# An Effective Joint Framework for Document Summarization

Min Gui*, Zhengkun Zhang, Zhenglu Yang*
CCCE, Nankai University, China
{nk_guimin,zzk446501998}@mail.nankai.edu.cn,
yangzl@nankai.edu.cn

Yanhui Gu
School of CS and Technology,
Nanjing Normal University, China
gu@njnu.edu.cn

Guandong Xu
Advanced Analytics Institute,
University of Technology Sydney,
Australia
Guandong.xu@uts.edu.au

## ABSTRACT

Document summarization is an important research issue and has attracted much attention from the academe. The approaches for document summarization can be classified as *extractive* and *abstractive*. In this work, we introduce an effective joint framework that integrates extractive and abstractive summarization models, which is much closer to the way human write summaries (first underlining important information). Preliminary experiments on real benchmark dataset demonstrate that our model is competitive with the state-of-the-art methods.

## KEYWORDS

Abstractive Summarization; Extractive summarization; Sequence-to-Sequence Framework

## 1 INTRODUCTION

Document summarization, a task to compress a document into a condensed but informative summary, has been extensively investigated to alleviate information overload. Studies on document summarization task have used either extractive or abstractive methods. A vast amount of previous work in summarization task has been extractive [2] due to the immaturity of text generation technologies and the simplicity of extractive methods, which generally identify key sentences or important phrases of an input document and reproduce them as a summary. However, extractive methods face incoherence problem and is different from the way human writes. Abstractive summarization attempts to produce a condensed representation, aspects of which may not appear as parts of the original input text. With the emergence of deep learning techniques as a viable alternative for Natural Language Process (NLP) tasks, researchers have begun applying modern neural networks to abstractive summarization [4], which is much closer to the way humans write summaries. Although abstractive methods have achieved remarkable success, they still remain challenging. Encoding and decoding a long sequence of multiple sentences fail to achieve satisfactory performance. The neural Seq2Seq framework for summarization tends to generate trivial and generic summaries with limited grammaticality and readability and is trained to predict the next word with previous ground-truth words as the input by using the maximum likelihood estimation (MLE) objective function [5].

To address these challenges, we introduce a joint framework which utilizes the advantages of extractive and abstractive summarization to improve the performance. According to the experiments and discussions in [4], using only the first 400 tokens of an article can yield significantly higher ROUGE scores than those obtained by using the first 800 tokens. This result may be explained by the nature of articles that most of articles especially news ones tend to be structured with the most important information at the start or in some sentences. Inspired by this observation, we use extractive summarization methods to select $k$ most important sentences before implementing abstractive methods. This process is similar to how human always underlie salient sentences first before they summarize an article.

## 2 OUR MODEL

In this section, we introduce our proposed model. The widely-used sequence-to-sequence framework is adopted to encode an article with multiple sentences and decode it as a short condensed summary. Our baseline model is similar to that of See et al. [4], and is illustrated in Figure. 1. The main distinction of our work is that we introduce an extractive layer which highlights salient sentences to improve the efficiency of decoder when attending to different parts of input document. In the following sections, we will first introduce the extractive layer, and then describe the combination of extractive and abstractive methods.

### 2.1 Extractive Layer

A document $d$ is a sequence of sentences $d = S_1, S_2, \cdots, S_n$, and a sentence $S_i$ is a sequence of words $S_i = w_{i1}, w_{i2}, \cdots, w_{ik}$, where $w_{ik}$ is the $k$-th word from the $i$-th sentence. Words are fed one-by-one into a single-layer bidirectional LSTM encoder. The encoder produces a sequence of encoder hidden state $h_i^e$ and the decoder has its state at time step $t$, denoted as $h_t^d$. Mimicking the way humans write summaries, we implement an extractive layer by using extractive summarization methods to highlight salient and informative sentences, whose content is more likely to be attended to by the decoder. Although many extractive methods can be utilized to select salient sentences, we simplify this process and follow Lead-3 model, which achieves outstanding performance and is surpassed by the best extractive system by only a small margin [2, 4]. It selects only the first $k$ sentences of each article as the basis of abstractive summary generation.

