# Abstractive Document Summarization via Bidirectional Decoder

Xin Wan[1,2], Chen Li[1], Ruijia Wang[1], Ding Xiao[1], and Chuan Shi[1,3]

[1] Beijing University of Posts and Telecommunications
[2] wanxin@bupt.edu.cn
[3] shichuan@bupt.edu.cn
http://www.shichuan.org

**Abstract.** Sequence-to-sequence architecture with attention mechanism is widely used in abstractive text summarization, and has achieved a series of remarkable results. However, this method may suffer from error accumulation. That is to say, at the testing stage, the input of decoder is the word generated at the previous time, so that decoder-side error will be continuously amplified. This paper proposes a **Sum**marization model using a **Bi**directional decoder (**BiSum**), in which the backward decoder provides a reference for the forward decoder. We use attention mechanism at both encoder and backward decoder sides to ensure that the summary generated by backward decoder can be understood. Also, pointer mechanism is added in both the backward decoder and the forward decoder to solve the out-of-vocabulary problem. We remove the word segmentation step in regular Chinese preprocessing, which greatly improves the quality of summary. Experimental results show that our work can produce higher-quality summary on Chinese datasets TTNews and English datasets CNN/Daily Mail.

**Keywords:** Abstractive Summarization · Bidirectional Decoder · Attention Mechanism · Sequence-to-Sequence Architecture.

## 1 Introduction

In the era of information explosion, summarization that can help people quickly extract knowledge is of great significance. Abstractive summarization is a technology of generating summary from source documents using deep learning methods, and hopes to keep the original meaning of the documents to the utmost extent.

Sequence-to-sequence architecture with attention mechanism (Seq2Seq-Attn) is a general solution for abstractive summarization in academic circles. In existing Seq2Seq-Attn summarization model, decoder-side input at the next time step is up to the referred summary while training. But at the testing stage, decoder-side input depends on its output at the previous time step. Hence, there will be a problem of error accumulation during testing. Once the decoder generates a wrong word, it will have a negative impact on the following predictions, which may result in the subsequent incorrect summary.

To solve this problem, we proposed an abstractive **Sum**marization model based on **Bi**directional decoder (**BiSum**), and Figure 1 is a diagram of it. A backward decoder is added in Seq2Seq-Attn model to generate the summary from right to left. The result of the backward decoder can provide a reference for the final summary through attention mechanism, thereby avoiding the error of the latter part of summary. Our model follows these steps: 1) generate the summary by the backward decoder from right to left in a similar manner to the Seq2Seq-Attn model; 2) apply attention mechanism to both encoder and backward decoder, so that the forward decoder can generate the summary from left to right. At the same time, the pointer mechanism [19] is also embedded in the forward decoder and backward decoder to address the out-of-vocabulary (OOV) problem. This problem is due to the limitation of the vocabulary size, that is, the vocabulary cannot cover all the words in the source documents and the referred summary. Both the Chinese summarization datasets TTNews and the English summarization datasets CNN/Daily Mail are conducted in our model. Since that the scale of Chinese summarization datasets is generally not large, the source documents are not segmented by words, but trained character by character. We find that this trick can significantly improve the quality of generated summary of BiSum and other models. Experimental results demonstrate that BiSum achieves excellent results on both datasets.

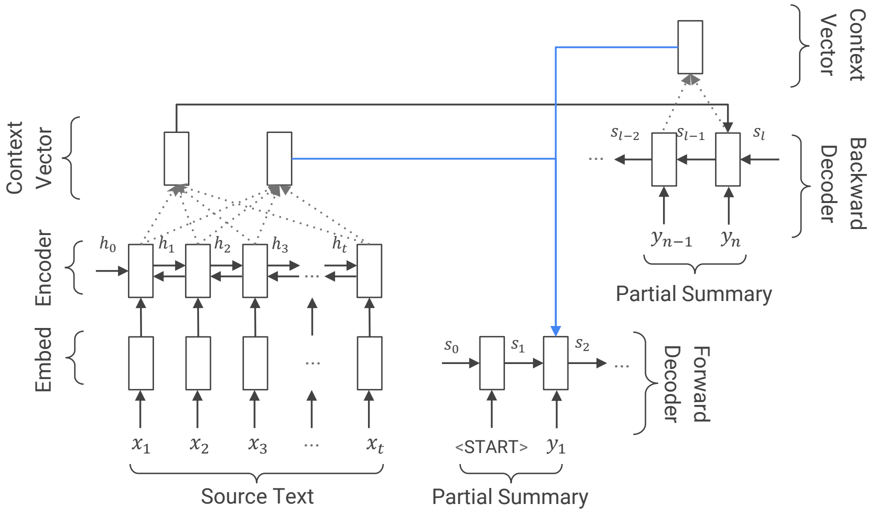


**Fig. 1.** BiSum is an Seq2Seq-Attn model with bidirectional decoder.

To sum up, the main contributions of our work include: 1) apply the bidirectional decoder into the abstractive summarization for the first time to avoid the accumulated errors; 2) integrate the pointer mechanism into the forward decoder and the backward decoder to solve the OOV problem; 3) propose not to segment the words in the source documents, which improves the quality of generated summary a lot; 4) verify the effectiveness and universality of our model on both Chinese and English datasets.

