Retrieve, Rerank and Rewrite: Soft Template Based Neural Summarization

Ziqiang Cao[1,2] **Wenjie Li**[1,2] **Furu Wei**[3] **Sujian Li**[4]
[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[2]Hong Kong Polytechnic University Shenzhen Research Institute, China
[3]Microsoft Research, Beijing, China
[4]Key Laboratory of Computational Linguistics, Peking University, MOE, China
{cszqcao, cswjli}@comp.polyu.edu.hk
fuwei@microsoft.com lisujian@pku.edu.cn

Abstract

Most previous seq2seq summarization systems purely depend on the source text to generate summaries, which tends to work unstably. Inspired by the traditional template-based summarization approaches, this paper proposes to use existing summaries as soft templates to guide the seq2seq model. To this end, we use a popular IR platform to Retrieve proper summaries as candidate templates. Then, we extend the seq2seq framework to jointly conduct template Reranking and template-aware summary generation (Rewriting). Experiments show that, in terms of informativeness, our model significantly outperforms the state-of-the-art methods, and even soft templates themselves demonstrate high competitiveness. In addition, the import of high-quality external summaries improves the stability and readability of generated summaries.

1 Introduction

The exponentially growing online information has necessitated the development of effective automatic summarization systems. In this paper, we focus on an increasingly intriguing task, i.e., abstractive sentence summarization (Rush et al., 2015a), which generates a shorter version of a given sentence while attempting to preserve its original meaning. It can be used to design or refine appealing headlines. Recently, the application of the attentional sequence-to-sequence (seq2seq) framework has attracted growing attention and achieved state-of-the-art performance on this task (Rush et al., 2015a; Chopra et al., 2016; Nallapati et al., 2016).

Most previous seq2seq models purely depend on the source text to generate summaries. However, as reported in many studies (Koehn and Knowles, 2017), the performance of a seq2seq model deteriorates quickly with the increase of the length of generation. Our experiments also show that seq2seq models tend to "lose control" sometimes. For example, 3% of summaries contain less than 3 words, while there are 4 summaries repeating a word for even 99 times. These results largely reduce the informativeness and readability of the generated summaries. In addition, we find seq2seq models usually focus on copying source words in order, without any actual "summarization". Therefore, we argue that, the free generation based on the source sentence is not enough for a seq2seq model.

Template based summarization (e.g., Zhou and Hovy (2004)) is a traditional approach to abstractive summarization. In general, a template is an incomplete sentence which can be filled with the input text using the manually defined rules. For instance, a concise template to conclude the stock market quotation is: *[REGION] shares [open/close] [NUMBER] percent [lower/higher]*, e.g., "hong kong shares close #.# percent lower". Since the templates are written by humans, the produced summaries are usually fluent and informative. However, the construction of templates is extremely time-consuming and requires a plenty of domain knowledge. Moreover, it is impossible to develop all templates for summaries in various domains.

Inspired by retrieve-based conversation systems (Ji et al., 2014), we assume the golden summaries of the similar sentences can provide a reference point to guide the input sentence summarization process. We call these existing summaries **soft templates** since no actual rules are

needed to build new summaries from them. Due to the strong rewriting ability of the seq2seq framework (Cao et al., 2017a), in this paper, we propose to combine the seq2seq and template based summarization approaches. We call our summarization system Re^{3}Sum, which consists of three modules: **Re**trieve, **Re**rank and **Re**write. We utilize a widely-used Information Retrieval (IR) platform to find out candidate soft templates from the training corpus. Then, we extend the seq2seq model to jointly learn template saliency measurement (Rerank) and final summary generation (Rewrite). Specifically, a Recurrent Neural Network (RNN) encoder is applied to convert the input sentence and each candidate template into hidden states. In Rerank, we measure the informativeness of a candidate template according to its hidden state relevance to the input sentence. The candidate template with the highest predicted informativeness is regarded as the actual soft template. In Rewrite, the summary is generated according to the hidden states of both the sentence and template.

