# An Extraction-Abstraction Hybrid Approach for Long Document Summarization

Si Huang, Rui Wang, Qing Xie, Lin Li, Yongjian Liu
School of Computer Science and Technology
Wuhan University of Technology, Wuhan, China
Email: {huangsi,rui,felixxq,cathylilin,liuyj}@whut.edu.cn

***Abstract*—In this paper, we propose a hybrid model of extractive and abstractive methods to tackle the long document automatic summarization task. The model first trains an extractor to extract salient sentences from the original text. Next, these salient sentences are put together to get a condensed version of the original text. Then we use the abstractive model to rewrite the extracted sentences to get the final summary. In order to avoid the exposure bias, reinforcement training is used to optimize the proposed model. Experiments in NLPCC2017 Shared Task 3 show that our models achieve competitive performance. Additionally, the ROUGE score of our model exceeds the score of the state-of-the-art model in the original NLPCC2017 Shared Task 3, where a sentence summary is generated from each Chinese news article.**

***Index Terms*—Text summarization, hybrid model, reinforcement learning**

## I. Introduction

The Internet contains massive files and is still growing exponentially. We are also faced with the problem of information overload while enjoying the convenience of accessing a large amount of shared information. How to help people quickly understand the massive information from various sources, and reduce information overload has become a research hotspot in the text analysis field. Automatic summarization technique is a good solution to address information overload. It takes a single or a set of documents as input and produces a concise summary that conveys the most important information.

The summarization technique can be formally divided into extractive summarization and abstractive summarization. Extractive summarization extracts salient sentences from the original text, which has the advantage of generating fluent sentences and retaining the meaning of the original document, but there are inevitably problems of inconsistency between sentences. Graph-based ranking model [1], [2] and neural network based extracting model [3], [4] are typical models for extractive summarization.

In contrast, the abstractive summarization uses new sentences and expressions to describe the content of the original text, which is closer to the way humans write summaries, but much more difficult than extractive summarization. Recently, encoder-decoder model with attention mechanism has been applied to abstractive summarization, such as Summarization with Pointer-Generator Networks [5]. The development of the abstractive summarization is mainly based on the use of the encoder-decoder model with attention mechanism. The original text is used as the input of the model, and a beam search is performed to obtain a summary output. The encoder-decoder model can get a higher ROUGE score when processing a short input sequence, but there are problems such as lack of long-term dependencies, when processing long input sequences.

In order to solve these problems, we propose a hybrid model that combines abstractive and extractive methods named Abstractive Summarization after Extraction (ASAE), which improves the model proposed by Chen et al. [6]. Our model first trains an extractor to extract salient sentences from the original text. Next, we put these salient sentences together, and get a condensed version of the original. Then we use the abstractive model to rewrite the extracted sentences to get the final summary, thus avoiding the problems that the abstractive summarization model has when dealing with long text.

However, when the baseline model was trained, the number of sentences extracted and the number of abstracts provided were the same. That is, for each summary sentence, the extractor selects the original sentence with the most relevance and lets the abstractor rewrite it. A summary based on one original sentence may be a bit biased and not comprehensive. In order to cope with this problem, we choose to extract multiple highly relevant sentences from the original, and the summary is abstracted from multiple correlated sentences. Out-of-vocabulary (OOV) words and duplicate words are serious challenges when dealing with long texts. We replace the basic seq2seq model with the pointer-generator network, and employ coverage mechanism to handle these problems.

Our contributions of this work are mainly summarized as follows:

1) In order to solve the problem of capturing long-term dependencies, we design a hybrid model combining the extractive and abstractive methods, which improves the baseline model by generating the summary from multiple extracted sentences from long documents.
2) We utilize the pointer-generator and coverage mechanisms to handle the OOV words and repetition problems.
3) We applied our model in NLPCC2017 Shared Task 3, and the achieved score exceeded the first place in the original NLPCC2017 Shared Task 3.

This work is partially supported by National Natural Science Foundation of China (Grant No.61602353) and the Fundamental Research Funds for the Central Universities (WUT:2019III054GX).