We organize this paper in the following sections. Section 2 summarizes the related work of text summarization and bidirectional decoder, and compares it with our work. Section 3 introduces the basic Seq2Seq-Attn model and defines BiSum on this basis. Experimental results, sample summary and the analysis of them are in the Section 4. We conclude our work in the last section, and propose possible directions for future improvement.

## 2 Related Work

### 2.1 Text Summarization

Text summarization problem studies how to automatically obtain a summary from the source texts. At present, two methods are mainly used for the problem: extractive and abstractive.

Extractive summarization is to select some sentences from the original text to summarize it. It guarantees the readability of the summary, but cannot maintain the logic between sentences, and there may be confusion in referential relation. The early extractive works [2,5] utilized the position of the sentence, cue word and other information to calculate the weight of each sentence, and the top-n sentences are chosen to compose a summary. Modern extractive methods mainly take advantage of deep learning methods [3,16,14] to achieve better results. For example, Nallapati et al. [14] exploited the GRU-based SummaRuNNer model to achieve state-of-the-art results in extractive summarization.

Unlike extractive methods, abstractive summarization relies on deep neural network and is closer to human thinking patterns. It requires the model to understand the meaning of the entire document and thus generate a summary based on it. With the continuous development of deep learning technologies in the field of natural language processing, especially the wide application of the Seq2Seq-Attn model in recent years, abstractive summarization has gradually become a new research hotspot. Since the input and output lengths often differ greatly, summarization problem is subdivided into the sentence-level summarization (generate headline based on the sentence) and the paragraph-level summarization (generate sentences based on the paragraph). Rush et al. [18] first proposed applying Seq2Seq-Attn model in abstractive summarization field and achieved the state-of-the-art results at that time on the DUC-2004 and Gigaword (two sentence-level summarization datasets).

In recent years, the optimization of this model is springing up. Zhou et al. [23] presented a sentence-level summarization method to copy the phrases in the source document from the model output in bundles. See et al. [19] introduced a paragraph-level summarization method to copy the input sequence into the output sequence by considering the attention distribution of the input sequence. Both of the above tasks are trying to solve the OOV problem, and they have achieved good results in the field. See et al. [19] proposed a coverage mechanism to reduce the weight of words that has been taken care of in the attention mechanism. Salesforce [17] put forward a self-attention mechanism to reduce the probability of generating duplicate words. Tan et al. [20] raised a hierarchical encoder and used a graph-based attention mechanism to decrease the probability of focusing on the same part of the source document. All three mentioned above attempted to prone the duplicate words in paragraph-level summarization. (There are a few duplicates in sentence-level summarization, because the generated headlines are short enough.)

In general, the current sentence-level summarization has been basically readable and has outperformed the extractive methods in some datasets. However, readability of existing paragraph-level summarization models is still not satisfying. There is still much room for improvement. Meanwhile, all of the above paragraph-level summarization works are based on the English datasets CNN/Daily Mail. Hu et al. [9] crawled news data from Sina Weibo to build a large-scale Chinese sentence-level summarization datasets, LCSTS. Chinese paragraph-level summarization work and datasets are rare nowadays [8], which does limit the development of it.

### 2.2 Bidirectional Decoder

In this work, we mainly use a bidirectional decoder to perform the summarization. Bidirectional decoder has been a pop topic in the field of machine translation and has been widely used in Neural Machine Translation (NMT) system. Watanabe et al. [21] first referred to the translation results generated in the forward and backward directions in the NMT. In recent years, the idea of bidirectional decoder has been continuously improved. Liu et al. [11] trained two bi-LSTM models at the same time, and then directly added the predict distributions of two models together to generate translations. Hoang et al. [7] proposed an approximate inference framework based on continuous optimization to decode the bidirectional model. Zhang et al. [22] used two independent attention distribution to let the forward decoder pay attention to the backward decoder.

To the best of our knowledge, our work is the first one to use the bidirectional decoder in the field of abstractive summarization. The bidirectional decoding mechanism in this paper is similar to the work of Zhang et al. [22]. The main difference between two works: 1) We add a pointer mechanism to both forward and backward decoders to avoid the OOV problem; 2) Different quality requirements for the summary generated at the backward decoder side. Duplicate output is common in abstractive summarization, but is rare in machine translation. Therefore, the output of backward decoder is stressed in BiSum.

## 3 Our Model

This section introduces the basic Seq2Seq-Attn model, then the definition of BiSum[4] and other tricks in it.

### 3.1 Seq2Seq-Attn

Seq2Seq-Attn is a regular solution to the abstractive summarization problem [12,18,15] and the method here is similar to [15]. Figure 2 describes this process.