We conduct extensive experiments on the popular Gigaword dataset (Rush et al., 2015b). Experiments show that, in terms of informativeness, Re^{3}Sum significantly outperforms the state-of-the-art seq2seq models, and even soft templates themselves demonstrate high competitiveness. In addition, the import of high-quality external summaries improves the stability and readability of generated summaries.

The contributions of this work are summarized as follows:

- We propose to introduce soft templates as additional input to improve the readability and stability of seq2seq summarization systems. Code and results can be found at http://www4.comp.polyu.edu.hk/~cszqcao/
- We extend the seq2seq framework to conduct template reranking and template-aware summary generation simultaneously.
- We fuse the popular IR-based and seq2seq-based summarization systems, which fully utilize the supervisions from both sides.

2 Method

As shown in Fig. 1, our summarization system consists of three modules, i.e., Retrieve, Rerank and Rewrite. Given the input sentence $\mathbf{x}$, the Retrieve module filters candidate soft templates $\mathbf{C} = \{\mathbf{r}_i\}$ from the training corpus. For validation and test, we regard the candidate template with the highest predicted saliency (a.k.a informativeness) score as the actual soft template $\mathbf{r}$. For training, we choose the one with the maximal actual saliency score in $\mathbf{C}$, which speeds up convergence and shows no obvious side effect in the experiments.

Then, we jointly conduct reranking and rewriting through a shared encoder. Specifically, both the sentence $\mathbf{x}$ and the soft template $\mathbf{r}$ are converted into hidden states with a RNN encoder. In the Rerank module, we measure the saliency of $\mathbf{r}$ according to its hidden state relevance to $\mathbf{x}$. In the Rewrite module, a RNN decoder combines the hidden states of $\mathbf{x}$ and $\mathbf{r}$ to generate a summary $\mathbf{y}$. More details will be described in the rest of this section

2.1 Retrieve

The purpose of this module is to find out candidate templates from the training corpus. We assume that similar sentences should hold similar summary patterns. Therefore, given a sentence $\mathbf{x}$, we find out its analogies in the corpus and pick their summaries as the candidate templates. Since the size of our dataset is quite large (over 3M), we leverage the widely-used Information Retrieve (IR) system Lucene[1] to index and search efficiently. We keep the default settings of Lucene[2] to build the IR system. For each input sentence, we select top 30 searching results as candidate templates.

2.2 Jointly Rerank and Rewrite

To conduct template-aware seq2seq generation (rewriting), it is a necessary step to encode both the source sentence $\mathbf{x}$ and soft template $\mathbf{r}$ into hidden states. Considering that the matching networks based on hidden states have demonstrated the strong ability to measure the relevance of two pieces of texts (e.g., Chen et al. (2016)), we propose to jointly conduct reranking and rewriting through a shared encoding step. Specifically, we employ a bidirectional Recurrent Neural Network (BiRNN) encoder (Cho et al., 2014) to read $\mathbf{x}$ and $\mathbf{r}$. Take the sentence $\mathbf{x}$ as an example. Its hidden

[1] https://lucene.apache.org/
[2] TextField with EnglishAnalyzer

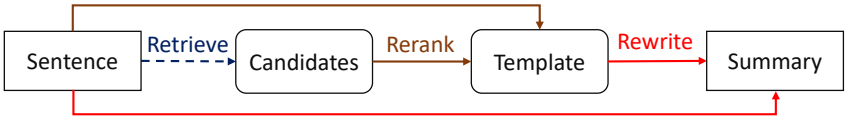

Figure 1: Flow chat of the proposed method. We use the dashed line for Retrieve since there is an IR system embedded.

state of the forward RNN at timestamp i can be represented by:

$$\overrightarrow{\mathbf{h}}_i^x = \text{RNN}(x_i, \overrightarrow{\mathbf{h}}_{i-1}^x) \quad (1)$$

