Sharing Pre-trained BERT Decoder for a Hybrid Summarization

Ran Wei[1,2], Heyan Huang[1,2], and Yang Gao[1,2]

[1] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[2] Beijing Engineering Research Center of High Volume Language Information Processing and Cloud Computing Applications, Beijing 100081, China
weiranbit@gmail.com,{hhy63, gyang}@bit.edu.cn

Abstract. Sentence selection and summary generation are two main steps to generate informative and readable summaries. However, most previous works treat them as two separated subtasks. In this paper, we propose a novel extractive-and-abstractive hybrid framework for single document summarization task by jointly learning to select sentence and rewrite summary. It first selects sentences by an extractive decoder and then generate summary according to each selected sentence by an abstractive decoder. Moreover, we apply the BERT pre-trained model as document encoder, sharing the context representations to both decoders. Experiments on the CNN/DailyMail dataset show that the proposed framework outperforms both state-of-the-art extractive and abstractive models.

Keywords: Text Summarization · Extractive and Abstractive · Pre-trained Based.

1 Introduction

Automatic text summarization has played an important role in a variety of natural language processing (NLP) applications, such as question answering [13, 21], report generation [14], and opinion mining [8]. Single document summarization, the task of generate short, representative and readable summaries from the original text while retaining the main ideas of source articles, has received much attention in recent years [16, 20, 5, 18].

Current methods for single document summarization using neural network architectures have primarily focused on two strategies: extractive and abstractive. Extractive summarization forms summaries by selecting originally important segments of the input documents [15, 17]. Abstractive summarization potentially generates new sentence or reorganizes their orders to form fluent summaries [20, 18, 6]. Both methods suffer from obvious drawbacks: extractive systems are sometimes redundant since they cannot trim the original sentences to fit into the summary, and they lack a mechanism to ensure overall coherence. In contrast, abstractive systems require natural language generation and semantic representation, which are complex and can hardly meet the demands of generating correct facts with proper word relations.

In this paper, we present a novel architecture that attempts to combine the extractive and abstractive methods. Our model first decides whether to choose a sentence based on its probability generated by an extractive decoder, and then rewrite the selected sentence by an abstractive decoder, which can remove the meaningless words, reorganize words orders and generate coherent contents. In this way, our model can extract informative contents and generate coherent summaries. Moreover, we choose Bidirectional Encoder Representations from Transformers (BERT [3]) pre-trained language model as basic document encoder, which can provide powerful pre-trained context representations. Both decoders share the same representations, so that our model can be trained simultaneously in one end-to-end framework. Our contributions in this paper are two-folds:

1. We propose a novel extractive-and-abstractive hybrid neural architecture combining the extractive and abstractive decoders, taking advantage of both summarization approaches.
2. We explore a new way that applies pre-trained language model into summarization task, making good use of the pre-trained context representations in the sharing encoder process.

Extensive experiments are conducted on CNN/DailyMail datastet [9, 16]. The results show that our model outperforms both extractive and abstrctive state-of-the-arts models.

The rest of this paper is organized as follows. We present the related work in Section 2. In Section 3, we introduce our extractive-and-abstractive hybrid model in details. In Section 4, we describe the experiments setup and implementation details. We present the results of our experiments and analysis the performance in Section 5. We conclude our work in Section 6.

2 Related work

2.1 Extractive summarization

Kageback et al. [10] and Yin and Pei [22] use neural networks to map sentences into vectors and select sentences based on those vectors. Cheng and Lapata [2] seletct sentences based on an LSTM classifier that predicts a binary label for each sentence. Nallapati et al. [15] adopt a similar approach, they differ in their neural architecture for sentence encoding and features used during label prediction, while Narayan et al. [17] equip the same architecture with a training algorithm based on reinforcement learning. While some extractive summarization methods obtain high ROUGE scores, most of them lack a machanism to ensure overall coherence and suffer from low readability.