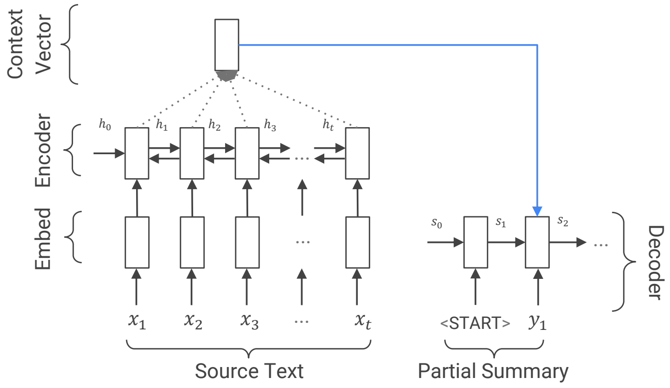


**Fig. 2.** The Seq2Seq-Attn model.

Seq2Seq architecture can be divided into the encoder and decoder. Encoder is usually a bidirectional, recurrent neural network (single-layer bi-LSTM in BiSum) that encodes the source text into a semantically hidden state vector $h_i$ ($i$ in $h_i$ represents the $i$th word in the sequence). The decoder is usually a recurrent neural network (single-layer LSTM in BiSum). It obtains the decoder's current state $s_t$ based on the decoder's hidden state vector $s_{t-1}$ at the previous time step. It should be noted that the decoder input $y$ will be the words of referred summary during training but the words of generated summary during testing. This is the reason why error accumulation occurs when testing mentioned in Section 1.

We use the same attention mechanism as Bahdanau et al. [1], so that each encoder's hidden state vector $h_i$ has an effect on every word generated by the decoder, but the attention distribution is different from each other. It can be derived as Equation 1.

[4] https://github.com/AnnieAldo/BiSum

$$
\begin{aligned}
e_i^t &= v^T \tanh(W_h h_i + W_s s_{t-1} + b_{attn}) \\
a_t &= softmax(e_i^t) \\
c &= \sum\nolimits_i a_i^t h_i
\end{aligned}
\tag{1}
$$

where $v, W_h, W_s, b_{attn}$ are the learnable variables, $a_t$ is the attention distribution of $h_i$, and $c$ is the context vector. $c$ is a weighted sum of $h_i$ and can be viewed as a collection of source text hidden state vectors of interest.

According to the context vector $c$ and the decoder current state $s_t$, we can calculate $P_{vocab}(w)$ (the probability distribution of our predicted word $w$ in the vocabulary), which also can be $P(w)$ (the probability distribution of the decoder output word).

$$
\begin{aligned}
P_{vocab} &= softmax(V'(V[s_t, c] + b) + b') \\
P(w) &= P_{vocab}(w)
\end{aligned}
\tag{2}
$$

In Equation 2, $V', V, b, b'$ are the learnable variables, and $P_{vocab}$ is the probability distribution of all vocabulary words, which is calculated through two linear layers. Assuming there is a training datasets $D = (x, y)$, where $x$ is the source document and $y$ is the reference summary, the objective function is defined as Equation 3.

$$
J(D) = \frac{1}{|D|} argmax \sum\nolimits_{(x,y)\in D} \log P(w) \tag{3}
$$

### 3.2 BiSum

The bidirectional decoder was implemented in the field of machine translation for the past two years [11,7,22] and can effectively solve the problem of error accumulation on the decoder-side. However, as far as we know, this article is the first work to apply it to abstractive summarization. The bidirectional decoder in this paper mainly refers to the work of Zhang et al. [22] in machine translation.

**Encoder.** BiSum has exactly the same encoder as Seq2Seq-Attn model. Attention distribution of the encoder is also consistent with the Equation 1.

**Backward Decoder.** Compared to the Seq2Seq-Attn model, BiSum has one more backward decoder. That is, before the summary is generated from left to right, a reverse summary is generated from right to left. Without considering the next steps, the encoder and the backward decoder can be understood as a Seq2Seq-Attn model, and only the direction of the generated summary is different from it. Therefore, the probability distribution of a vocabulary word generated by forward decoder is defined as Equation 4.

$$\hat{P}(w) = softmax(V'(V[s_t, c] + b) + b') \tag{4}$$

where hat symbol ^ represents operations related to the backward decoder, such as $\hat{P}(w)$ here.

**Forward Decoder.** Combining the results of the encoder and the backward decoder, we can further calculate the results of the forward decoder. At this point, we apply the attention mechanism to both the encoder and the backward decoder. For the encoder, the attention distribution $a_t$ can be calculated as Equation 5.

$$\begin{aligned} e_i^t &= v^T \tanh(W_h h_i + W_s s_{t-1} + b_{attn}) \\ a_t &= softmax(e_i^t) \\ c &= \sum\nolimits_i a_i^t h_i \end{aligned} \tag{5}$$

where $v, W_h, W_s, b_{attn}$ are the learnable variables, and $s_{t-1}$ is the hidden state vector of the forward decoder at the previous time step. $a_t$ is the attention distribution of $h_i$, and $c$ is the context vector.

