Improving Abstractive Document Summarization with Salient Information Modeling

Yongjian You[1,2], Weijia Jia[2,1*], Tianyi Liu[1,2], Wenmian Yang[1,2]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]State Key Lab of IoT for Smart City, CIS, University of Macau, Macao, SAR China
{youyongjian, jiawj, liutianyi, sdq11111}@sjtu.edu.cn

Abstract

Comprehensive document encoding and salient information selection are two major difficulties for generating summaries with adequate salient information. To tackle the above difficulties, we propose a Transformer-based encoder-decoder framework with two novel extensions for abstractive document summarization. Specifically, (1) to encode the documents comprehensively, we design a focus-attention mechanism and incorporate it into the encoder. This mechanism models a Gaussian focal bias on attention scores to enhance the perception of local context, which contributes to producing salient and informative summaries. (2) To distinguish salient information precisely, we design an independent saliency-selection network which manages the information flow from encoder to decoder. This network effectively reduces the influences of secondary information on the generated summaries. Experimental results on the popular CNN/Daily Mail benchmark demonstrate that our model outperforms other state-of-the-art baselines on the ROUGE metrics.

1 Introduction

Document summarization is a fundamental task of natural language generation which condenses the given documents and generates fluent summaries with salient information automatically. Recent successes of neural sequence-to-sequence (seq2seq) models (Luong et al., 2015; Wu et al., 2016; Tu et al., 2016) enable the end-to-end framework for natural language generation, which inspires the research on abstractive summarization. Abstractive document summarization employs an end-to-end language model to encode a document into high-dimensional representations and then decode the representations into an abstractive summary. Though promising improvements have been achieved recently (Li et al., 2018c; Kryściński et al., 2018), there are still many problems are not studied well, such as the incompetence of salient information modeling.

Documents: a [duke student] has [admitted to hanging a noose made of rope] from a tree near a student union , [university officials] said thursday . the prestigious private school did n't identify the student , citing federal privacy laws . in a news release , it said the student was [no longer] on campus and [will face] student conduct [review] . the [student was identified during an investigation] by campus police and the office of student affairs and admitted to placing the noose on the tree early wednesday , the university said . ... at a forum held on the steps of duke chapel , close to where [the noose was discovered at 2 a.m]. , hundreds of people gathered . " you came here for the reason that you want to say with me , ' this is no duke we will accept . ...
Reference summary: student is no longer on duke university campus and will face disciplinary review . school officials identified student during investigation and the person admitted to hanging the noose , duke says . the noose , made of rope , was discovered on campus about 2 a.m.

Table 1: Example of a document and its corresponding reference summary. We consider the reference summary contains all salient information and mark the words or phrases appearing in the document in [red].

Modeling salient information contains the procedure of information representation and discrimination. Generally, the most essential prerequisite for a practical document summarization model is that the generated summaries should contain adequate salient information of the original documents. However, previous seq2seq models are still incapable of achieving convincing performance, which are restricted by the following two difficulties.

The first difficulty lies in the procedure of encoding. Considering a document is a long sequence of multiple sentences, the semantics of

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2132–2141
Florence, Italy, July 28 - August 2, 2019.

each token in document contain the dependencies with other distant tokens and its local context information. They both contribute to producing high-quality summaries with adequate salient information. The lack of long-term dependencies among tokens often leads to generating incomplete summaries (Li et al., 2018c). Unfortunately, traditional seq2seq encoders (recurrent or convolution based) are deficient in modeling dependencies among distant segments (Bengio et al., 1994; Li et al., 2018c). In recent years, the Transformer model (Vaswani et al., 2017) reveals remarkable performance in many similar tasks (Devlin et al., 2018) due to exploiting long-term dependencies, but recent studies point out this model may overlook local context occasionally (Yang et al., 2018). The absence of local context information accounts for inadequate details of salient information. Therefore, it is challenging to encode global information and local context comprehensively for each token in documents, which requires the capability of capturing long-term dependencies and local semantics at the same time. The second difficulty is to distinguish salient information from long documents precisely. In the example shown in Table 1, salient segments account for only a small part of the whole document, which is laborious for naive seq2seq models to distinguish important information from much secondary information. The summaries generated by these models usually lose salient information of original documents or even contain repetitions (Li et al., 2018c).