2.2 Abstractive summarization

Rush et al. [19] first bring up the abstractive summarization task and use attention-based encoder to read the input text and generate the summary. Nallapati et al. [16] apply a more powerful sequence-to-sequence model. See et al. [20] combine pointer networks into their models to deal with the out-of-vocabulary (OOV) words. Paulus et al. [18] use policy gradient on summarization and state out the fact that high ROUGE scores might still lead to low human evaluation scores. However, most of them underperform or are on par with the baseline of simply selecting the leading sentences in the document as summaries, the best results for abstractive summarization have been achieved with models that are more extractive in nature than abstractive, since most of the words in the summary are copied from the document (Gehrmann et al. [6])

2.3 Pre-trained model summarization

Edunov et al. [4] combine the pre-trained embedding to the encoder network to enhance the text representations. Zhang et al. [23] propose a BERT based encoder-decoder framework, which use BERT as encoder and refine every word in the draft summary. All these methods demonstrate a pre-trained model on vast corpora can provide improvements for summarization task.

In this paper, we propose a novel hybrid end-to-end architecture combining extractive and abstractive model by using the extractive decoder to select informative sentences and rewrite these selected contents by a abstractive decoder. Both decoders share the same pre-trained representations provided by a BERT encoder.

3 Our Model

Our model extracts sentences from a given document and further rewrites these sentences by a sequence-to-sequence architecture. We denote a document $D = (s_1,...,s_M)$ as a sequence of M sentences, and a sentence $s_i = (w_{i1},...,w_{iN})$ as a sequence of N words. A extractive decoder are used assign a label $z_i \in \{0, 1\}$ to each sentence s_i. $z_i = 1$ indicates s_i is selected and $z_i = 0$ means s_i is skipped. On the other hand, the abstractive decoder generates the summary text $y_j = (w_{j1},...,w_{jK})$, where y_j is the summary rewritten by the j^{th} selected sentence. Only the sentences with $z_i = 1$ will go through the abstractive decoder. Fig. 1 demonstrates an overview of our proposed model. In the following, we introduce each of its components in details.

3.1 Document Encoder

We use BERT as the basic document encoder, because extractive and abstractive decoder share the same pre-trained document encoder. We require it to output

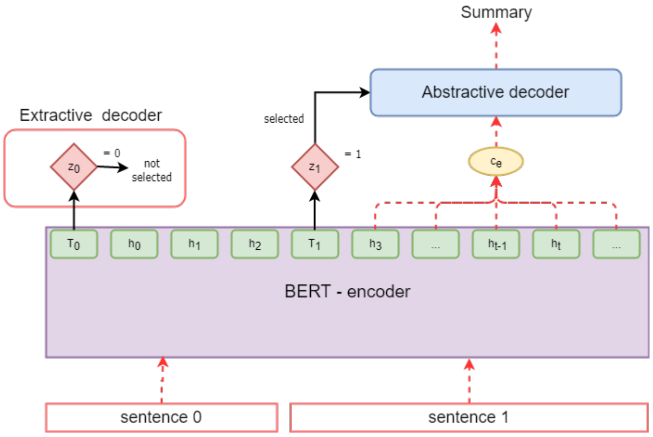


Fig. 1. Framework of our summarization system. The model extracts the most relevant sentences by taking into account the sentence representation T_i. If a sentence is selected ($z_i = 1$), its word level representations are fed into the abstractive decoder to generate final summary.

representations in sentence level and word level. To get the representation for each sentence, we adopt similar modifications [12] to the input sequence and embedding of BERT, we insert a [CLS] token before each sentence and use interval segment embeddings to distinguish multiple sentences within a document. As illustrated in Fig. 2, the vector T_i which is the i^{th} [CLS] symbol from the top BERT layer will be used as the representation for sentence s_i, the vector h_i is the representation for each word.

