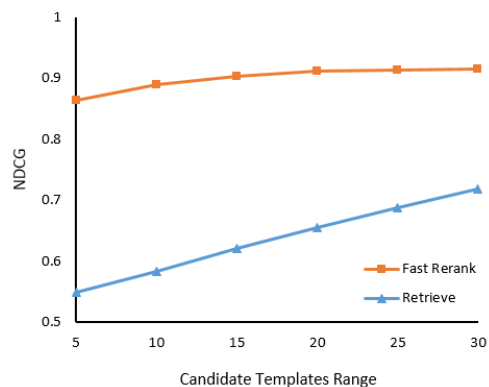
Attention works impressively in machine reading comprehension, it performs even worse in this task than the simple concatenation. We conjecture the reason is that the DCN Attention attempts to fuse the template information into an article as in machine reading comprehension, rather than selects key information from the two to form an enhanced article representation.

Interaction method	ROUGE-1	ROUGE-2	ROUGE-L
Concatenation	32.26	15.30	30.19
Concate+multi self-att	33.15	15.93	31.21
DCN Attention	31.53	13.77	27.96
Bi-selective layer	39.11	19.78	36.87

Table 2: Results of different interaction approaches.

4.4 BiSET

The overall performance of all the studied models is shown in Table 3. The results show that our model significantly outperforms all the baseline models and sets a new state of the art for abstractive sentence summarization. To evaluate the impact of templates on our model, we also implemented BiSET with two other types of templates: randomly-selected templates and best templates identified by Fast Rank under different ranges. As shown in Table 4, the performance of our model improves constantly with the improvement of template quality (larger ranges lead to better chances for good templates). Even with randomly-selected templates, our model still works with stable performance, demonstrating its robustness.

4.5 Speed Comparison

Our model is designed for both accuracy and efficiency. Due to the parallelizable nature of CNN, the Fast Rerank module only takes about 30 minutes for training and 3 seconds for inference on

2158

Model	ROUGE-1	ROUGE-2	ROUGE-L
ABS[‡] (Rush et al., 2015)	29.55	11.32	26.42
ABS+[‡] (Rush et al., 2015)	29.78	11.89	26.97
RAS-Elman[‡] (Chopra et al., 2016)	33.78	15.97	31.15
Featseq2seq[‡] (Nallapati et al., 2016)	32.67	15.59	30.64
Open-NMT[‡] (Klein et al., 2017)	34.07	16.35	31.78
SEASS[‡] (Zhou et al., 2017)	36.15	17.54	33.63
S2S+CGU[‡] (Lin et al., 2018)	36.30	18.00	33.80
FTSum[‡] (Cao et al., 2018b)	37.27	17.65	34.24
R^3Sum[‡] (Cao et al., 2018a)	37.04	19.03	34.46
BiSET	**39.11**	**19.78**	**36.87**

Table 3: Performance of all the models, where results marked with [‡] are taken from the corresponding papers.

Template Type	ROUGE-1	ROUGE-2	ROUGE-L
Random	33.85	15.83	31.14
5-rerank	37.69	18.62	34.38
10-rerank	38.34	19.35	34.97
20-rerank	38.89	19.64	36.67
30-rerank	39.11	19.78	36.87

Table 4: Performance of BiSET with different types of templates, where **Random** means randomly-selected templates, and **N-rerank** denotes the best templates reranked by Fast Rerank under range N.

the whole test set. The BiSET model takes about 8 hours for training (GPU:GTX 1080), 6 times faster than R^3Sum (Cao et al., 2018a)[6].

4.6 Ablation Study

The purpose of this study is to examine the roles of the bi-directional selective layer and its two gates. Firstly, we removed the selective layer and replaced it with the direct concatenation of an article with its template representation. As the results show in Table 5, the model performs even worse than some ordinary sequence-to-sequence models in Table 3. The reason might be that templates would overwhelm the original article representations and become noise after concatenation. Then, we removed the Template-to-Article (T2A) gate, and as a result the model shows a great decline in performance, indicating the importance of templates in article representations. Finally, when we removed the Article-to-Template (A2T) gate, whose role is to control the weight of T2A in article representations, only a small performance decline is observed. This may suggest that the T2A gate alone can already capture most of the important article information, while A2T plays some supplemental role.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Concatenation	32.26	15.30	30.19
BiSET without T2A	34.51	16.55	31.17
BiSET without A2T	39.02	19.21	36.02
BiSET(full)	39.11	19.78	36.87

Table 5: ROUGE F1 scores of ablated models.

4.7 Human Evaluation

We then carried out a human evaluation to evaluate the generated summaries from another perspective. Our evaluators include 8 graduate students and 4 senior undergraduates, while the dataset is 100 randomly-selected articles from the test set. Each sample in this dataset also includes: 1 reference summary, 5 summaries generated by Open-NMT[7] (Klein et al., 2017), R^3Sum[8] (Cao et al., 2018a) and BiSET under three settings, respectively, and 3 randomly-selected summaries for trapping. We asked the evaluators to independently rate each summary on a scale of 1 to 5, with respect to its quality in *informativity*, *conciseness*, and *readability*. While collecting the results, we rejected the samples in which more than half evaluators rate the *informativity* of the reference summary below 3. We also rejected the samples in which the *informativity* of a randomly-selected summary is scored higher than 3. Finally, we obtained 43 remaining samples and calculated an average score for each aspect. As the results show in Table 6, our model not only performs much better than the baselines, it also shows quite comparable performance with the reference summaries.