For the backward decoder, the attention distribution $\hat{a}_t$ is:

$$\begin{aligned} \hat{e}_{t'}^t &= \hat{v}^T \tanh(W_{\hat{s}} \hat{s}_{t'} + W_s' s_{t-1} + \hat{b}_{attn}) \\ \hat{a}_t &= softmax(\hat{e}_{t'}^t) \\ \hat{c} &= \sum\nolimits_{t'} \hat{a}_{t'}^t \hat{s}_{t'} \end{aligned} \tag{6}$$

where $\hat{v}, W_{\hat{s}}, W_s', \hat{b}_{attn}$ are the learnable variables. $\hat{s}_{t'}$ is the hidden state vector of backward decoder at $t'$th time step, and $s_{t-1}$ is the hidden state vector of forward decoder at the previous time step. It should be noted here that when we utilize the attention mechanism to the backward decoder, the attention distribution is based on its hidden state vector $\hat{s}_{t'}$. Otherwise, if the mechanism is applied to the generated reverse summary, the error will affect the forward decoder more easily, which we do not wish to.

Founded on two attention mechanisms, we can derive the final probability distribution of the predicted word $w$ like Seq2Seq-Attn model.

$$P(w) = softmax(V'(V[s_t, c, \hat{c}] + b) + b') \tag{7}$$

where $V', V, b, b'$ are the learnable variables.

Finally, in order to guarantee the quality of the summary produced by the backward decoder, we reconstructed the objective function. The maximum likelihood of the backward decoder is added to the objective function and is balanced with the hyperparameter $\lambda$.

$$J(D) = \frac{1}{|D|} argmax \sum\nolimits_{(x,y)\in D} \left[\lambda \cdot \log P(w) + (1-\lambda) \cdot \log \hat{P}(\hat{w})\right] \tag{8}$$

In general, when we train a model incorporating a bidirectional decoder, we will first encode the input sequence, and then use a backward decoder to generate a reverse summary, and finally, employ a forward decoder to produce summary using the information provided by the former encoder and the backward decoder. Since the time complexity of beam search is too large, we use a smaller beam size in backward decoder to generate the reverse summary, which speeds up the model to some extent.

**Pointer Mechanism.** Pointer mechanism is used in the forward and backward decoder. It uses a soft attention distribution mechanism to make the output sequence map the input sequence, and can be very helpful to solve the OOV problem. The pointer mechanism used in this article is the same as that of See et al. [19], that is, controls the model's generating and pointing by $p_{gen}$. $p_{gen}$ $[0, 1]$ is defined as a probability of generating a word from the vocabulary. According to this definition, $(1 - p_{gen})$ represents the probability of copying a word from the input. The $p_{gen}$ at time step $t$ is calculated as shown in Equation 9.

$$p_{gen} = \sigma(w_c^T c + w_s^T s_t + w_y^T y_t + b_{ptr}) \tag{9}$$

where $w_c, w_s, w_y, b_{ptr}$ are learnable parameters, and $y_t$ is the input of decoder. $\sigma$ is the sigmoid function. Thus, we have got an extended vocabulary that contains all words in the vocabulary and all words that can be copied from the original texts. Considering $p_{gen}$ as a soft switch, the probability distribution of this extended vocabulary is shown in Equation 10.

$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum\nolimits_{i:w_i=w} a_i^t \tag{10}$$

**No Word Segmentation.** Intuitively speaking, when doing Chinese processing, we segmented both the source text and the referred summary word by word. However, in the following experiments, we find that if we remove the word segmentation part and train the text sequence character by character, the results of the model are significantly better regardless of the model we used. This is due to the fact that when we segment the corpora, we naturally get a very large vocabulary, which is detrimental to smaller datasets and can lead to sparse data. But when we generate the summary character by character, the vocabulary size is reduced, and the relevance between characters is strengthened. Considering that there are no large-scale Chinese paragraph-level summarization datasets in academic circles and industrial circles, it is a good way to improve the summary quality.

# 4 Experiments

## 4.1 Datasets

We evaluated our model on two paragraph-level summarization datasets. They are NLPCC 2017 TTNews in Chinese and CNN/Daily Mail in English.

**TTNews.** It contains a large number of news articles from Toutiao.com and corresponding manual summaries for news feeds. In addition, the training set also contains another set of news articles without summaries (perhaps provided for semi-supervised methods, not used in this paper). As far as we know, TTNews is the largest single-document paragraph-level summarization corpus in Chinese, with 50,000 news articles containing summaries and 52,000 news articles without summaries.

**CNN/Daily Mail.** This corpus was recently widely used in the paragraph-level summrization field. [6,15,19,17] It contains news articles in CNN/Daily Mail and manual summaries for them. We used the data preprocessing scripts provided by Nallapati et al. [15] to obtain 312,084 article and summary pairs. We did not use Named Entity Recognition [13] (NER) technology to replace proper nouns, because of the pointer mechanism in the forward and backward decoder.

The specific statistics of the two corpus are shown in Table 1.