The BiRNN consists of a forward RNN and a backward RNN. Suppose the corresponding outputs are $[\overrightarrow{\mathbf{h}}_1^x; \cdots; \overrightarrow{\mathbf{h}}_{-1}^x]$ and $[\overleftarrow{\mathbf{h}}_1^x; \cdots; \overleftarrow{\mathbf{h}}_{-1}^x]$, respectively, where the index "-1" stands for the last element. Then, the composite hidden state of a word is the concatenation of the two RNN representations, i.e., $\mathbf{h}_i^x = [\overrightarrow{\mathbf{h}}_i^x; \overleftarrow{\mathbf{h}}_i^x]$. The entire representation for the source sentence is $[\mathbf{h}_1^x; \cdots; \mathbf{h}_{-1}^x]$. Since a soft template $\mathbf{r}$ can also be regarded as a readable concise sentence, we use the same BiRNN encoder to convert it into hidden states $[\mathbf{h}_1^r; \cdots; \mathbf{h}_{-1}^r]$.

2.2.1 Rerank

In Retrieve, the template candidates are ranked according to the text similarity between the corresponding indexed sentences and the input sentence. However, for the summarization task, we expect the soft template $\mathbf{r}$ resembles the actual summary $\mathbf{y}^*$ as much as possible. Here we use the widely-used summarization evaluation metrics ROUGE (Lin, 2004) to measure the actual saliency $s^*(\mathbf{r}, \mathbf{y}^*)$ (see Section 3.2). We utilize the hidden states of $\mathbf{x}$ and $\mathbf{r}$ to predict the saliency s of the template. Specifically, we regard the output of the BiRNN as the representation of the sentence or template:

$$\mathbf{h}_x = [\overleftarrow{\mathbf{h}}_1^x; \overrightarrow{\mathbf{h}}_{-1}^x] \quad (2)$$

$$\mathbf{h}_r = [\overleftarrow{\mathbf{h}}_1^r; \overrightarrow{\mathbf{h}}_{-1}^r] \quad (3)$$

Next, we use Bilinear network to predict the saliency of the template for the input sentence.

$$s(\mathbf{r}, \mathbf{x}) = \text{sigmoid}(\mathbf{h}_r \mathbf{W}_s \mathbf{h}_x^{\mathbf{T}} + b_s), \quad (4)$$

where $\mathbf{W}_s$ and b_s are parameters of the Bilinear network, and we add the sigmoid activation function to make the range of s consistent with the actual saliency s^*. According to Chen et al. (2016), Bilinear outperforms multi-layer forward neural networks in relevance measurement. As shown later, the difference of s and s^* will provide additional supervisions for the seq2seq framework.

2.2.2 Rewrite

The soft template $\mathbf{r}$ selected by the Rerank module has already competed with the state-of-the-art method in terms of ROUGE evaluation (see Table 4). However, $\mathbf{r}$ usually contains a lot of named entities that does not appear in the source (see Table 5). Consequently, it is hard to ensure that the soft templates are faithful to the input sentences. Therefore, we leverage the strong rewriting ability of the seq2seq model to generate more faithful and informative summaries. Specifically, since the input of our system consists of both the sentence and soft template, we use the concatenation function[3] to combine the hidden states of the sentence and template:

$$\mathbf{H}_c = [\mathbf{h}_1^x; \cdots; \mathbf{h}_{-1}^x; \mathbf{h}_1^r; \cdots; \mathbf{h}_{-1}^r] \quad (5)$$

The combined hidden states are fed into the prevailing attentional RNN decoder (Bahdanau et al., 2014) to generate the decoding hidden state at the position t:

$$\mathbf{s}_t = \text{Att-RNN}(\mathbf{s}_{t-1}, y_{t-1}, \mathbf{H}_c), \quad (6)$$

where y_{t-1} is the previous output summary word. Finally, a $softmax$ layer is introduced to predict the current summary word:

$$\mathbf{o_t} = softmax(\mathbf{s}_t \mathbf{W}_o), \quad (7)$$

where $\mathbf{W}_o$ is a parameter matrix.

2.3 Learning

There are two types of costs in our system. For Rerank, we expect the predicted saliency $s(\mathbf{r}, \mathbf{x})$ close to the actual saliency $s^*(\mathbf{r}, \mathbf{y}^*)$. Therefore,

[3] We also attempted complex combination approaches such as the gate network Cao et al. (2017b) but failed to achieve obvious improvement. We assume the Rerank module has partially played the role of the gate network.