Model	Info	Concise	Read
R^3Sum	3.30	3.83	3.90
Open-NMT	3.26	3.69	3.86
BiSET(random template)	3.09	3.69	3.71
BiSET(without A2T)	3.24	3.75	3.72
BiSET(best template)	3.35	**3.98**	**3.93**
Reference	**3.55**	3.91	3.89

Table 6: Results of human evaluation.

In Table 7 we present two real examples, which show the templates found by our model are indeed related to the source articles, and with their aid, our model succeeds to keep the main content of the source articles for summarization while discarding unrelated words like 'US' and 'Olympic Games'.

---

[6]It takes about 2 days for training.

[7]https://github.com/OpenNMT/OpenNMT-py
[8]http://www4.comp.polyu.edu.hk/~cszqcao/

Source	factory orders for manufactured goods rose #.# percent in September, the commerce department said here Thursday.
Ref	September factory orders up #.# percent.
Temp	January factory orders in US up #.# percent.
BiSET	factory orders up #.# percent in September.

Source	some #.# billion people worldwide are expected to watch Germany face Costa Rica on television at the opening match of football's World Cup, German public broadcaster zdf said Thursday.
Ref	#.# billion tv viewers expected for opening World Cup match.
Temp	billions around world watch the Olympic Games opening ceremony.
BiSET	#.# billions around world expected to watch World Cup.

Table 7: Examples of the generated templates and summaries by our model. '#' refers to masked numbers.

5 Related Work

Abstractive sentence summarization, a task analogous to headline generation or sentence compression, aims to generate a brief summary given a short source article. Early studies in this problem mainly focus on statistical or linguistic-rule-based methods, including those based on extractive and compression (Jing and McKeown, 2000; Knight and Marcu, 2002; Clarke and Lapata, 2010), templates (Zhou and Hovy, 2004) and statistical machine translation (Banko et al., 2000).

The advent of large-scale summarization corpora accelerates the development of various neural network methods. Rush et al. (2015) first applied an attention-based sequence-to-sequence model for abstractive summarization, which includes a convolutional neural network (CNN) encoder and a feed-forward network decoder. Chopra et al. (2016) replaced the decoder with a recurrent neural network (RNN). Nallapati et al. (2016) further changed the sequence-to-sequence model to a fully RNN-based model. Besides, Gu et al. (2016) found that this task benefits from copying words from the source articles and proposed the CopyNet correspondingly. With a similar purpose, Gulcehre et al. (2016) proposed to use a switch gate to control when to copy from the source article and when to generate from the vocabulary. Zhou et al. (2017) employed a selective gate to filter out unimportant information when encoding.

Some other work attempts to incorporate external knowledge for abstractive summarization. For example, Nallapati et al. (2016) proposed to en-

rich their encoder with handcrafted features such as named entities and part-of-speech (POS) tags. Guu et al. (2018) also attempted to encode human-written sentences to improve neural text generation. Similar to our work, Cao et al. (2018a) proposed to retrieve a related summary from the training set as soft template to assist with the summarization. However, their approach tends to oversimplify the role of the template, by directly concatenating a template after the source article encoding. In contrast, our bi-directional selective mechanism exhibits a novel attempt to selecting key information from the article and the template in a mutual manner, offering greater flexibility in using the template.

6 Conclusion

In this paper, we presented a novel **Bi**-directional **S**elective **E**ncoding with **T**emplate (BiSET) model for abstractive sentence summarization. To counteract the verbosity and insufficiency of training data, we proposed to retrieve high-quality existing summaries as templates to assist with source article representations through an ingenious bidirectional selective layer. The enhanced article representations are expected to contribute towards better summarization eventually. We also developed the corresponding retrieval and re-ranking modules for obtaining quality templates. Extensive evaluations were conducted on a standard benchmark dataset and experimental results show that our model can quickly pick out high-quality templates from the training corpus, laying key foundation for effective article representations and summary generations. The results also show that our model outperforms all the baseline models and sets a new state of the art. An ablation study validates the role of the bi-directional selective layer, and a human evaluation further proves that our model can generate informative, concise, and readable summaries.

7 Acknowledgement

References

Michele Banko, Vibhu O. Mittal, and Michael J. Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 318–325.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018a. 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)*, volume 1, pages 152–161.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018b. Faithful to the original: Fact aware neural abstractive summarization. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2358–2367.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

James Clarke and Mirella Lapata. 2010. Discourse constraints for document compression. *Computational Linguistics*, 36(3):411–441.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 933–941.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. 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)*, volume 1, pages 1631–1640.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 140–149.

Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association of Computational Linguistics*, 6:437–450.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Hongyan Jing and Kathleen R. McKeown. 2000. Cut and paste based text summarization. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 178–185.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 655–665.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Chin Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 71–78.

Junyang Lin, Sun Xu, Shuming Ma, and Qi Su. 2018. Global encoding for abstractive summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 163–169.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Joint Workshop on Automatic Knowledge Base Construction and Web-Scale Knowledge Extraction*, pages 95–100.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017*.

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

Liang Zhou and Eduard Hovy. 2004. Template-filtered headline summarization. *Text Summarization Branches Out*.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1095–1104.