**Table 1.** Information of the two datasets TTNews and CNN/Daily Mail.

| Datasets | | The Number of Documents | The Average Length of Documents | The Average Length of Summaries |
|---|---|---:|---:|---:|
| **TTNews** | Training Set (with summary) | 50,000 | 1,036 | 45 |
| | Training Set (without summary) | 50,000 | 1,526 | / |
| | Test Set | 2,000 | 1,037 | 45 |
| **CNN/ Daily Mail** | Training Set | 287,226 | 781 | 56 |
| | Test Set | 13,368 | 781 | / |
| | Validation Set | 11,490 | 781 | 56 |

For the CNN/Daily Mail datasets, the file format is .story, and for the TTNews datasets is .txt. We format them and further convert them into binary files. Also, for the convenience of training, every 1000 samples are integrated as a chunk.

## 4.2 Baseline

**Seq2Seq-Attn.** [15] The Seq2Seq-Attn model mentioned in Section 3.

**Seq2Seq-Attn + Pointer.** [19] The Seq2Seq-Attn model with the pointer mechanism mentioned in Section 3. It's worth mentioning that we generated the summary from left to right and from right to left, respectively, to ensure the effectiveness of the bidirectional decoder.

### 4.3 Setup

For the TTNews datasets, due to the absence of the referred summary test set, we first separate 5000 samples from the training set into the test set and 5000 samples into the validation set. Meanwhile, because the word segmentation does not take place, the vocabulary size is greatly reduced. For the data with word segmentation, vocabulary size is set to be 60k. And for the data without word segmentation, vocabulary size is 6k.

For the CNN/Daily Mail datasets, Seq2Seq-Attn model uses a vocabulary size of 150k (because it encounters the OOV problem), and other models have 50k words in the vocabulary.

In all the experiments in this paper, the dimension of the hidden state vector is 256, and the dimension of the word embedded vector is 128. Instead of using the pre-trained word vectors, we learn them with other parameters during training. Adagrad [4] is used for training with a learning rate of 0.15 and an initial accumulator value of 0.1. The maximum gradient norm is configured to be 5. To prevent over-fitting, we achieved early stop by observing the losses on the validation set.

Since limiting the length of documents and summaries speeds up the training and testing and improves the performance of the model [19], we limit document length to 400 words and the summary length to 100 words in training, 120 words in testing. The model is trained on a single GTX 1080 Ti GPU, and the batch size is 16. Beam search is used to generate the summary with width 2 in backward decoder and width 4 in forward decoder.

For the Seq2Seq-Attn model, we trained about 236K iterations on the CNN/Daily Mail dataset which spent 1 day 17 hours. Segmented TTNews is trained about 125k iterations which took 14 hours 40 minutes, and 85k iterations took 8 hours 40 minutes for non-segmented datasets. For BiSum, English, segmented and non-segmented Chinese datasets trained about 103K, 160k, and 90k iterations and took 21 hours 7 minutes, 22 hours 40 minutes, and 18 hours 58 minutes, respectively.

### 4.4 Effectiveness Experiments

The experimental results are shown in Table 2. We use the standard ROUGE metric[10] to evaluate BiSum, and give the F1 scores for ROUGE-1, ROUGE-2, and ROUGE-L (character overlap, 2-grams overlap, and longest common subsequence overlap for the generated summary and referred summary, respectively).

Our work outperforms the baseline in most cases, which shows that bidirectional decoders do play a positive role in the summarization. It can be also found

**Table 2.** Experimental results of the baseline and BiSum.

| Datasets | Model / Evaluation | ROUGE | | |
|---|---|---|---|---|
| | | 1 | 2 | L |
| **CNN/ Daily Mail** | Seq2Seq-Attn | 30.49 | 11.17 | 28.08 |
| | Seq2Seq-Attn + Pointer (left to right) | 36.44 | 15.66 | 33.42 |
| | Seq2Seq-Attn + Pointer (right to left) | 35.46 | 15.30 | 33.28 |
| | BiSum | **37.01** | **15.95** | **33.66** |
| **TTNews (segmented)** | Seq2Seq-Attn | 32.71 | 15.42 | 29.26 |
| | Seq2Seq-Attn + Pointer (left to right) | 35.03 | **18.03** | 30.38 |
| | Seq2Seq-Attn + Pointer (right to left) | 34.59 | 17.51 | 30.14 |
| | BiSum | **35.18** | 17.70 | **30.61** |
| **TTNews (non-segmented)** | Seq2Seq-Attn | 36.43 | 21.17 | 30.41 |
| | Seq2Seq-Attn + Pointer (left to right) | 39.85 | 23.69 | 32.52 |
| | Seq2Seq-Attn + Pointer (right to left) | 39.14 | 23.62 | 32.38 |
| | BiSum | **40.89** | **25.04** | **34.97** |

that the performance of the two Seq2Seq-Attn + Pointer models are always similar, but the ROUGE scores of left-to-right model are often higher. We speculate that the model is more likely to focus on the first sentences of the source document when summary is generated from left to right, so that characteristic of the news datasets (the first few sentences are likely to be a good summary) leads to this phenomenon. In all experiments, the performance of non-segmented models is better than segmented models.