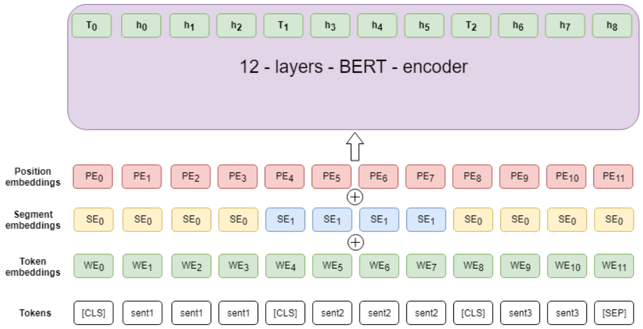

T_0
h_0
h_1
h_2
T_1
h_3
h_4
h_5
T_2
h_6
h_7
h_8
12 - layers - BERT - encoder
Position embeddings
PE_0 PE_1 PE_2 PE_3 PE_4 PE_5 PE_6 PE_7 PE_8 PE_9 PE_{10} PE_{11}
Segment embeddings
SE_0 SE_0 SE_0 SE_0 SE_1 SE_1 SE_1 SE_1 SE_0 SE_0 SE_0 SE_0
Token embeddings
WE_0 WE_1 WE_2 WE_3 WE_4 WE_5 WE_6 WE_7 WE_8 WE_9 WE_{10} WE_{11}
Tokens
[CLS] sent1 sent1 sent1 [CLS] sent2 sent2 sent2 [CLS] sent3 sent3 [SEP]

Fig. 2. Document encoder architecture. We insert a [CLS] token before each sentence and use interval segment embeddings to distinguish different sentences.

3.2 Extractive Decoder

The extractive decoder selects the most informative sentences, which cover necessary information that is belonged to the gold summary. For each sentence s_i, the extractive decoder takes a decision based on the encoder representation T_i which is the [CLS] symbol representation from the top BERT encoder layer for s_i. Benefit from self-attention mechanism in the multi-layer transformer based BERT encoder, the sentence representation T_i not only contains the semantic information of s_i but also represent the relationship between s_i and other sentences in the document. The extractive decoder adds a linear layer into the encoder outputs T_i and compute the probability of action $z_i \in \{0, 1\}$ to sentence s_i as:

$$p(z_i|T_i) = \sigma(W_0 T_i + b_0) \tag{1}$$

where W_0 and b_0 are the model parameters, σ is the Sigmoid function. To optimize the extractive decoder, We use a Cross Entropy Loss:

$$L_{ext} = -\frac{1}{N}\sum_{i=1}^{N}(l_i \ln p(z_i = 1|T_i) + (1 - l_i)\ln(1 - p(z_i = 0|T_i)) \tag{2}$$

where $l_i \in \{0, 1\}$ is the ground-truth label for sentence s_i and N is the number of sentences. When $l_i = 1$, it indicates that sentence s_i should be ectracted and be attended to the abstractive decoder.

3.3 Abstractive Decoder

After the sentence s_i is selected in the extractive decoder, we input s_i into the abstractive decoder to rewrite the original sentence in a more abstractive way. In practice, we introduce a 4 layer Transformer decoder to learn the conditional probability $P(y_i|T_i)$, where T_i is the hidden representation for s_i from the BERT encoder, y_i is the gold summary sentence.

Transformer decoder Transformer decoder takes a shifted sequence of target summary word embeddings as input and produces contextualized representations $o_1, ..., o_n$, from which the target tokens are predicted by a softmax layer. As shown in Eq.(3), at step t, the decoder predicts output probability $P_t^{vocab}(w)$ conditioned on previous outputs $c_{<t}$ and encoder outputs c_e as follow:

$$P_t^{vocab}(w) = softmax(W_1[c_{<t}, c_e] + b_1) \tag{3}$$

$$c_{<t} = \sum_{j=1}^{t-1} \alpha_j^{<t} o_j \tag{4}$$

$$c_e = \sum_{j=1}^{e} \alpha_j^{e} h_j \tag{5}$$

$$\alpha_j^{<t} = Multihead - attention(o_j, o_1, ..., o_{t-1}) \tag{6}$$

$$\alpha_j^{e} = Multihead - attention(h_j, h_1, ..., h_m) \tag{7}$$

The transformer decoder calculates source attention α_j^e over the representations $h_1, ..., h_m$ of selected sentence, while the target side self attention $\alpha_j^{<t}$ should not be able to look at the representations of later positions. The decoder's learning objective is to minimize negative likelihood of conditional probability as follow:

$$L_{abs} = -\frac{1}{C}\sum_{t=1}^{C} \ln(P(w_t = w_t^* | c_{<t}, c_e)) \tag{8}$$

where w_t^* is the word in gold summary at step t.