978-1-7281-4762-8/19/$31.00 

## II. Related Works

### A. Extractive Summarization

Most of the summary models studied in the past are essentially extractive models, usually by identifying the most salient phrases in the input document and reorganizing them into a new sequence of summarization. Traditional methods for extractive summarization can be broadly classified into greedy methods [7], graph-based methods [2], and constrained optimization-based methods [8]. In recent years, neural network-based methods have become popular for extractive summarizations. Kageback et al. [9] used a recursive auto encoder to summarize the document and achieved the best results on the Opinosis data set. Cheng and Lapata [10] proposed an attention-based encoder-decoder model for extracting single document summarizations and applied it to the CNN/Daily Mail corpus. Nallapati et al. [3] proposed an RNN-based sequence model called SummaRuNNer for extractive summarization, which considers the extraction problem as a two-class problem. Narayan et al. [4] proposed the model REFRESH, which considers the extractive automatic summarization as a sentence ordering problem.

Although the extractive summarization is grammatically and syntactically guaranteed, it also faces certain problems, such as wrong choice of content, poor consistency, and poor flexibility.

### B. Abstractive Summarization

The abstractive summarization model has more freedom and can create more novel sequences. Most of the current abstractive models are based on the encoder-decoder mechanism. In recent years, research has focused on solving the problems of duplicates, inaccuracies, and inability to generate out-of-vocabulary (OOV) words in abstractive summaries.

To address the OOV problem, Nallapati et al. [11] proposed the Switching Generator/Pointer mechanism, which allows the decoder network to point back subsequences of the input and copy them into the output sequence. Gu et al. [12] and Zhou et al. [13] proposed the copy mechanism to solve it. More importantly, the abstractive summarization model to handle long text is not efficient. Cao et al. [14] used the ground truth summary to guide the generation of the summary.

### C. Reinforce-Guided Summarization

Paulus et al. [15] used reinforcement learning on encoders and decoders to mitigate exposure bias problems. Celikyilmaz et al. [16] used multi-agents to solve the problem of inaccurate long text summarization generation. Chen et al. [6] used a reinforcement learning network to bridge extractors and abstractors to deal with long text summarization generation. Our model is similar, but Chen et al. assume that each digest sentence is generated from a sentence in the original text, and we believe that each summary sentence is usually relevant with multiple sentences in the original text and should be generated by multiple sentences.

## III. Model Formulation

In this section, we describe our proposed Abstractive Summarization after Extraction (ASAE) model, which consists of an extractor to extract the informative sentences and an abstractor to generate the summarization. In the following, we will give the details of (1) our baseline model framework, (2) the multi-sentence extraction, and (3) the pointer-generator and coverage mechanisms.

### A. The Training Framework of ASAE

Our model is derived from the baseline model proposed in [6], which puts forward a hybrid extractive-abstractive network. Similar to the way that humans summarize long documents, the model extracts some salient sentences from the original documents by the extractor, and then, the abstractor generates summaries from the extracted sentences. However, the baseline model simply rewrites each extracted sentence to generate the final summary, and fails to induce the important information from multiple sentences. Differently, our model generates each summary sentence according to multiple sentences extracted from the original text, in order to cover more important information. Fig. 1 depicts the overall training framework of the proposed ASAE model.

The ASAE model training follows the reinforcement learning process. In Fig. 1, $d_i$ stands for the sentences in the original document. The generated summary sentences are denoted as $[\{\mathrm{d}_{jt}\}]$. The groundtruth sentences are represented by $s$. $v_{EOE}$ stands for the extraction candidate in the sentences, and "EOE" means "end of extraction". When $v_{EOE}$ is extracted, the process of sentence extraction ends. $g(\cdot)$ is the function of rewriting the extraction sentences.