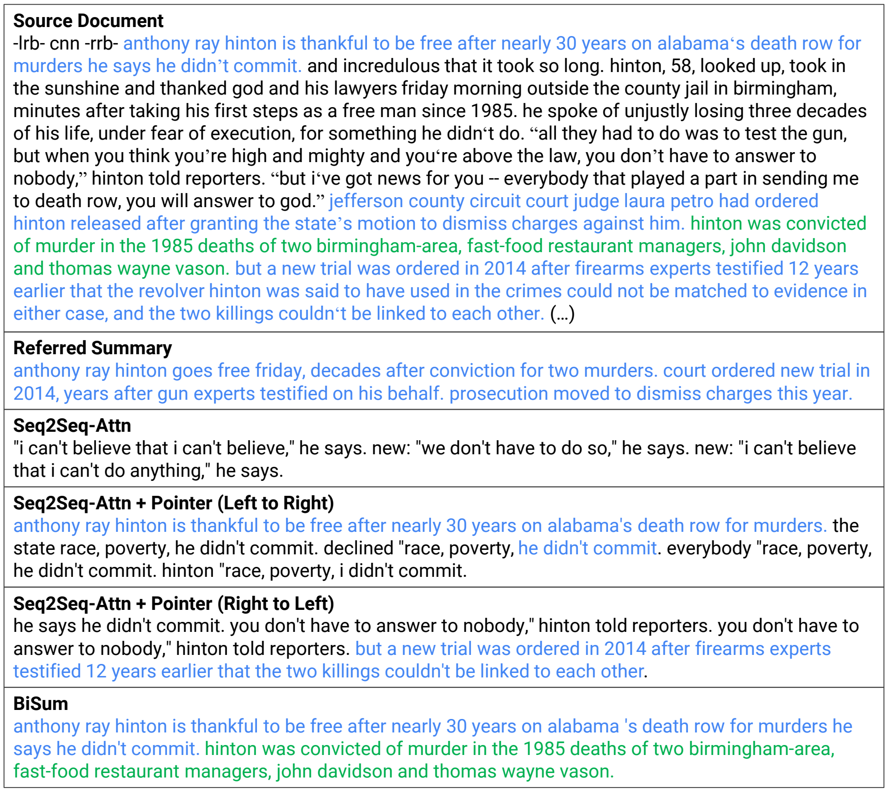
### 4.5 Case Study

We showed some sample summaries in the Figure 3 and Figure 4. Obviously, reverse summaries generated by the right-to-left pointer model tend to have better performance at the tail of the summaries, which are in line with our expectations and form a complementary with the left-to-right summaries. However, the summaries are sometimes likely to repeat themselves, and this problem also appears in BiSum. We will try to address this problem in further work.

In CNN/Daily Mail datasets, we can find that although the summary generated by BiSum is not exactly the same as the referred summary, the sentences they attended to are adjacent. Since summarization is a subjective work, it can be considered that BiSum produces a meaningful summary here. In TTNews datasets, the results of the segmented model are significantly more prone to bias. That is because the vocabulary is too large but the datasets are small: when the two documents have a same word, the decoder may generate the words in another document.

## 5 Conclusion and Further Work

Based on the traditional Seq2Seq-Attn model, this work introduces a bidirectional decoder for the problem of error accumulation when generating sum-



**Source Document**
-lrb- cnn -rrb- anthony ray hinton is thankful to be free after nearly 30 years on alabama‘s death row for murders he says he didn’t commit. and incredulous that it took so long. hinton, 58, looked up, took in the sunshine and thanked god and his lawyers friday morning outside the county jail in birmingham, minutes after taking his first steps as a free man since 1985. he spoke of unjustly losing three decades of his life, under fear of execution, for something he didn‘t do. “all they had to do was to test the gun, but when you think you’re high and mighty and you‘re above the law, you don’t have to answer to nobody,” hinton told reporters. “but i‘ve got news for you -- everybody that played a part in sending me to death row, you will answer to god.” jefferson county circuit court judge laura petro had ordered hinton released after granting the state’s motion to dismiss charges against him. hinton was convicted of murder in the 1985 deaths of two birmingham-area, fast-food restaurant managers, john davidson and thomas wayne vason. but a new trial was ordered in 2014 after firearms experts testified 12 years earlier that the revolver hinton was said to have used in the crimes could not be matched to evidence in either case, and the two killings couldn‘t be linked to each other. (...)

**Referred Summary**
anthony ray hinton goes free friday, decades after conviction for two murders. court ordered new trial in 2014, years after gun experts testified on his behalf. prosecution moved to dismiss charges this year.