Copy mechanism As some summary tokens are out-of-vocabulary(OOV) words yet occur in the selected sentences, we incorporate copy mechanism(Gu et al. [7]) into the Transformer decoder. At decoder time step t, we first calculate the attention probability distribution over the selected sentence s_i using dot product of the last layer transformer decoder output o_t and the encoder output h_j as the following:

$$u_t^j = o_t W_c h_j \tag{9}$$

$$\alpha_t^j = \frac{\exp(u_t^j)}{\sum_{k=1}^{N} \exp(u_t^k)} \tag{10}$$

We then calculate copying gate $g_t \in [0, 1]$, which represents the probability of selecting words from source sentence:

$$g_t = sigmoid(W_g[o_t, h] + b_g) \tag{11}$$

$$h = \sum_{j=1} \alpha_t^j h_j \tag{12}$$

Finally we use g_t to calculate the final probability at time step p, which is a weighted sum of copy probability and generation probability:

$$P_t(w) = (1 - g_t)P_t^{vocab}(w) + g_t \sum_{i:w_i = w} a_t^i \tag{13}$$

3.4 Learning and Inference

During training stage, the goal is the combination of extractive and abstractive loss. We use both sentence level label and ground-truth summary jointly training our model and minimizing the following objective:

$$L_{model} = L_{ext} + L_{abs} \tag{14}$$

In the inference stage, we calculate a decision variable $z_i = 1$ if $p(z_i|T_i) > 0.5$ for selecting the sentence s_i, and $z_i = 0$ if $p(z_i|T_i) < 0.5$ for skiping this sentence. To control the length of summaries, we only keep the top 4 selected sentences with the max $p(z_i|T_i)$ if too many sentences selected by our extractive decoder. We use beam search to generate the abstractive summaries.

4 Experiments Setup

In this section, we introduce the summarization dataset, implementation details and evaluation protocol in our experiments.

4.1 Dataset And Preprocess

Experiments are performed on the CNN/DailyMail dataset (Hermann et al. [9]; Nallapati et al. [16]; See et al. [20]) which contains news stories in CNN and Dauly Mail websites. We used the standard splits of Hermann et al. [9] for training, validation, and testing (287,113 training pairs, 13,368 validation pairs and 11,490 test pairs.) We follow See et al. [20] and use the non-anonymized version data.

Data preprocess To train the extractive decoder in our model, sentence level labels are needed, which indicate the sentence should be selected or not. Besides we need to align each selected sentence to one ground-truth summary, so that we can use the abstractive decoder to learn this kind of corresponding relationship. However, CNN/DailyMail dataset only contains abstractive gold summaries, we apply a greedy preprocess algorithm to generate these labels. For each sentence in the gold summary, we caculate ROUGE-1, ROUGE-2 and ROUGE-L to every sentence in the document and assign label 1 to the sentence with maximum sum of these three ROUGE scores, here all the ROUGE scores are the recall value because we want these selected sentences contain complete information of the gold summaries. In order to keep the origin context order of the selected sentences, we also apply a greedy algorithm, for example we have a document $D = (s_1,...,s_M)$, a gold summary $S = (y_1,\ y_2)$ and y_1 aligns to the s_i, we will find the selected sentence aligns to y_2 only in the sentences set $(s_{i+1},...,s_M)$.

4.2 Implementation Details

In this work, we use the 'bert-base-uncased' version BERT as document encoder. We use the same WordPiece vocabulary (30522 words) for both encoder and abstractive decoder and set the transformer layer to 4, and set the attention heads number to 12. We train the model using an Adam optimizer with learning rate of $5e-4$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and use dynamic learning rate. BERT encoder, extractive and abstractive decoder are jointly trained for 10 epochs on 4 GPUs(GTX 1080 Ti with 11GB memory) with a batch size to 24(6 in each GPU). Model checkpoints are saved and evaluated on the validation set every 5000 steps.

4.3 Model Evaluation

We evaluated summarization quality using ROUGE F1(Lin and Hovy [11]), which is the standard evaluation metric for summarization . We report results in terms of unigram and bigram overlap (ROUGE-1) and (ROUGE-2) as a means of assessing informativeness, and the longest common subswquence (ROUGE-L) as a means of assessing fluency.

5 Results and Analysis

In this section, we first compare our model with both extractive and abstractive baselines on benchmark datasets. We then conduct ablation experiments to study the effect of the hybrid decoders. Finally, we present some examples output from our model.

5.1 Evaluation Results

Table 1. Testing results on the CNN/DailyMail dataset using ROUGE F1.