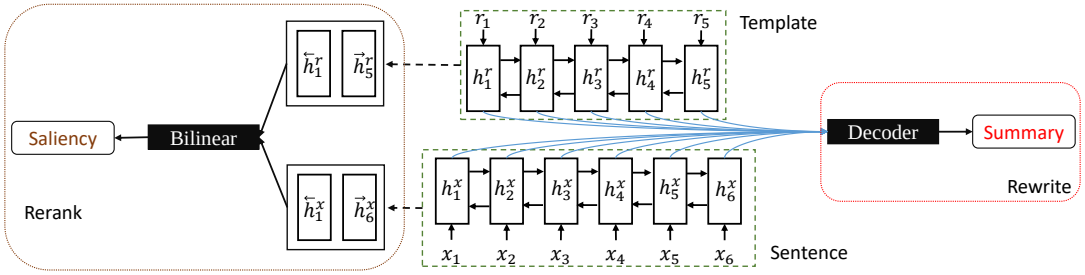

Figure 2: Jointly Rerank and Rewrite

we use the cross entropy (CE) between s and s^* as the loss function:

$$J_R(\theta) = \mathrm{CE}(s(\mathbf{r}, \mathbf{x}), s^*(\mathbf{r}, \mathbf{y}^*)) \quad (8)$$
$$= -s^* \log s - (1 - s^*) \log(1 - s),$$

where θ stands for the model parameters. For Rewrite, the learning goal is to maximize the estimated probability of the actual summary $\mathbf{y}^*$. We adopt the common negative log-likelihood (NLL) as the loss function:

$$J_G(\theta) = -\log(p(\mathbf{y}^*|\mathbf{x}, \mathbf{r})) \quad (9)$$
$$= -\sum_t \log(\mathbf{o}_t[y_t^*])$$

To make full use of supervisions from both sides, we combine the above two costs as the final loss function:

$$J(\theta) = J_R(\theta) + J_G(\theta) \quad (10)$$

We use mini-batch Stochastic Gradient Descent (SGD) to tune model parameters. The batch size is 64. To enhance generalization, we introduce dropout (Srivastava et al., 2014) with probability $p = 0.3$ for the RNN layers. The initial learning rate is 1, and it will decay by 50% if the generation loss does not decrease on the validation set.

3 Experiments

3.1 Datasets

We conduct experiments on the Annotated English Gigaword corpus, as with (Rush et al., 2015b). This parallel corpus is produced by pairing the first sentence in the news article and its headline as the summary with heuristic rules. All the training, development and test datasets can be downloaded at https://github.com/harvardnlp/sent-summary. The statistics of the Gigaword corpus is presented in Table 1.

Dataset	Train	Dev.	Test
Count	3.8M	189k	1951
AvgSourceLen	31.4	31.7	29.7
AvgTargetLen	8.3	8.3	8.8
COPY(%)	45	46	36

Table 1: Data statistics for English Gigaword. AvgSourceLen is the average input sentence length and AvgTargetLen is the average summary length. COPY means the copy ratio in the summaries (without stopwords).

3.2 Evaluation Metrics

We adopt ROUGE (Lin, 2004) for automatic evaluation. ROUGE has been the standard evaluation metric for DUC shared tasks since 2004. It measures the quality of summary by computing the overlapping lexical units between the candidate summary and actual summaries, such as unigram, bi-gram and longest common subsequence (LCS). Following the common practice, we report ROUGE-1 (uni-gram), ROUGE-2 (bi-gram) and ROUGE-L (LCS) F1 scores[4] in the following experiments. We also measure the actual saliency of a candidate template $\mathbf{r}$ with its combined ROUGE scores given the actual summary $\mathbf{y}^*$:

$$s^*(\mathbf{r}, \mathbf{y}^*) = \mathrm{RG}(\mathbf{r}, \mathbf{y}^*) + \mathrm{RG}(\mathbf{r}, \mathbf{y}^*), \quad (11)$$

where "RG" stands for ROUGE for short.