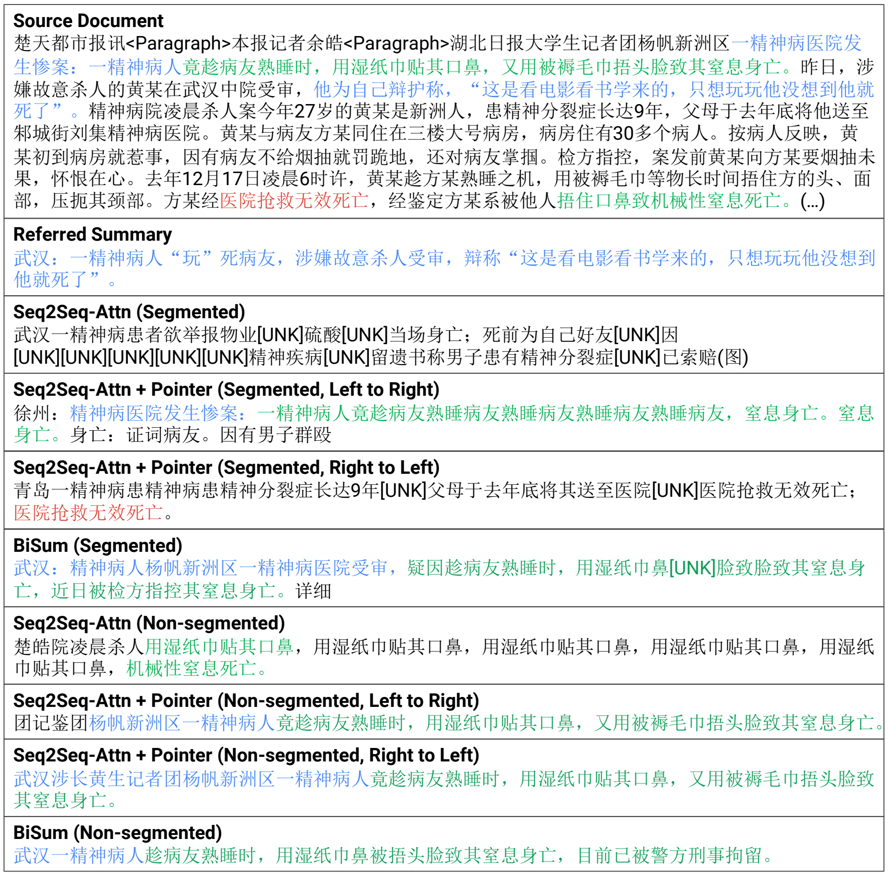
**Seq2Seq-Attn**
"i can't believe that i can't believe," he says. new: "we don't have to do so," he says. new: "i can't believe that i can't do anything," he says.

**Seq2Seq-Attn + Pointer (Left to Right)**
anthony ray hinton is thankful to be free after nearly 30 years on alabama's death row for murders. the state race, poverty, he didn't commit. declined "race, poverty, he didn't commit. everybody "race, poverty, he didn't commit. hinton "race, poverty, i didn't commit.

**Seq2Seq-Attn + Pointer (Right to Left)**
he says he didn't commit. you don't have to answer to nobody," hinton told reporters. you don't have to answer to nobody," hinton told reporters. but a new trial was ordered in 2014 after firearms experts testified 12 years earlier that the two killings couldn't be linked to each other.

**BiSum**
anthony ray hinton is thankful to be free after nearly 30 years on alabama 's death row for murders he says he didn't commit. hinton was convicted of murder in the 1985 deaths of two birmingham-area, fast-food restaurant managers, john davidson and thomas wayne vason.

**Fig. 3.** Sample English summaries of the models. Texts in blue show the correct referred summary. Texts in green indicate that although they are inconsistent with the referred summary, they can be subjectively interpreted as a correct summary.



**Source Document**
楚天都市报讯<Paragraph>本报记者余皓<Paragraph>湖北日报大学生记者团杨帆新洲区一精神病医院发生惨案：一精神病人竟趁病友熟睡时，用湿纸巾贴其口鼻，又用被褥毛巾捂头脸致其窒息身亡。昨日，涉嫌故意杀人的黄某在武汉中院受审，他为自己辩护称，“这是看电影看书学来的，只想玩玩他没想到他就死了”。精神病院凌晨杀人案今年27岁的黄某是新洲人，患精神分裂症长达9年，父母于去年底将他送至邾城街刘集精神病医院。黄某与病友方某同住在三楼大号病房，病房住有30多个病人。按病人反映，黄某初到病房就惹事，因有病友不给烟抽就罚跪地，还对病友掌掴。检方指控，案发前黄某向方某要烟抽未果，怀恨在心。去年12月17日凌晨6时许，黄某趁方某熟睡之机，用被褥毛巾等物长时间捂住方的头、面部，压扼其颈部。方某经医院抢救无效死亡，经鉴定方某系被他人捂住口鼻致机械性窒息死亡。(...)

**Referred Summary**
武汉：一精神病人“玩”死病友，涉嫌故意杀人受审，辩称“这是看电影看书学来的，只想玩玩他没想到他就死了”。

**Seq2Seq-Attn (Segmented)**
武汉一精神病患者欲举报物业[UNK]硫酸[UNK]当场身亡；死前为自己好友[UNK]因[UNK][UNK][UNK][UNK][UNK]精神疾病[UNK]留遗书称男子患有精神分裂症[UNK]已索赔(图)