In this paper, we propose the **E**xtended **T**ransformer model for **A**bstractive **D**ocument **S**ummarization (ETADS) to tackle the above issues. Specifically, we design a novel focus-attention mechanism and saliency-selection network equipped in the encoder and decoder respectively: (1) To comprehensively encode the documents, we design a focus-attention mechanism, where a learnable Gaussian focal bias is employed as a regularization term on attention scores. This focal bias implicitly aggregates attention on local continuous scopes to emphasize the corresponding part of document. (2) To distinguish salient information in documents, we design an independent saliency-selection network to manage the information flow from encoder to decoder explicitly. The saliency-selection network employs a gate mechanism to assign a salient score for each token in source documents according to their encoded representations. We consider the lower-score tokens are relatively insignificant and reduce their likelihood of appearing in final summaries. Finally, we conduct extensive experiments on the CNN/Daily Mail dataset which is prevailing and widely used for document summarization task. The experimental results show that ETADS achieves state-of-the-art ROUGE scores and outperforms many strong baselines.

2 Related Work

With the development of seq2seq model on neural translation task, more and more researchers take note of its great potential in text summarization area (Fan et al., 2017; Ling and Rush, 2017; Cheng and Lapata, 2016), especially for abstractive methods. Rush et al. (2015) is the first to apply seq2seq model with attention mechanism to abstractive summarization and achieve promising improvement. Nallapati et al. (2016) modify the basic model with RNN-based encoder and decoder and propose several techniques. Chen et al. (2016) further propose to improve the novelty of generated summaries and design a distraction-based attentional model. Li et al. (2017) creatively incorporate the variational auto-encoder into the seq2seq model to learn the latent structure information. However, these models are nearly designed for abstractive sentence summarization, which focus on encoding and mining salient information on sentence-level and lead to unsatisfactory performances for document summarization.

Some recent work improves the performance of neural abstractive models on document summarization task from various aspects. To better grasp the essential meaning for summarization, Chen et al. (2016) propose not only to pay attention to specific regions and content of input documents with attention models, but also distract them to traverse between different content. Tan et al. (2017) propose a graph-based attention mechanism in a hierarchical encoder-decoder framework to generate multi-sentence summary. Gehrmann et al. (2018) presents a content selection model for summarization that identifies phrases within a document that are likely included in its summary. To produce more informative summaries, (Gu et al., 2016) is the first to show that the copy mechanism(Vinyals et al., 2015) can alleviate the Out-Of-Vocabulary problem by copying words from

the source documents. See et al. (2017) rebuild this pointer-generator network and incorporate an additional coverage mechanism into the decoder. Li et al. (2018b) notice the necessity of explicit information selection and they build a gated global information filter and local sentence selection mechanism. Moreover, reinforcement learning (RL) approaches have been shown to further improve performance on these tasks(Celikyilmaz et al., 2018; Li et al., 2018a). Pasunuru and Bansal (2018) develop a loss-function based on whether salient segments are included in a summary. However, the optimization of RL-based models can be difficult to tune and slow to train.

3 Model

In this section, we describe our approach from three aspects: (i) the Transformer-based encoder-decoder framework, (ii) the focus-attention mechanism for the encoder to emphasize the local context, and (iii) the saliency-selection network for the decoder to select salient information.

3.1 Encoder-Decoder Framework

Given a document $X = (x^1, x^2, ..., x^m)$, the encoder maps its corresponding symbol representations $E = (e^1, e^2, ..., e^m)$ to a sequence of continuous representations $Z = (z^1, z^2, ..., z^m)$, where m is the length of document. The decoder then decode Z into continuous representations $S = (s^1, s^2, ..., s^n)$ and generates abstractive summary $Y = (y^1, y^2, ..., y^n)$ one token a time, where n is the length of summary. $\mathcal{V}_s$ and $\mathcal{V}_t$ are the source/target vocabularies and $x^i \in \mathcal{V}_s$, $y^j \in \mathcal{V}_t$. E is the sum of word embedding representations and position embedding representations, where $e^i \in \mathbb{R}^{d_e}$. Both embedding representations are initialized as (Vaswani et al., 2017) and learned during the process of optimization.