Model	ROUGE-1	ROUGE-2	ROUGE-L
PGN	39.53	17.28	37.98
BOTTOM-UP	41.22	18.68	38.34
DCA	41.69	19.47	37.92
LEAD	40.42	17.62	36.67
REFRESH	41.0	18.80	37.70
NEUSUM	41.59	19.01	37.98
SRC-ELMO + SHDEMB	41.56	18.94	38.47
TWO-STAGE + RL	41.71	**19.49**	38.79
our model	**41.76**	19.31	**38.86**

The experimental results on CNN/Dailymail dataset are shown in Table 1. Our model aims to take advantage of both extractive and abstractive approaches, so we compare our model with several previously proposed extractive and abstractive systems.

- **LEAD** is an extractive baseline which uses the first-3 sentences of the document as a summary.
- **REFRESH**(Narayan et al. [17]) is an extractive summarization system trained by globally optimizing the ROUGE metric with reinforcement learning.
- **NEUSUM**(Zhou et al. [24]) is the state-of-the-art extractive system that jontly score and select sentences.
- **PGN**(See et al. [20]), is the Pointer Generator Network, an abstractive summarization system based on an encoder-decoder architecture.
- **BOTTOM-UP**(Gehrmann et al. [6]), is a state-of-the-art abstractive summarization system using a bottom-up attention.

- **DCA**(Celikyilmaz et al. [1]) is the Deep Communicating Agents, a state-of-the-art abstractive summarization system with multiple agents to represent the document as well as hierarchical attention mechanism over the agents for decoding.

We also compare our model with two pre-trained based summarization approaches.

- **SRC-ELMO + SHDEMB**(Edunov et al. [4]) is an abstractive model using pre-trained embedding to enhance text representations.
- **TWO-STAGE + RL**(Zhang et al. [23]) is a two stage encoder-decoder model, which applies BERT to the encoder and draft-refine part.

As illustrated in the Table 1, our model outperforms both extractive and abstractive previous baselines. We get a state-of-the-art result in ROUGE-1 and ROUGE-L, which shows that our model can generates informative and coherent summaries. On the ROUGE-2 metric, our model is also comparable with most baselines. Only the DCA and TWO-STAGE + RL outperform our model, in which both apply reinforcement learning for optimizing objective directly derived by ROUGE metrics. Compared to the SRC-ELMO + SHDEMB and TWO-STAGE + RL model, which use pre-trained representations, our model makes a better use of pre-trained language model and achieves better performance.

5.2 Ablation Study

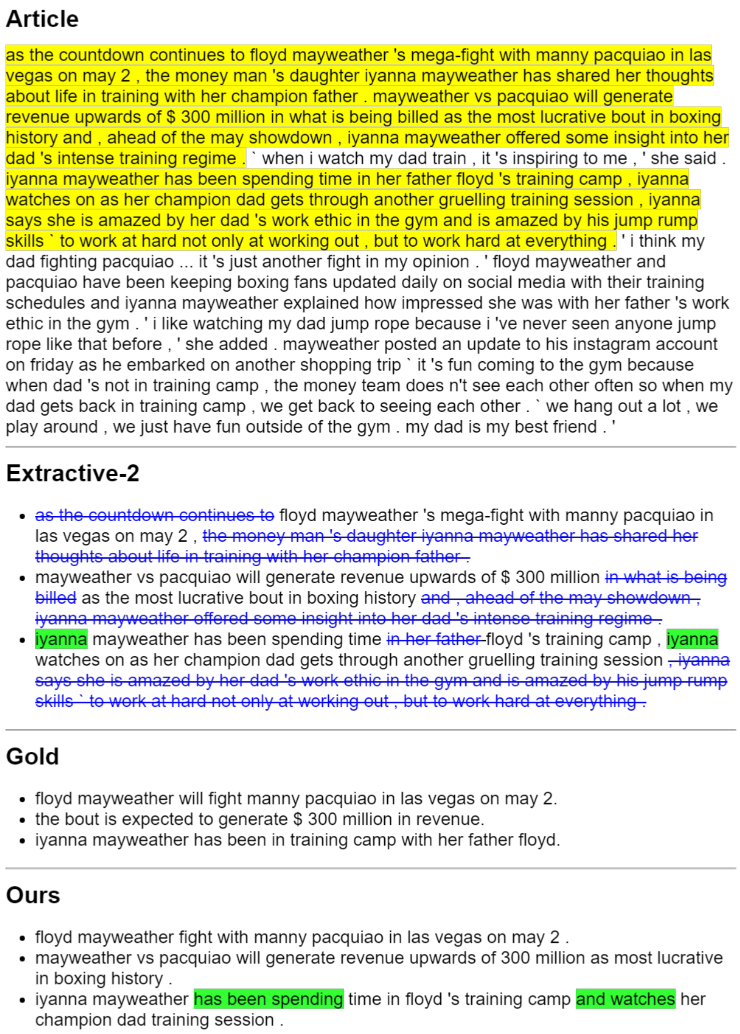
Additionally, in order to study the effect of our abstractive decoder, we perform two extensive baselines.