**Seq2Seq-Attn + Pointer (Segmented, Left to Right)**
徐州：精神病医院发生惨案：一精神病人竟趁病友熟睡病友熟睡病友熟睡病友熟睡病友，窒息身亡。窒息身亡。身亡：证词病友。因有男子群殴

**Seq2Seq-Attn + Pointer (Segmented, Right to Left)**
青岛一精神病患精神病患精神分裂症长达9年[UNK]父母于去年底将其送至医院[UNK]医院抢救无效死亡；医院抢救无效死亡。

**BiSum (Segmented)**
武汉：精神病人杨帆新洲区一精神病医院受审，疑因趁病友熟睡时，用湿纸巾鼻[UNK]脸致脸致其窒息身亡，近日被检方指控其窒息身亡。详细

**Seq2Seq-Attn (Non-segmented)**
楚皓院凌晨杀人用湿纸巾贴其口鼻，用湿纸巾贴其口鼻，用湿纸巾贴其口鼻，用湿纸巾贴其口鼻，用湿纸巾贴其口鼻，机械性窒息死亡。

**Seq2Seq-Attn + Pointer (Non-segmented, Left to Right)**
团记鉴团杨帆新洲区一精神病人竟趁病友熟睡时，用湿纸巾贴其口鼻，又用被褥毛巾捂头脸致其窒息身亡。

**Seq2Seq-Attn + Pointer (Non-segmented, Right to Left)**
武汉涉长黄生记者团杨帆新洲区一精神病人竟趁病友熟睡时，用湿纸巾贴其口鼻，又用被褥毛巾捂头脸致其窒息身亡。

**BiSum (Non-segmented)**
武汉一精神病人趁病友熟睡时，用湿纸巾鼻被捂头脸致其窒息身亡，目前已被警方刑事拘留。

**Fig. 4.** Sample Chinese summaries of the models. Texts in red point to the source of error.

maries. We added the pointer mechanism and remove the word segmentation for the datasets (only for TTNews), which effectively reducing the vocabulary size and improving the model's performance. The experimental results indicate that our model can obtain remarkable results in Chinese and English summarization tasks. Also, we find that there are still repeated words appearing in the generated abstract, and we hope to deal with this problem in the follow-up work.

**Acknowledgement.** This work is supported in part by the National Natural Science Foundation of China (No. 61772082, 61806020, 61375058), and the Beijing Municipal Natural Science Foundation (4182043).

## References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Baxendale, P.B.: Machine-made index for technical literature—an experiment. IBM Journal of Research and Development **2**(4), 354–361 (1958)
3. Cao, Z., Wei, F., Dong, L., Li, S., Zhou, M.: Ranking with Recursive Neural Networks and Its Application to Multi-Document Summarization. In: AAAI. pp. 2153–2159 (2015)
4. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**(Jul), 2121–2159 (2011)
5. Edmundson, H.P.: New methods in automatic extracting. Journal of the ACM (JACM) **16**(2), 264–285 (1969)
6. Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P.: Teaching machines to read and comprehend. In: NIPS. pp. 1693–1701 (2015)
7. Hoang, C.D.V., Haffari, G., Cohn, T.: Towards Decoding as Continuous Optimisation in Neural Machine Translation. In: EMNLP. pp. 146–156 (2017)
8. Hou, L., Hu, P., Bei, C.: Abstractive Document Summarization via Neural Model with Joint Attention. In: National CCF Conference on Natural Language Processing and Chinese Computing. pp. 329–338. Springer (2017)
9. Hu, B., Chen, Q., Zhu, F.: Lcsts: A large scale chinese short text summarization dataset. arXiv preprint arXiv:1506.05865 (2015)
10. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out: ACL workshop (2004)
11. Liu, L., Utiyama, M., Finch, A., Sumita, E.: Agreement on target-bidirectional neural machine translation. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 411–416 (2016)
12. Lopyrev, K.: Generating news headlines with recurrent neural networks. arXiv preprint arXiv:1512.01712 (2015)
13. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30**(1), 3–26 (2007)
14. Nallapati, R., Zhai, F., Zhou, B.: SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In: AAAI. pp. 3075–3081 (2017)

15. Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., Others: Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023 (2016)
16. Narayan, S., Papasarantopoulos, N., Cohen, S.B., Lapata, M.: Neural extractive summarization with side information. arXiv preprint arXiv:1704.04530 (2017)
17. Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304 (2017)
18. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015)
19. See, A., Liu, P.J., Manning, C.D.: Get To The Point: Summarization with Pointer-Generator Networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. vol. 1, pp. 1073–1083 (2017)
20. Tan, J., Wan, X., Xiao, J.: Abstractive document summarization with a graph-based attentional neural model. In: ACL. vol. 1, pp. 1171–1181 (2017)
21. Watanabe, T., Sumita, E.: Bidirectional decoding for statistical machine translation. In: Proceedings of the 19th international conference on Computational linguistics-Volume 1. pp. 1–7. Association for Computational Linguistics (2002)
22. Zhang, X., Su, J., Qin, Y., Liu, Y., Ji, R., Wang, H.: Asynchronous Bidirectional Decoding for Neural Machine Translation. arXiv preprint arXiv:1801.05122 (2018)
23. Zhou, Q., Yang, N., Wei, F., Zhou, M.: Sequential Copying Networks. In: AAAI (2018)