ROUGE mainly evaluates informativeness. We also introduce a series of metrics to measure the summary quality from the following aspects:

LEN_DIF The absolute value of the length difference between the generated summaries and the actual summaries. We use mean value ± standard deviation to illustrate this item. The average value partially reflects the readability and informativeness, while the standard deviation links to stability.

[4]We use the ROUGE evaluation option: -m -n 2 -w 1.2

LESS_3 The number of the generated summaries, which contains less than three tokens. These extremely short summaries are usually unreadable.

COPY The proportion of the summary words (without stopwords) copied from the source sentence. A seriously large copy ratio indicates that the summarization system pays more attention to compression rather than required abstraction.

NEW_NE The number of the named entities that do not appear in the source sentence or actual summary. Intuitively, the appearance of new named entities in the summary is likely to bring unfaithfulness. We use Stanford CoreNLP (Manning et al., 2014) to recognize named entities.

3.3 Implementation Details

We use the popular seq2seq framework OpenNMT[5] as the starting point. To make our model more general, we retain the default settings of OpenNMT to build the network architecture. Specifically, the dimensions of word embeddings and RNN are both 500, and the encoder and decoder structures are two-layer bidirectional Long Short Term Memory Networks (LSTMs). The only difference is that we add the argument "-share_embeddings" to share the word embeddings between the encoder and decoder. This practice largely reduces model parameters for the monolingual task. On our computer (GPU: GTX 1080, Memory: 16G, CPU: i7-7700K), the training spends about 2 days.

During test, we use beam search of size 5 to generate summaries. We add the argument "-replace_unk" to replace the generated unknown words with the source word that holds the highest attention weight. Since the generated summaries are often shorter than the actual ones, we introduce an additional length penalty argument "-alpha 1" to encourage longer generation, like Wu et al. (2016).

3.4 Baselines

We compare our proposed model with the following state-of-the-art neural summarization systems:

ABS Rush et al. (2015a) used an attentive CNN encoder and a NNLM decoder to summarize the sentence.

ABS+ Rush et al. (2015a) further tuned the ABS model with additional hand-crafted features to balance between abstraction and extraction.

RAS-Elman As the extension of the ABS model, it used a convolutional attention-based encoder and a RNN decoder (Chopra et al., 2016).

Featseq2seq Nallapati et al. (2016) used a complete seq2seq RNN model and added the hand-crafted features such as POS tag and NER, to enhance the encoder representation.

Luong-NMT Chopra et al. (2016) implemented the neural machine translation model of Luong et al. (2015) for summarization. This model contained two-layer LSTMs with 500 hidden units in each layer.

OpenNMT We also implement the standard attentional seq2seq model with OpenNMT. All the settings are the same as our system. It is noted that OpenNMT officially examined the Gigaword dataset. We distinguish the official result[6] and our experimental result with suffixes "O" and "I" respectively.

FTSum Cao et al. (2017b) encoded the facts extracted from the source sentence to improve both the faithfulness and informativeness of generated summaries.

In addition, to evaluate the effectiveness of our joint learning framework, we develop a baseline named "PIPELINE". Its architecture is identical to Re^{3}Sum. However, it trains the Rerank module and Rewrite module in pipeline.

3.5 Informativeness Evaluation

Model	Perplexity
ABS†	27.1
RAS-Elman†	18.9
FTSum†	16.4
OpenNMT$_I$	13.2
PIPELINE	**12.5**
Re^{3}Sum	12.9

Table 2: Final perplexity on the development set. † indicates the value is cited from the corresponding paper. ABS+, Featseq2seq and Luong-NMT do not provide this value.