3.1.1 Encoder

The encoder is composed of a stack of N identical layers, and each layer has two sub-layers. The first is the self-attention sub-layer and the second is the feed-forward sub-layer. The residual connection is employed around each of the two sub-layers, followed by layer normalization. Given the example input t, the output of each sub-layer can be formalized as $\mathrm{LayerNorm}(t + \mathrm{SubLayer}(t))$. For encoder, the $\mathrm{SubLayer}(t)$ can be replaced with $\mathrm{ATT}(t)$ or $\mathrm{FFN}(t)$, which represents the pre-output of self-attention sub-layer or feed-forward sub-layer respectively. The details of each sub-layer are presented as follows.

The self-attention sub-layer takes the output of previous layer as the input. Formally, the input for the self-attention sub-layer of the l-th layer is $Z_{l-1} \in \mathbb{R}^{m \times d_m}$, where d_m is the dimension of output. Specially, $Z_0 = E$ and the output of encoder $Z = Z_N$. In the process of computation, three matrices query $Q_l \in \mathbb{R}^{m \times d_m}$, key $K_l \in \mathbb{R}^{m \times d_m}$ and value $V_l \in \mathbb{R}^{m \times d_m}$ are obtained firstly by the linear projections from Z_{l-1} with three different metrics $W_l^Q \in \mathbb{R}^{d_m \times d_m}$, $W_l^K \in \mathbb{R}^{d_m \times d_m}$ and $W_l^V \in \mathbb{R}^{d_m \times d_m}$. Then the pre-output of self-attention sub-layer can be computed with the scaled dot-product attention mechanism:

$$\begin{aligned} \mathrm{ATT}(Z_{l-1}) &= att(Q_l, K_l, V_l) \\ &= \mathrm{softmax}(\frac{Q_l K_l^T}{\sqrt{d_m}}) V_l \end{aligned} \tag{1}$$

and the final output A_l of this sub-layer is obtained with residual connection and layer normalization. Moreover, the self-attention sub-layer can be further extended into multi-head manner. Namely,

$$\begin{aligned} \mathrm{ATT}^M(Z_{l-1}) &= concat(H_1, ..., H_h) W_l^C \\ \text{where } H_i &= att(Q_l W_{l,i}^Q, K_l W_{l,i}^K, V_l W_{l,i}^V) \end{aligned} \tag{2}$$

where h is the number of heads, $W_{l,i}^Q \in \mathbb{R}^{d_m \times d_h}$, $W_{l,i}^K \in \mathbb{R}^{d_m \times d_h}$, $W_{l,i}^V \in \mathbb{R}^{d_m \times d_h}$ and $W_l^C \in \mathbb{R}^{h * d_h \times d_m}$ are four learnable weight matrices, d_h is the dimension for each head, we set $d_h = d_m / h$.

The feed-forward sub-layer takes the output of self-attention sub-layer A_l as the input and the computation of pre-output $\mathrm{FFN}(A_l)$ is straightforward with a position-wise fully connected feed-forward network:

$$\mathrm{FFN}(A_l) = \mathrm{relu}(A_l W_l^1 + b_l^1) W_l^2 + b_l^2 \tag{3}$$

where $W_l^1 \in \mathbb{R}^{d_m \times d_f}$ and $W_l^2 \in \mathbb{R}^{d_f \times d_m}$ are two learnable weight matrices, d_f is the dimension of intermediate output. $b_l^1 \in \mathbb{R}^{d_f}$ and $b_l^2 \in \mathbb{R}^{d_m}$ are two learnable biases. The final output of feed-forward sub-layer Z_l is also the output for the l-th layer which is obtained after residual connection and layer normalization.