First, we feed the entire document into the extractor and get the salient sentences as outputs. Then, if the extracted sentence is not $v_{EOE}$, it means that the extraction process is not over, and the reward is 0. On the contrary, if the $v_{EOE}$ is extracted, it means the end of the extraction process. Finally, we use the extracted sentences as input to the abstractor network, and get the summary as outputs, and at the same time, the agent obtains the rewards. After the abstractor summarizes the extracted sentences, we use $\theta = \{\theta_a, \omega\}$ as the trainable parameters of the extractor agent for the decoder and the layered encoder, respectively. Then we can train the extractor with a policy-based reinforcement learning process.

In the following part, we will introduce the technical details of the extractor and abstractor of ASAE model.

### B. Muli-sentence Extraction Improvement

We first introduce the structure of the extractor. The entire document is fed into the extractor and the salient sentences are output. First, we employ the temporal convolutional model to obtain the representation of each individual sentence in the document, denoted by $r_j$. Next, the encoder (bidirectional LSTM, shown in green in Fig. 2) re-encodes $r_j$ into $h_j$, and the decoder selects the salient sentences based on $h_j$. At the decoding step $t$, the decoder receives the sentence $h_{j_{t-1}}$ extracted at the previous step $t-1$ as input, and produces a decoder

978-1-7281-4762-8/19/$31.00 © 2019 IEEE

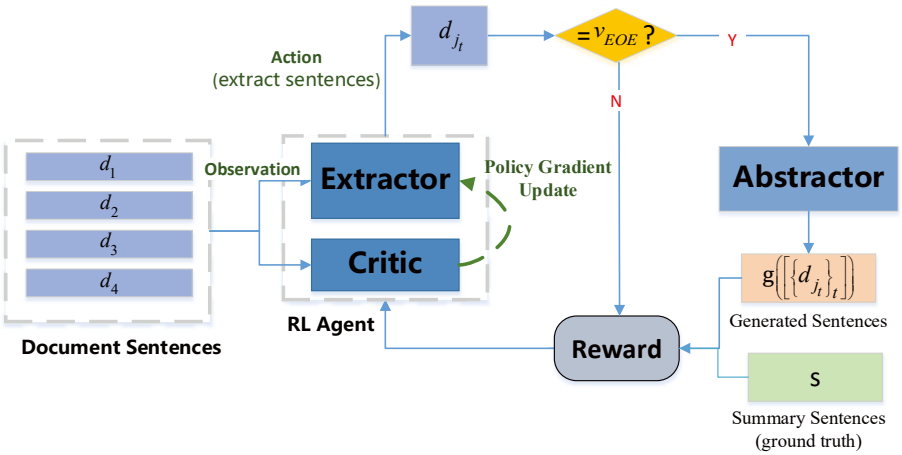


Fig. 1. The framework of ASAE model.

state $s_t$. The calculation formula of attention distribution $a^t$ is as follows:

$$\alpha_j^t = v_g^T \tanh(W_{g1} h_j + W_{g2} s_t) \quad (1)$$

$$a^t = \text{softmax}(\alpha^t) \quad (2)$$

where $v_g$, $W_{g1}$ and $W_{g2}$ are learnable parameters. The attention distribution $a^t$ can be taken as the significance weight distribution of individual sentences in the input document. The context vector can be calculated based on the attention distribution:

$$c_t = \sum\nolimits_j a_j^t W_{g1} h_j \quad (3)$$

The context vector $c_t$ plays as a fixed size representation of the content read from the input document sentence representation at the decoding step $t$.

Next, the extraction probability of each sentence can be estimated based on $c_t$.

$$u_j^t = \begin{cases} v_p^T \tanh(W_{p1} h_j + W_{p2} c_t) & \forall k < t, j_t \neq j_k \\ -\infty & \text{otherwise} \end{cases} \quad (4)$$

and

$$P(j_t | j_1, \ldots j_{t-1}) = \text{softmax}(u^t) \quad (5)$$

Here $v_p$, $W_{p1}$ and $W_{p2}$ are learnable parameters, $t$ represents the decoding step $t$ time, $j_k$ represents all previously extracted sentences, and sentences that have already been extracted cannot be extracted again, i.e., $u_j^t = -\infty$.