Let's first look at the final cost values (Eq. 9) on the development set. From Table 2, we can

[5] https://github.com/OpenNMT/OpenNMT-py

[6] http://opennmt.net/Models/

Model	RG-1	RG-2	RG-L
ABS†	29.55*	11.32*	26.42*
ABS+†	29.78*	11.89*	26.97*
Featseq2seq†	32.67*	15.59*	30.64*
RAS-Elman†	33.78*	15.97*	31.15*
Luong-NMT†	33.10*	14.45*	30.71*
FTSum†	**37.27**	17.65*	34.24
OpenNMT$^\dagger_O$	33.13*	16.09*	31.00*
OpenNMT$_I$	35.01*	16.55*	32.42*
PIPELINE	36.49	17.48*	33.90
Re^{3}Sum	37.04	**19.03**	**34.46**

Table 3: ROUGE F1 (%) performance. "RG" represents "ROUGE" for short. "*" indicates statistical significance of the corresponding model with respect to the baseline model on the 95% confidence interval in the official ROUGE script.

Type	RG-1	RG-2	RG-L
Random	2.81	0.00	2.72
First	24.44	9.63	22.05
Max	38.90	19.22	35.54
Optimal	52.91	31.92	48.63
Rerank	28.77	12.49	26.40

Table 4: ROUGE F1 (%) performance of different types of soft templates.

see that our model achieves much lower perplexity compared against the state-of-the-art systems. It is also noted that PIPELINE slightly outperforms Re^{3}Sum. One possible reason is that Re^{3}Sum additionally considers the cost derived from the Rerank module.

The ROUGE F1 scores of different methods are then reported in Table 3. As can be seen, our model significantly outperforms most other approaches. Note that, ABS+ and Featseq2seq have utilized a series of hand-crafted features, but our model is completely data-driven. Even though, our model surpasses Featseq2seq by 22% and ABS+ by 60% on ROUGE-2. When soft templates are ignored, our model is equivalent to the

Item	Template	OpenNMT	Re^{3}Sum
LEN_DIF	2.6±2.6	3.0±4.4	2.7±2.6
LESS_3	0	53	1
COPY(%)	31	80	74
NEW_NE	0.51	0.34	0.30

Table 5: Statistics of different types of summaries.

Type	RG-1	RG-2	RG-L
+Random	32.60	14.31	30.19
+First	36.01	17.06	33.21
+Max	41.50	21.97	38.80
+Optimal	46.21	26.71	43.19
+Rerank(Re^{3}Sum)	37.04	19.03	34.46

Table 6: ROUGE F1 (%) performance of Re^{3}Sum generated with different soft templates.

standard attentional seq2seq model OpenNMT$_I$. Therefore, it is safe to conclude that soft templates have great contribute to guide the generation of summaries.

We also examine the performance of directly regarding soft templates as output summaries. We introduce five types of different soft templates:

Random An existing summary randomly selected from the training corpus.

First The top-ranked candidate template given by the Retrieve module.

Max The template with the maximal actual ROUGE scores among the 30 candidate templates.

Optimal An existing summary in the training corpus which holds the maximal ROUGE scores.

Rerank The template with the maximal predicted ROUGE scores among the 30 candidate templates. It is the actual soft template we adopt.

As shown in Table 4, the performance of Random is terrible, indicating it is impossible to use one summary template to fit various actual summaries. Rerank largely outperforms First, which verifies the effectiveness of the Rerank module. However, according to Max and Rerank, we find the Rerank performance of Re^{3}Sum is far from perfect. Likewise, comparing Max and First, we observe that the improving capacity of the Retrieve module is high. Notice that Optimal greatly exceeds all the state-of-the-art approaches. This finding strongly supports our practice of using existing summaries to guide the seq2seq models.

3.6 Linguistic Quality Evaluation

We also measure the linguistic quality of generated summaries from various aspects, and the results are present in Table 5. As can be seen from the rows "LEN_DIF" and "LESS_3", the performance of Re^{3}Sum is almost the same as that of soft templates. The soft templates indeed well

Source	grid positions after the final qualifying session in the indonesian motorcycle grand prix at the sentul circuit , west java , saturday : UNK
Target	indonesian motorcycle grand prix grid positions
Template	grid positions for **british** grand prix
OpenNMT	circuit
Re^{3}Sum	grid positions for indonesian grand prix
Source	india 's children are getting increasingly overweight and unhealthy and the government is asking schools to ban junk food , officials said thursday .
Target	indian government asks schools to ban junk food
Template	**skorean** schools to ban **soda** junk food
OpenNMT	india 's children getting fatter
Re^{3}Sum	indian schools to ban junk food