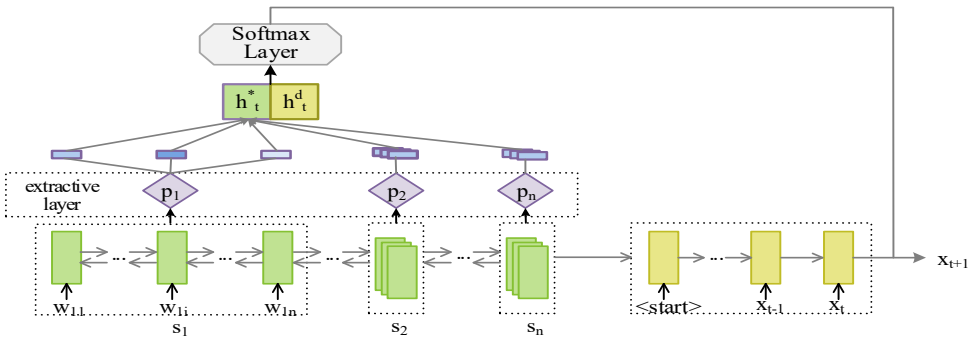


**Figure 1: Overview of the proposed model**

## 2.2 Neural Attentional Model

The attention mechanism [1] is usually introduced to alleviate the burden of remembering the whole input sequence and different parts of input document at different time step. In our work, the attention distribution $a^t$ are calculated as as in See et al. [4]:

$$a^t = softmax(v^T tanh(W_h h_i^e + W_s h_t^d + b_{attn})) \quad (1)$$

where$v$, $W_h$, $W_s$ and $b_{attn}$ are learnable parameters. The context vector $h_t^*$, which is a weighted sum of the encoder hidden states, is set to be different at different time step, namely, $h_t^* = \sum_i a_i^t h_i^e$.

Then the decoder hidden state is concatenated with the context vector and fed through two linear layers to produce the vocabulary distribution:

$$P_{vocab} = softmax(V^*(V([h_t^d, h_t^*]) + b) + b^*) \quad (2)$$

where $V^*$, $V$, $b$, and $b^*$ are learnable parameters. Similar to See et al. [4] and Nallapati et al. [3], we introduce a pointer mechanism that allows both copying words via pointing and generating words from the vocabulary, to approach the out-of-vocabulary words. We define $u$ as a binary value, $u = 1$ means the pointer mechanism working, 0 otherwise. The final distribution is:

$$P(y_t) = p(u_t = 1)p(y_t|u_t = 1) + p(u_t = 0)p_{vocab}(y_t) \quad (3)$$

where $p(y_t = x_i|u_t = 1)$ is equal to the attention weight of input token $x_i$, which is copied from source document.

, then output of time step $t$ is denoted as $y_t$,

The loss function $L$ of the model is the negative log likelihood of generating summaries. We define the ground-truth output sequence as $y^* = y_1^*, y_2^*, \cdots, y_n^*$ for a given input $d$, during training, the loss is calculated as $L = -\sum_{t=1}^{n} logp(y_t^*|y_1^*, \cdots, y_{t-1}^*, x)$.

## 3 EXPERIMENT

We conduct experiments on *CNN/Daily Mail* corpus, which is widely used in abstractive document summarization and comprises news stories with multi-sentence human generated summaries. The corpus contains 287,226 training pairs, 13,368 validation pairs, and 11,490 test pairs. The value of $k$, which denotes the number of sentences being selected as the basis of abstractive summarization process, is set to 5.

We compare our model with two state-of-the-art approaches, i.e., graph-based attention model (GBA) [6] and pointer-generator network (PGN, without coverage mechanism) [4]. We have conducted preliminary experiments on the proposed model with the extractive layer due to limited experimental condition. As illustrated in Table 1, our model can improve about 2.45 point in terms of ROUGE-2. Furthermore, we trained our model for 30000 iterations, which is much fewer than the 600,000 iterations required by PGN [4], while we have achieved better performance than PGN.

**Table 1: Performance Comparison on ROUGE-N**

| Models | GBA | PGN | Ours |
|---|---|---|---|
| ROUGE-1 | 38.10 | 36.09 | 36.38 |
| ROUGE-2 | 13.90 | 15.10 | **16.35** |
| ROUGE-L | 34.00 | 32.78 | 33.35 |
| Human | 3.14 | 3.89 | **3.91** |

We also perform human evaluation to evaluate output summaries. We randomly select 30 articles from the dataset, three evaluators are asked to score summaries generated by typical models from 1 point to 5 point, where 1 indicated the lowest readability and 5 indicates the highest level. From Table 1, we can see that the proposed method can improve the readability of summaries.

## 4 CONCLUSION

In this paper, we proposed a joint framework of extractive and abstractive summarization methods. Experimental results demonstrated that our model improves the performance on baseline dataset and can generate more readable and natural summaries.

## ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant No.U1636116, 11431006, 41571382, the Research Fund for International Young Scientists under Grant No. 61650110510 and 61750110530, and Jiangsu Higher Education Institutions of China under Grant No. 15KJA420001.

## REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR* (2014).
[2] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In *AAAI*.
[3] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *SIGNLL*.
[4] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *ACL*.
[5] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NIPS*.
[6] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive Document Summarization with a Graph-Based Attentional Neural Model. In *ACL*.