### C. The Abstractor

In our model, the abstractor needs to generate a summary from multiple sentences. There exist many challenges in the summary generation of long documents in the conventional seq2seq model, such as repetitive problems and OOV problems.

For the abstractor in our model, we replace the seq2seq model with the pointer-generator network shown in the Fig. 3, since pointer-generator network is good at dealing with OOV words. We solved the problem of long-term dependencies by turning a long document into a short document through generating extractive summaries. Specifically, we originally extract summary sentences from mutually exclusive sentences, so as to avoid the problem of sentence redundancy. For the problem of duplicate words in the model, we use the coverage mechanism to handle it.

*1) Pointer-Genarator Network:* The inputs of the abstractor are important sentences extracted from the original text, and the outputs are summaries. $p_{gen}$ is used as a soft switch to decide whether to generate words from the vocabulary based on vocabulary distribution, or directly copy words in the input sequence based on attention distribution.

$$p_{gen} = \sigma(w_{h^*}^T c_t + w_s^T s_t + w_x^T x_t + b_{ptr}) \quad (6)$$

Then, we predict that the final distribution of words is

$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum\nolimits_{i:w_i=w} a_i^t \quad (7)$$

where $c_t$ represents context vector, $s_t$ represents decoder states, and $x_t$ represents the decoder input, $a^t$ is attention distribution and $w_{h^*}$, $w_s$, $w_x$ and $b_{ptr}$ are learnable parameters.

*2) Coverage Mechanism:* We define a coverage loss $\text{covloss}_t$ to penalize words in the input sequence that scored too high attention scores in past decoding steps.

$$\text{covloss}_t = \sum\nolimits_i \min(a_i^t, c_i^t) \quad (8)$$

During the training process, the loss of time step $t$ is the negative log likelihood of the target word $w_t^*$ of the time step plus the coverage loss.

$$\text{loss}_t = -\log P(w_t^*) + \lambda \sum\nolimits_i \min(a_i^t, c_i^t) \quad (9)$$

And the total loss of the entire sequence is

$$\text{loss} = \frac{1}{\text{T}} \sum\nolimits_{t=0}^{T} \text{loss}_t \quad (10)$$

978-1-7281-4762-8/19/$31.00 

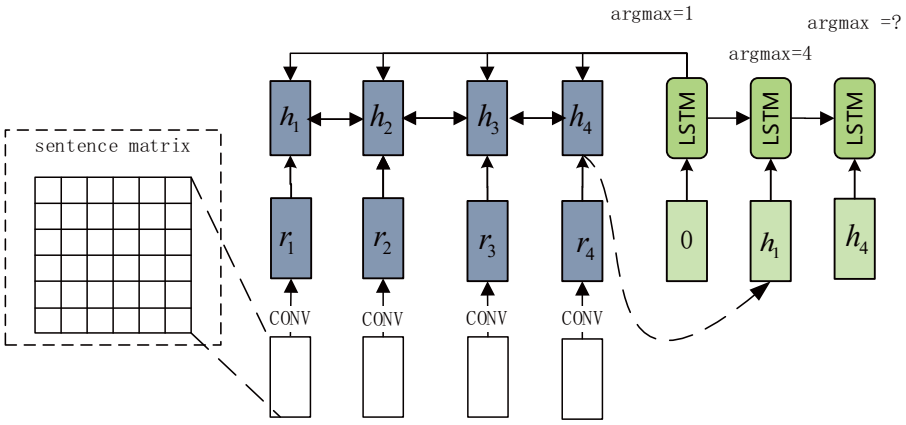


Fig. 2. The structure of the extractor.