3.1.2 Decoder

The decoder in our framework has a similar stacked structure with N identical layers. In addition to the two sub-layers introduced above,

the decoder inserts another self-attention sub-layer in between, which performs multi-head attention over the output of the encoder. For clarity, we use the "bridge sub-layer" to refer to this additional self-attention sub-layer and $\mathrm{BATT}(Z, t)$ to represent the pre-output of this sub-layer, where Z is the encoder output and t is a example of encoded partial generated summary. The calculation of $\mathrm{BATT}(Z, t)$ is similar to the Eq.(1). Specifically, for the l-th bridge sub-layer in the decoder, key K_l and value V_l are obtained by linear projections from Z. Apart from the additional sub-layer, the rest of computation process is the same as the encoder, and the output of last layer H_N is considered as the final decoder output H.

Finally, for the i-th decoding step, we compute a distribution over the $\mathcal{V}_t$ for target elements y^i by projecting the output of decoder stack S^i via a linear layer with weights $W^o \in \mathbb{R}^{d_m \times T}$ and bias $b^o \in \mathbb{R}^T$,

$$p(y^i|y^1, ..., y^{i-1}; X) = \mathrm{softmax}(W^o S^i + b^o) \tag{4}$$

where T is the size of vocabulary $\mathcal{V}_t$.

3.2 Focus-Attention Mechanism

To take full advantage of documents information during the process of encoding, we design a focus-attention mechanism and build it in the self-attention sub-layers of the encoder, which is depicted as Figure 1. The "dotted boxes" indicate that the corresponding modules can be adapted into the multi-head manner.

The focus-attention mechanism models a focal bias as a regularization term on attention scores which is determined by the position of center and effective coverage scope. In the l-th self-attention sub-layer, since the query Q_l, key K_l and value V_l are obtained by linear projections from the input Z_{l-1}, so that they contain similar information in different semantic space. To reduce the amount of calculation, we only utilize the query matrices Q_l to compute the position vector and coverage scope vector. Specifically, for the i-th encoding step in l-th layer, the center position scalar $\mu_l^i \in \mathbb{R}$ and the coverage scope scalar $\sigma_l^i \in \mathbb{R}$ are calculated by two linear projections, namely:

$$\begin{aligned} \mu_l^i &= U_c^T \tanh(W_p Q_l^i + W_g G_l) \\ \sigma_l^i &= U_d^T \tanh(W_p Q_l^i + W_g G_l) \end{aligned} \tag{5}$$

where $W_p \in \mathbb{R}^{d_m \times d_m}$, and $W_g \in \mathbb{R}^{d_m \times d_m}$ are two shared weight matrices. $U_c \in \mathbb{R}^{d_m}$ and $U_d \in \mathbb{R}^{d_m}$

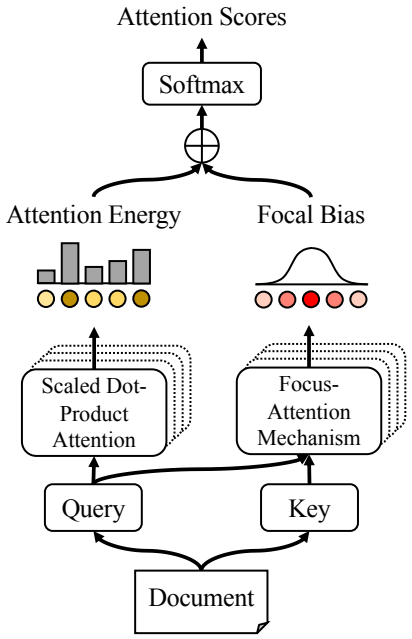

Figure 1: The focus-attention mechanism.

are two different linear projection weight vectors, m is the length of input document and $G_l = \frac{1}{m}\sum_{i=1}^{m} Q_l^i$ is the mean vector to provide complementary information. Furthermore, we regulate μ_l^i and σ_l^i to the closed interval $[0, m]$,

$$\begin{aligned} \tilde{\mu}_l^i &= m * \mathrm{sigmoid}(\mu_l^i) \\ \tilde{\sigma}_l^i &= m * \mathrm{sigmoid}(\sigma_l^i) \end{aligned} \tag{6}$$

According to the definition of Gaussian distribution, the focal bias for the i-th step $f_l^i \in \mathbb{R}^m$ can be easily obtained with $\tilde{\mu}_l^i$ and $\tilde{\sigma}_l^i$ as follows:

$$f_l^{i,j} = -\frac{(P^j - \tilde{\mu}_l^i)^2}{(\tilde{\sigma}_l^i)^2/2} \tag{7}$$

where P^j is the absolute position of word x^j in the document. $f_l^{i,j} \in [-\infty, 0]$ measures the distance between word x^j and the center position $\tilde{\mu}_l^i$.