Table 7: Examples of generated summaries. We use Bold font to indicate the crucial rewriting behavior from the templates to generated summaries.

guide the summary generation. Compared with Re^{3}Sum, the standard deviation of LEN_DF is 0.7 times larger in OpenNMT, indicating that OpenNMT works quite unstably. Moreover, OpenNMT generates 53 extreme short summaries, which seriously reduces readability. Meanwhile, the copy ratio of actual summaries is 36%. Therefore, the copy mechanism is severely overweighted in OpenNMT. Our model is encouraged to generate according to human-written soft templates, which relatively diminishes copying from the source sentences. Look at the last row "NEW_NE". A number of new named entities appear in the soft templates, which makes them quite unfaithful to source sentences. By contrast, this index in Re^{3}Sum is close to the OpenNMT's. It highlights the rewriting ability of our seq2seq framework.

3.7 Effect of Templates

In this section, we investigate how soft templates affect our model. At the beginning, we feed different types of soft templates (refer to Table 4) into the Rewriting module of Re^{3}Sum. As illustrated in Table 6, the more high-quality templates are provided, the higher ROUGE scores are achieved. It is interesting to see that,while the ROUGE-2 score of Random templates is zero, our model can still generate acceptable summaries with Random templates. It seems that Re^{3}Sum can automatically judge whether the soft templates are trustworthy and ignore the seriously irrelevant ones. We believe that the joint learning with the Rerank model plays a vital role here.

Next, we manually inspect the summaries generated by different methods. We find the outputs of Re^{3}Sum are usually longer and more fluent than the outputs of OpenNMT. Some illustrative examples are shown in Table 7. In Example 1, there is no predicate in the source sentence. Since OpenNMT prefers selecting source words around the predicate to form the summary, it fails on this sentence. By contract, Re^{3}Sum rewrites the template and produces an informative summary. In Example 2, OpenNMT deems the starting part of the sentences are more important, while our model, guided by the template, focuses on the second part to generate the summary.

In the end, we test the ability of our model to generate diverse summaries. In practice, a system that can provide various candidate summaries is probably more welcome. Specifically, two candidate templates with large text dissimilarity are manually fed into the Rewriting module. The corresponding generated summaries are shown in Table 8. For the sake of comparison, we also present the 2-best results of OpenNMT with beam search. As can be seen, with different templates given, our model is likely to generate dissimilar summaries. In contrast, the 2-best results of OpenNMT is almost the same, and often a shorter summary is only a piece of the other one. To sum up, our model demonstrates promising prospect in generation diversity.

4 Related Work

Abstractive sentence summarization aims to produce a shorter version of a given sentence while preserving its meaning (Chopra et al., 2016). This task is similar to text simplification (Saggion, 2017) and facilitates headline design and

Source	anny ainge said thursday he had two one-hour meetings with the new owners of the boston celtics but no deal has been completed for him to return to the franchise .
Target	ainge says no deal completed with celtics
Templates	major says no deal with spain on gibraltar
	roush racing completes deal with red sox owner
Re^{3}Sum	ainge **says no deal done** with **celtics**
	ainge **talks** with **new owners**
OpenNMT	ainge talks with **celtics** owners
	ainge talks with **new** owners
Source	european stock markets advanced strongly thursday on some bargain-hunting and gains by wall street and japanese shares ahead of an expected hike in us interest rates .
Target	european stocks bounce back UNK UNK with closing levels
Templates	european stocks bounce back strongly
	european shares sharply lower on us interest rate fears
Re^{3}Sum	european **stocks bounce back** strongly
	european **shares rise** strongly **on bargain-hunting**
OpenNMT	european stocks rise ahead of **expected** us rate hike **hike**
	european stocks rise ahead of us rate hike

Table 8: Examples of generation with diversity. We use Bold font to indicate the difference between two summaries

refine. Early studies on sentence summarization include template-based methods (Zhou and Hovy, 2004), syntactic tree pruning (Knight and Marcu, 2002; Clarke and Lapata, 2008) and statistical machine translation techniques (Banko et al., 2000). Recently, the application of the attentional seq2seq framework has attracted growing attention and achieved state-of-the-art performance on this task (Rush et al., 2015a; Chopra et al., 2016; Nallapati et al., 2016).