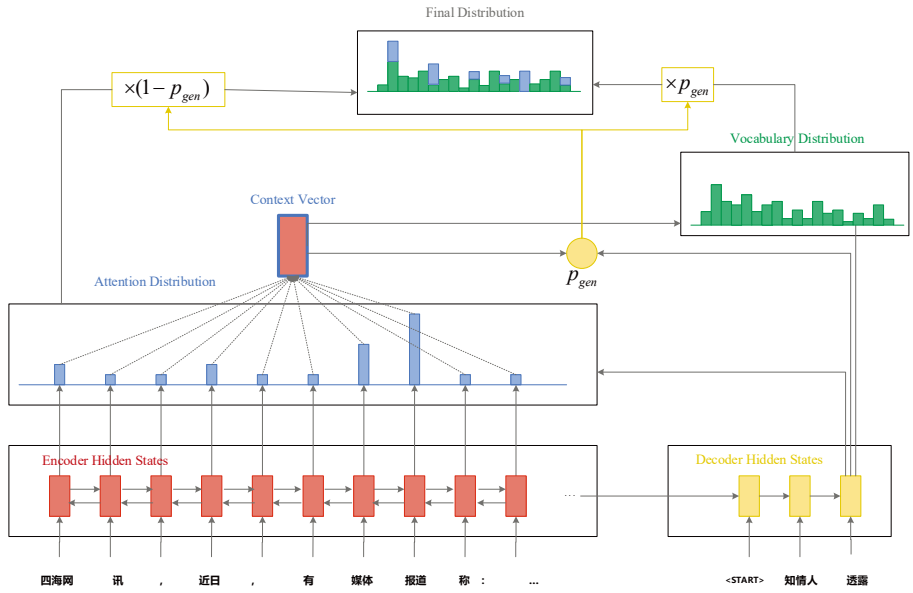


Fig. 3. Pointer-generator network.

### D. Reinforcement Learning Design

In ASAE model, the extractor and abstractor are combined in the reinforcement learning process. We use A2C-based reinforcement learning to adjust the parameters of the extractor. In the work of literature [6], the extracted sentences are fed to the abstractor one by one. Then the output of the abstractor will be compared with the entire reference summary, and the reward is used to adjust the parameters of the extractor. In our model, in order to generate the summary from multiple salient sentences, $v_{EOE}$ is employed during the reinforcement learning process. $v_{EOE}$ is used as one of the extraction candidates. If the extracted sentence is not $v_{EOE}$, it means that the extraction process is not over, and the reward is 0. On the contrary, if the $v_{EOE}$ is extracted, it means the end of the extraction process, and the following rewards are obtained:

$$r(t+1) = ROUGE\_L_{F_1}\left(g\left(\left[\{d_{j_t}\}_t\right]\right), s\right) \qquad (11)$$

After extracting the sentences, we put the previously extracted sentences together as the input of the abstractor, and get the generated summary $g\left(\left[\{d_{j_t}\}_t\right]\right)$ from the abstractor. Then, the generated summary is compared with the reference summary, and the evaluation result $r(t+1)$ is used as a reward to adjust the parameters of the extractor. So far, we have completed the work of generating a summary of a document.

## IV. Experiment

In this section, we will report the empirical evaluation of our model on real dataset for long document summarization.

### A. Dataset

In this paper, we consider a large corpus of Chinese single text summarization, TTNews, which is provided by NLPCC2018 Shared Task 3, to evaluate the performance of our model and other different methods.

978-1-7281-4762-8/19/$31.00 © 2019 IEEE

The TTNews corpus is a news dataset that includes 50000 news-summary pair training sets, 2000 news-summary versus validation sets, and 2000 news (no reference summaries) test sets, which are written by experts from the Byte Beat Company. The news data come from a variety of media, including a wide range of topics covering sports, food, entertainment, politics, technology, and economics. The data set is recorded in text file. The content is represented in json format and contains three keys: summarization, article and index, corresponding to the abstract, original and number.

It is also worth noting that TTNews is a long text summary dataset, and the average length statistics for news and abstracts in the dataset are shown in Table I.