Eventually, this focal bias is added to the attention energy of encoder layers before softmax normalization.

$$\begin{aligned} \mathrm{ATT}(Z_{l-1}) &= att(Q_l, K_l, V_l) \\ &= \mathrm{softmax}(\frac{Q_l K_l^T}{\sqrt{d}} \oplus f_l) V_l \end{aligned} \tag{8}$$

where $\oplus$ denotes the addition.

Moreover, we further adapt the focus-attention mechanism into the multi-head manner as Eq.2. Accordingly, the distinct focal biases are assigned for each head and different weight matrices are utilized in the process of computation.

3.3 Saliency-Selection Network

Abstractive document summarization is a special NLP generation task which requires to reduce the influence of secondary information and integrate salient segments to produce a condensed summary. Traditional seq2seq models often have limited performance on distinguishing salient segments (Tan et al., 2017), which emphasizes the necessity of customized selection network. In this work, we design the saliency-selection network for information selection, which is depicted as Figure 2.

Concretely, we measure the saliency of each word in the document by assigning a saliency score and make a soft selection. For the i-th decoding step in l-th layer, the saliency-selection network takes query matrices $Q_l^i \in \mathbb{R}_m^d$ and key matrices $K_l \in \mathbb{R}^{m \times d_m}$ as the input, where m is the length of the input document. Then, the network computes saliency score $g_l^i \in \mathbb{R}^m$ as:

$$g_l^{i,j} = \text{sigmoid}((W_h Q_l^i)(W_s K_l^j)^T) \quad (9)$$

where $W_h \in \mathbb{R}^{d_m \times d_m}$ and $W_s \in \mathbb{R}^{d_m \times d_m}$ are two learnable weight matrices. $g_l^{i,j} \in [0,1]$ measures the saliency of the j-th token in document for the i-th position in summary. Furthermore, we incorporate the computed saliency score g_l into the attention network of bridge sub-layer by:

$$\begin{aligned} \text{BATT}(Z, S^{l-1}) &= att(Q_l, K_l, V_l) \\ &= g_l \otimes \text{softmax}(\frac{Q_l K_l^T}{\sqrt{d}}) V_l \end{aligned} \quad (10)$$

where $\otimes$ denotes element-wise multiplication.

Moreover, we also adopt the saliency-selection network into the multi-head manner, which allows to model saliency from different perspectives at different positions.

3.4 Objective Function

Our goal is to maximize the output summary probability given the input document. Therefore, we optimize the negative log-likelihood loss function:

$$\mathcal{L} = -\frac{1}{|\tau|} \sum_{(X,Y) \in \tau} \log\ p(Y|X;\theta) \quad (11)$$

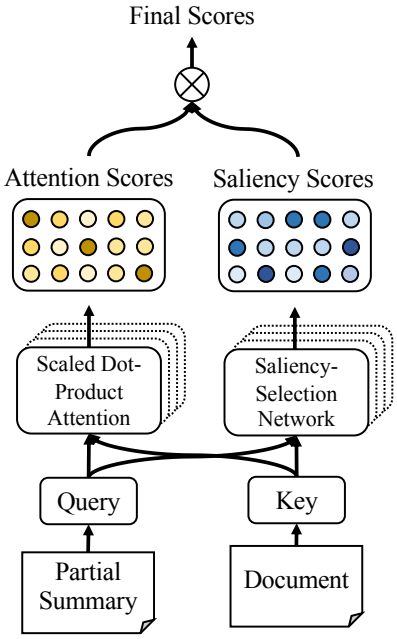

Figure 2: The saliency-selection netowrk.

where θ is the model parameter, and (X, Y) is a document-summary pair in training set τ, then

$$\log(Y|X;\theta) = \sum_{i=1}^{n} \log p(y^i|y^1, ..., y^{i-1}, X; \theta) \quad (12)$$

where $p(y^i|y^1, ..., y^{i-1}, X; \theta)$ is calculated by the decoder.