- **Extractive-1** just uses the sentence level label to train the extractive decoder and select summaries from document.
- **Extractive-2** uses both sentence level label and ground-truth summaries to train complete model, but generates summaries only according to the extractive decoder.

Table 2. Test set results for ablation study.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Extractive-1	40.54	17.97	37.04
Extractive-2	41.15	18.74	37.51
Complete model	41.76	19.31	38.86

The ablation results are shown in Table 2. The results of Extractive-2 performs better than Extractive-1, suggesting that jointly training of extractive and abstractive decoder can improve the model's ability of extracting informative con-



Article

as the countdown continues to floyd mayweather 's mega-fight with manny pacquiao in las vegas on may 2 , the money man 's daughter iyanna mayweather has shared her thoughts about life in training with her champion father . mayweather vs pacquiao will generate revenue upwards of $ 300 million in what is being billed as the most lucrative bout in boxing history and , ahead of the may showdown , iyanna mayweather offered some insight into her dad 's intense training regime . ` when i watch my dad train , it 's inspiring to me , ' she said . iyanna mayweather has been spending time in her father floyd 's training camp , iyanna watches on as her champion dad gets through another gruelling training session , iyanna says she is amazed by her dad 's work ethic in the gym and is amazed by his jump rump skills ` to work at hard not only at working out , but to work hard at everything . ' i think my dad fighting pacquiao ... it 's just another fight in my opinion . ' floyd mayweather and pacquiao have been keeping boxing fans updated daily on social media with their training schedules and iyanna mayweather explained how impressed she was with her father 's work ethic in the gym . ' i like watching my dad jump rope because i 've never seen anyone jump rope like that before , ' she added . mayweather posted an update to his instagram account on friday as he embarked on another shopping trip ` it 's fun coming to the gym because when dad 's not in training camp , the money team does n't see each other often so when my dad gets back in training camp , we get back to seeing each other . ` we hang out a lot , we play around , we just have fun outside of the gym . my dad is my best friend . '

Extractive-2

- ~~as the countdown continues to~~ floyd mayweather 's mega-fight with manny pacquiao in las vegas on may 2 , ~~the money man 's daughter iyanna mayweather has shared her thoughts about life in training with her champion father .~~
- mayweather vs pacquiao will generate revenue upwards of $ 300 million ~~in what is being billed~~ as the most lucrative bout in boxing history ~~and , ahead of the may showdown , iyanna mayweather offered some insight into her dad 's intense training regime .~~
- iyanna mayweather has been spending time ~~in her father~~ floyd 's training camp , iyanna watches on as her champion dad gets through another gruelling training session ~~, iyanna says she is amazed by her dad 's work ethic in the gym and is amazed by his jump rump skills ` to work at hard not only at working out , but to work hard at everything .~~

Gold

- floyd mayweather will fight manny pacquiao in las vegas on may 2.
- the bout is expected to generate $ 300 million in revenue.
- iyanna mayweather has been in training camp with her father floyd.

Ours

- floyd mayweather fight with manny pacquiao in las vegas on may 2 .
- mayweather vs pacquiao will generate revenue upwards of 300 million as most lucrative in boxing history .
- iyanna mayweather has been spending time in floyd 's training camp and watches her champion dad training session .

Fig. 3. Example output summaries, article, Extractive-2 result and gold summary. Our entractive decoder selects informative sentences of the article (yellow highlight). The abstractive decoder rewrites sentence by removing unrelevant words (blue part) and generating coherent expressions (green highlight, here "spending" and "watches" are the actions of same subject, our model use "and" to keep the sentence coherence instead of using two separated subject "iyanna").

tents. This improvement mainly benefits from that we use both sentence level label and ground-truth summary to train our model. Compared to the Extractive-2, the complete model achieves significant improvements which demonstrates the abstractive decoder is helpful to generate informative and readable summaries.