TABLE I
TTNEWS TEXT LENGTH STATISTICS.

| Length | TrainSet | | ValidateSet | | TestSet | |
|---|---|---|---|---|---|---|
| | News | Summary | News | Summary | News | Summary |
| Min | 32 | 6 | 35 | 8 | 7 | NA |
| Max | 13186 | 85 | 8871 | 44 | 7596 | NA |
| Average | 579.6 | 25.9 | 579.9 | 25.8 | 7426.2 | NA |

### B. Data Preprocessing and Parameter Setting

We preprocessed the data by filtering out duplicate news-summary pairs and invalid news-summary pairs (Invalid situations include 1. Abstracts missing; 2. Original news missing; 3. News and abstracts do not match). Before using the data, we adopted jieba word segmentation tool to process the word segmentation.

Our abstractive model has a 600-dimensional implicit state bidirectional LSTM encoder and a 1200-dimensional implicit state unidirectional LSTM decoder with only one layer per encoder and decoder and a vocabulary size of 50000. For maximum likelihood training of extractor and abstractor, we use Adam as the default hyperparameter optimization algorithm (learning rate $a = 0.001$). For the reinforcement learning training of the extractor, the learning rate is set to 0.0001 and the discount factor $\gamma$ is set to 0.95. We conducted our experiments on the NVIDIA GTX1080TI GPU and the entire model was trained for 40 hours.

### C. Experiment Design

For the proposed ASAE model, we perform the experiments as the following methods:

- Verify the various mechanisms used in the abstractor network model and observe the impact of different mechanisms on the entire ASAE model.
- The ASAE model was compared with multiple models of NLPCC2018.

The ASAE model is compared with the multiple entry models of NLPCC2018 and the baseline model of this work, named as **fast_abs_rl**. The descriptions of these models are as follows:

- CCNU_NLP [17]: This model is based on the pointer-generator network model, and the idea is to extract the keywords in the document, and use these keywords as part of the input of the model, so as to improve the ability of the model to notice the key information in the document.
- Summary++ [18]: This model is based on the pointer-generator network model, but removes the vocabulary generation function and only applies the pointer generation function. That is, in the decoding step, each predicted word is derived from the original document. In addition, this model is different from most of other models because it does not use Chinese words after the word segmentation as inputs, but directly uses Chinese characters as inputs.
- DLUT_815 [19]: This is the champion model of the NLPCC2018 Chinese single document summary, and also based on the pointer generation network model. It introduces the Batch Normal Layer and Focal Loss to improve the performance of the model.
- fast_abs_rl [6]: This is the hybrid model proposed by Chen et al., and is the baseline method of the proposed ASAE model.

### D. Experiment Result and Analysis

Because the ROUGE toolkit cannot directly evaluate Chinese, we follow the processing conventions. We first convert Chinese into numbers, and then perform ROUGE evaluation. To ensure the validity of the comparison results, we also measured our results on the NLPCC2017 Shared Task 3 online test site [1]. The optimal experimental results of all models are shown in Table II. The online evaluation of NLPCC2018 Shared Task 3 [2] has been closed, and only the results submitted by previous models can be retrieved, so the model trained in this paper cannot obtain the evaluation results.

TABLE II
THE COMPARISON RESULTS OF THE ONLINE EVALUATION FOR DIFFERENT MODELS.

| Model | ROUGE-1 | ROUGE-2 | ROUGE-L | Test2017 | Test2018 |
|---|---|---|---|---|---|
| CCNU_NLP | 0.4460 | 0.3046 | 0.3883 | - | 0.2828 |
| Summary++ | 0.4600 | 0.3000 | 0.4000 | - | 0.2853 |
| DLUT_815 | **0.5178** | 0.2090 | - | 0.3245 | **0.3129** |
| fast_abs_rl | 0.4933 | 0.3487 | 0.4765 | 0.3258 | - |
| ASAE | 0.5138 | **0.3597** | **0.4975** | **0.3419** | - |

From the results, we can observe that for most of the evaluation scores and online tests, our approach can achieve the best performance. Even for the ROUGE-1 score, the result of ASAE model is very close to the best score achieved by DLUT_815. The remarkable record is that ASAE model can achieve better performance than the baseline approach fast_abs_rl, because the abstractor of ASAE model will generate each of the summary sentences from multiple sentences other than only one, so the generated sentence can deliver more information and achieve better summarization quality.