4 Experiments

In this section, we introduce the experiment setup, the implementation details, the baseline models and the experimental results.

4.1 Setup

We conduct the experiments on a large-scale corpus of CNN/Daliy Mail, which has been widely used for the explorations on document summarization. The corpus is originally constructed by collecting human generated highlights for new stories in CNN and Daily Mail website (Hermann et al., 2015). We use the scripts supplied by Nallapati et al. (2016) to further obtain the CNN/Daily Mail dataset. This dataset contains 287,226 training pairs, 13,368 validation pairs and 11,490 test pairs. We use the same non-anonymized version of dataset as See et al. (2017) which requires no pre-processing[1]. The average number of sentences

[1]https://github.com/abisee/cnn-dailymail

in documents and summaries are 42.1 and 3.8, respectively. We assume the length of all documents should not exceed 400 tokens and all summaries should not exceed 100 tokens. The word dictionary shared by documents and summaries contains 50,000 most popular tokens in documents.

In our model, we set the number of encoder/decoder layers $N = 4$ and the number of heads $h = 8$. The dimensions of the signal representation d_e and output d_m are set to 512, and the dimension of intermediate output d_f is set to 2048. Besides, the dropout rate is set to 0.8 in the process of training. We implement our model in PyTorch[2] 1.0. In all experiment, the batch size is set to 4096. We use the Adam optimizer (Kingma and Ba, 2014) to train our model with $\beta_1 = 0.9$, $\beta_2 = 0.998$ and $\epsilon = 10^{-8}$. The learning rate varies every step with the Noam decay strategy (Vaswani et al., 2017) and the warmup threshold is 8000. The maximum norm of gradient-clipping is set to 2. In the end, we conduct our experiment on one machine with 4 NVIDIA Titan Xp GPUs and the training process lasts 200,000 steps for each model.

We use the beam search algorithm (Sutskever et al., 2014) with coverage technique (Tu et al., 2016) to generate multiple summary candidates in parallel to obtain better results, the coverage weight is set to 1. For fear of favoring shorter generated summaries, we utilize length penalty (Wu et al., 2016) during the process of inference. We set the beam size to 10, the length penalty parameter α to 0.9 and β to 5. The minimum length of the generated summary is set to 35 and the batch size for inference is set to 1.

Following the previous studies, we use the ROUGE scores (Lin and Hovy, 2003) to evaluate the performance of our model with Python implementation[3] and standard options. ROUGE scores measure the quality of summary by computing overlapping lexical units with references, such as uni-gram, bi-gram and longest common subsequence (LCS). F-measures ROUGE-1 (unigram), ROUGE-2 (bi-gram) and ROUGE-L (LCS) are reported as the evaluation metrics.

4.2 Baselines

In this work, we compare our approach with these following state-of-the-art baselines:

[2]https://pytorch.org/

[3]https://github.com/falcondai/pyrouge

Models	RG-1	RG-2	RG-L
words-1vt2k-temp-att	36.64	15.66	33.42
PG+cov	39.53	17.28	36.38
ConvS2S	39.75	17.29	36.54
Explicit-Selection	41.54	18.18	36.47
ROUGEEsal+Ent	40.43	18.00	37.10
Bottom-Up	41.22	18.68	38.34
Basic model	39.45	17.20	36.49
+Focus-Attention	40.29	18.63	38.11
+Saliency-Selection	40.76	18.40	37.67
ETADS	**41.75**	**19.01**	**38.89**

Table 2: ROUGE scores on the CNN/Daliy Mail test set. All ROUGE scores have **95% confidence interval of at most ±0.24** computed by the official ROUGE script. To save space, we use "PG+cov" and "Bottom-Up" to denote the baseline "Pointer-Generator+coverage" and "Bottom-Up Summarization". The symbol "+" stands for the corresponding module is added on the "Basic model" which is a vanilla Transformer with 4 identical layers.

words-1vt2k-temp-att: Nallapati et al. (2016) build this model with the basic seq2seq encoder-decoder architecture and attention mechanism, which is a pioneering effort for much other work.