5.3 Case Study

We investigate an example of generated output in Fig. 3. Our extractive decoder first extracts 3 informative sentences from document (the sentences in Extractive-2). Then, our abstractive decoder rewrites these sentences and generates the final summaries. More specifically, our abstractive decoder is able to remove the meaningless and unrelevant words in the selected sentences, reorganizes words orders and generates coherent expressions.

6 Conclusion

In this work, we propose a novel extractive-and-abstractive hybrid model combining extractive and abstractive decoder for text summarization task. Experimental results shows that the ability of our model to extract informative contents and generate coherent and readable abstractive summary. Our model outperforms both state-of-the-art extractive and abstractive systems on the CNN/DailyMail dataset.

Acknowledgments

This work is supported by Ministry of Education - China Mobile Research Foundation NO. MCM20170302.

References

1. Celikyilmaz, A., Bosselut, A., He, X., Choi, Y.: Deep communicating agents for abstractive summarization. arXiv preprint arXiv:1803.10357 (2018)
2. Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. arXiv preprint arXiv:1603.07252 (2016)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Edunov, S., Baevski, A., Auli, M.: Pre-trained language model representations for language generation. arXiv preprint arXiv:1903.09722 (2019)
5. Fan, A., Grangier, D., Auli, M.: Controllable abstractive summarization. arXiv preprint arXiv:1711.05217 (2017)
6. Gehrmann, S., Deng, Y., Rush, A.M.: Bottom-up abstractive summarization. arXiv preprint arXiv:1808.10792 (2018)
7. Gu, J., Lu, Z., Li, H., Li, V.O.: Incorporating copying mechanism in sequence-to-sequence learning. arXiv preprint arXiv:1603.06393 (2016)

8. Hariharan, S., Srimathi, R., Sivasubramanian, M., Pavithra, S.: Opinion mining and summarization of reviews in web forums. In: Proceedings of the third annual ACM Bangalore conference. p. 24. ACM (2010)
9. Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P.: Teaching machines to read and comprehend. In: Advances in neural information processing systems. pp. 1693–1701 (2015)
10. Kågebäck, M., Mogren, O., Tahmasebi, N., Dubhashi, D.: Extractive summarization using continuous vector space models. In: Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC). pp. 31–39 (2014)
11. Lin, C.Y., Hovy, E.: Automatic evaluation of summaries using n-gram cooccurrence statistics. In: Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (2003)
12. Liu, Y.: Fine-tune bert for extractive summarization. arXiv preprint arXiv:1903.10318 (2019)
13. Liu, Y., Li, S., Cao, Y., Lin, C.Y., Han, D., Yu, Y.: Understanding and summarizing answers in community-based question answering services. In: Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1. pp. 497–504. Association for Computational Linguistics (2008)
14. Mani, S., Catherine, R., Sinha, V.S., Dubey, A.: Ausum: approach for unsupervised bug report summarization. In: Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering. p. 11. ACM (2012)
15. Nallapati, R., Zhai, F., Zhou, B.: Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
16. Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al.: Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023 (2016)
17. Narayan, S., Cohen, S.B., Lapata, M.: Ranking sentences for extractive summarization with reinforcement learning. arXiv preprint arXiv:1802.08636 (2018)
18. Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304 (2017)
19. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015)
20. See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368 (2017)
21. Shi, Z., Melli, G., Wang, Y., Liu, Y., Gu, B., Kashani, M.M., Sarkar, A., Popowich, F.: Question answering summarization of multiple biomedical documents. In: Conference of the Canadian Society for Computational Studies of Intelligence. pp. 284–295. Springer (2007)
22. Yin, W., Pei, Y.: Optimizing sentence modeling and selection for document summarization. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
23. Zhang, H., Gong, Y., Yan, Y., Duan, N., Xu, J., Wang, J., Gong, M., Zhou, M.: Pretraining-based natural language generation for text summarization. arXiv preprint arXiv:1902.09243 (2019)
24. Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., Zhao, T.: Neural document summarization by jointly learning to score and select sentences. arXiv preprint arXiv:1807.02305 (2018)