We also list some representative summarization examples in Fig. 4, where the contents of left column are the original Chinese news documents, the reference summary, and the results generated by fast_abs_rl and ASAE respectively. The English translations are provided in the right column.

[1] https://www.biendata.com/competition/nlptask03/
[2] https://biendata.com/competition/nlpcc2018/

978-1-7281-4762-8/19/$31.00 © 2019 IEEE

| | |
|---|---|
| **Document:** 北京时间 7 月 7 日消息，据《雅虎体育》报道，爆料王阿德里安-沃纳罗斯基称，莫里斯-威廉姆斯已经与克里夫兰骑士达成了签约协议，他将签下一份 2 年 430 万美元的合同。沃纳罗斯基还表示，威廉姆斯曾考虑过多支球队，包括圣安东尼奥马刺，达拉斯小牛和新奥尔良鹈鹕，但他最终还是选择了去克里夫兰，与勒布朗-詹姆斯重聚…… | **Document:** Beijing time on July 7th news, according to "Yahoo Sports" report, the news king Adrian Werner Roskie said that Morris Williams has signed a contract with the Cleveland Cavaliers, he will sign A two-year, $4.3 million contract. Werner Roskie also said that Williams had considered too many teams, including the San Antonio Spurs, the Dallas Mavericks and the New Orleans, but he eventually chose to go to Cleveland and reunite with LeBron James…… |
| **Ground truth summary:** 据雅虎体育，莫里斯-威廉姆斯与骑士达成协议，将与詹皇重聚，他将签下 2 年 430 万美元的合同。 | **Ground truth summary:** According to Yahoo! Sports, Maurice Williams and the Cavaliers reached an agreement to reunite with King James, who will sign a two-year, $4.3 million contract. |
| **fast_abs_rl:** 莫里斯-威廉姆斯与骑士达成签约协议，签下 2 年 430 万美元的合同。 | **fast_abs_rl:** Morris Williams reached a contract with the Cavaliers to sign a two-year, $4.3 million contract. |
| **ASAE:** 莫里斯-威廉姆斯与骑士达成签约协议，将签下 2 年 430 万美元的合同，与詹姆斯重聚。 | **ASAE:** Morris Williams and the Cavaliers signed a contract to sign a two-year, $4.3 million contract to reunite with James. |
| **Document:** 四川新闻网-成都晚报评论假期七天工资该怎么算？据记者了解，今年国庆长假 1 日、2 日、3 日为法定节假日，按照劳动法，用人单位如果法定节假日安排加班，工资是平时的 3 倍。4 日、5 日、6 日、7 日为公休日，如果休息日安排加班，工资是平时的 2 倍。按最低工资标准 1200 元(每日约 55.2 元) 计算，国庆 7 天最低加班费为 938.4 元…… | **Document:** Sichuan News Net - Chengdu Evening News comment on the seven-day salary of the holiday how to calculate? It's reported that on the 1st, 2nd, and 3rd day of the National Day holiday this year is a statutory holiday. According to the Labor Law, if the employer arranges overtime work on a statutory holiday, the salary is three times that of normal. On the 4th, 5th, 6th, and 7th, it is a public holiday. If the rest day is arranged to work overtime, the salary is twice as much as usual. Calculated according to the minimum wage of 1,200 yuan (about 55.2 yuan per day), the minimum overtime pay for the 7-day National Day is 938.4 yuan…… |
| **Ground truth summary:** 国庆加班七天成都最低可领加班费 938.4 元。 | **Ground truth summary:** In Chengdu, the minimum number of overtime pay for employees who work overtime for seven days on National Day is 938.4 yuan. |
| **fast_abs_rl:** 最低工资 1200 元，国庆最低加班费为 938.4 元。 | **fast_abs_rl:** The minimum wage is 1,200 yuan, and the minimum overtime pay for the National Day is 938.4 yuan. |
| **ASAE:** 成都今年国庆长假假期七天工资 3 倍：国庆 7 天最低加班费为 938.4 元。 | **ASAE:** In Chengdu, national holiday seven days salary three times: National Day 7 days minimum overtime pay is 938.4 yuan. |

Fig. 4. Text summarization examples.