In addition to the direct application of the general seq2seq framework, researchers attempted to integrate various properties of summarization. For example, Nallapati et al. (2016) enriched the encoder with hand-crafted features such as named entities and POS tags. These features have played important roles in traditional feature based summarization systems. Gu et al. (2016) found that a large proportion of the words in the summary were copied from the source text. Therefore, they proposed CopyNet which considered the copying mechanism during generation. Recently, See et al. (2017) used the coverage mechanism to discourage repetition. Cao et al. (2017b) encoded facts extracted from the source sentence to enhance the summary faithfulness. There were also studies to modify the loss function to fit the evaluation metrics. For instance, Ayana et al. (2016) applied the Minimum Risk Training strategy to maximize the ROUGE scores of generated summaries. Paulus et al. (2017) used the reinforcement learning algorithm to optimize a mixed objective function of likelihood and ROUGE scores.

Guu et al. (2017) also proposed to encode human-written sentences to improvement the performance of neural text generation. However, they handled the task of Language Modeling and randomly picked an existing sentence in the training corpus. In comparison, we develop an IR system to find proper existing summaries as soft templates. Moreover, Guu et al. (2017) used a general seq2seq framework while we extend the seq2seq framework to conduct template reranking and template-aware summary generation simultaneously.

5 Conclusion and Future Work

This paper proposes to introduce soft templates as additional input to guide the seq2seq summarization. We use the popular IR platform Lucene to retrieve proper existing summaries as candidate soft templates. Then we extend the seq2seq framework to jointly conduct template reranking and template-aware summary generation. Experiments show that our model can generate informative, readable and stable summaries. In addition, our model demonstrates promising prospect in generation diversity.

We believe our work can be extended in various aspects. On the one hand, since the candidate templates are far inferior to the optimal ones, we intend to improve the Retrieve module, e.g., by indexing both the sentence and summary fields. On the other hand, we plan to test our system on the other tasks such as document-level summarization and short text conversation.

Acknowledgments

The work described in this paper was supported by Research Grants Council of Hong Kong (PolyU 152036/17E), National Natural Science Foundation of China (61672445 and 61572049) and The Hong Kong Polytechnic University (G-YBP6, 4-BCDV).

References

Shiqi Shen Ayana, Zhiyuan Liu, and Maosong Sun. 2016. Neural headline generation with minimum risk training. *arXiv preprint arXiv:1604.01904*.

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

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. Association for Computational Linguistics.

Ziqiang Cao, Chuwei Luo, Wenjie Li, and Sujian Li. 2017a. Joint copying and restricted generation for paraphrase. In *AAAI*, pages 3152–3158.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2017b. Faithful to the original: Fact aware neural abstractive summarization. *arXiv preprint arXiv:1711.04434*.

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. *arXiv preprint arXiv:1606.02858*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Sumit Chopra, Michael Auli, Alexander M Rush, and SEAS Harvard. 2016. Abstractive sentence summarization with attentive recurrent neural networks. *Proceedings of NAACL-HLT16*, pages 93–98.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2017. Generating sentences by editing prototypes. *arXiv preprint arXiv:1709.08878*.

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988*.

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. *arXiv preprint arXiv:1706.03872*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the ACL Workshop*, pages 74–81.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of ACL: System Demonstrations*, pages 55–60.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015a. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015b. A neural attention model for abstractive sentence summarization. In *Proceedings of EMNLP*, pages 379–389.

Horacio Saggion. 2017. Automatic text simplification. *Synthesis Lectures on Human Language Technologies*, 10(1):1–137.

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

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

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