## V. Conclusion

In order to deal with the long-term dependency of the abstractive summary model in dealing with long texts, in this paper, we proposed a hybrid approach to make use of both extractive and abstractive approach. The extractor extracts the important sentences in the original text, and the abstractor rewrites the summary sentences based on multiple extracted sentences to get the final summarization, which solves the shortcomings of the existing abstractive models in dealing with long texts. The experimental results on NLPCC2017 and NLPCC2018 Shared Task 3 show that our model is effective and can achieve better ROUGE scores.

## References


[1] R. Mihalcea and P. Tarau, "Textrank: Bringing order into texts," in *Conference on Empirical Methods in Natural Language Processing*, 2004, pp. 404–411.

[2] G. Erkan and D. R. Radev, "Lexrank: graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, no. 1, pp. 457–479, 2004.

[3] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *The 31st AAAI Conference on Artificial Intelligence*, 2017, pp. 3075–3081.

[4] S. Narayan, N. Papasarantopoulos, M. Lapata, and S. B. Cohen, "Neural extractive summarization with side information," *arXiv: Computation and Language*, 2017.

[5] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *The 55th annual meeting of the Association for Computational Linguistics*, 2017, pp. 1073–1083.

[6] Y.-C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," in *The 56th annual meeting of the Association for Computational Linguistics*, 2018, pp. 675–686.

[7] J. Carbinell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," *SIGIR Forum*, vol. 51, no. 2, pp. 209–210, 2017.

[8] R. T. McDonald, "A study of global inference algorithms in multi-document summarization," in *European Conference on Information Retrieval*, 2007, pp. 557–564.

[9] M. Krageback, O. Mogren, N. Tahmasebi, and D. P. Dubhashi, "Extractive summarization using continuous vector space models," in *The 2nd Workshop on Continuous Vector Space Models and their Compositionality*, 2014, pp. 31–39.

[10] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," in *The 54th annual meeting of the Association for Computational Linguistics*, 2016, pp. 484–494.

[11] R. Nallapati, B. Zhou, C. N. dos Santos, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," in *Conference on Computational Natural Language Learning*, 2016, pp. 280–290.

[12] J. Gu, Z. Lu, H. Li, and V. O. K. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *The 54th annual meeting of the Association for Computational Linguistics*, 2016, pp. 1631–1640.

[13] Q. Zhou, N. Yang, F. Wei, and M. Zhou, "Sequential copying networks," in *The 32nd AAAI Conference on Artificial Intelligence*, 2018, pp. 4987–4995.

[14] Z. Cao, W. Li, S. Li, and F. Wei, "Retrieve, rerank and rewrite: Soft template based neural summarization," in *The 56th annual meeting of the Association for Computational Linguistics*, 2018, pp. 152–161.

[15] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," in *International Conference on Learning Representations*, 2018.

[16] A. Celikyilmaz, A. Bosselut, X. He, and Y. Choi, "Deep communicating agents for abstractive summarization," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 1662–1675.

[17] X. Jiang, P. Hu, L. Hou, and X. Wang, "Improving pointer-generator network with keywords information for chinese abstractive summarization," in *The 7th International Conference on Natural Language Processing and Chinese Computing*, 2018, pp. 464–474.

[18] J. Zhao, T. L. Chung, B. Xu, and M. Jiang, "Summary++: Summarizing chinese news articles with attention," in *The 7th International Conference on Natural Language Processing and Chinese Computing*, 2018, pp. 27–37.

[19] Y. Shi, J. Meng, J. Wang, H. Lin, and Y. Li, "A normalized encoder-decoder model for abstractive summarization using focal loss," in *The 7th International Conference on Natural Language Processing and Chinese Computing*, 2018, pp. 383–392.


978-1-7281-4762-8/19/$31.00 