Pointer-generator+coverage: To deal with Out-Of-Vocabulary words (OOV words) and repeating problem, See et al. (2017) combine the pointer network into the RNN-based seq2seq model and design a coverage mechanism.

ConvS2S: Gehring et al. (2017) creatively utilize convolution neural networks to build seq2seq model and achieve high performance on many tasks, including abstractive summarization.

Explicit-Selection: Li et al. (2018b) propose to extend the basic seq2seq model with an information selection layer to explicitly control information flow.

ROUGESal+Ent(RL): Pasunuru and Bansal (2018) address main difficulties via a reinforcement learning approach with two novel reward functions.

Bottom-Up Summarization: This work combines extractive and abstractive summarization by firstly using a data-efficient content selector to over-determine phrase related (Gehrmann et al., 2018).

4.3 Results

The experimental results are given in Table 2. Overall, **ETADS** achieves advantages of ROUGE

F1 scores over all of the other baselines (reported in their own articles) and two extensions we proposed both improve the performances based on the basic model. Concretely, we design the focus-attention mechanism to improve the capability of capturing the local context information and further encode the document comprehensively. Therefore, **the basic model with focus-attention mechanism** is expected to achieve improvement in producing summaries with continuous salient segments. The significant improvement on ROUGE-L verifies our hypothesis. Besides, we notice that the improvements provided by **the basic model with saliency-selection network** particularly lie in ROUGE-1 F1 scores. We consider the reason may lie in the saliency-selection network is more sensitive to the short segments due to the separate saliency measuring process.

Comparing with the two classical RNN-based baselines **words-1vt2k-temp-att** and **Pointer-generator+coverage** and one CNN-based baseline **ConvS2S**, our basic model is capable of achieving equivalent performance. We believe it should give credit to the capability of modeling long-term dependencies. When compared with more recent work, **Explicit-Selection** equips a selection layer similar to our saliency-selection network to mine salient information. Despite being aware of this problem, our saliency-selection network achieves better performance with the help of stacked architecture. The performance of reinforcement learning based model **ROUGEEsal+Ent** is worse than our model obviously. The strongest baseline **Bottom-Up Summarization** combines the advantages of CNN-based model and RNN-based model but is also slightly inferior to our model.

4.4 Case Study

To further illustrate the effectiveness of our proposed ETADS vividly and analyze the reasons of improving the performance, we compare the generated summaries by baselines **words-1vt2k-temp-att**, **Bottom-Up Summarization** and our **ETADS** approach. For the case in Table 3, the input document focuses on analyzing the latest financial report of the Apple company and further discusses the impact of the new Apple Watch on retail revenue. The performance of **words-1vt2k-temp-att** is unsatisfactory, three generated sentences are irrelevant to the main concepts and

Reference summary: apple sold more than 61 million iphones in the quarter . apple did n't report any results for the new apple watch . believed around 2 million watches have been sold , according to estimates .
words-1vt2k-temp-att: the iphone is still the engine behind apple 's phenomenal success . apple has vied with south korea 's samsung for the no. 1 position in the global smartphone market . apple ceo tim cook has said he 's optimistic about new markets such as [china china china china china ...]
Bottom-Up Summarization: [apple sold more than 61 million iphones in the quarter] , accounting for more than two-thirds of its $ 58 billion in revenue for the quarter and the lion 's share of $ 13.6 billion in profit - and up 40 % from a year ago . $ 200 billion in cash , up from around $ 150 billion for one year . revenue from mac computers rose 2 % to $ 5.6 billion .
ETADS: [apple sold more than 61 million iphones in the quarter .] it was a 40 percent increase over the number of iphones sold in the first three months of 2014 . [apple did n't report any results for the new apple watch] , which it began selling this month , after the quarter ended .

Table 3: Example of generated summaries. We highlight the words or sentences in [red] which are consistent with partial reference summary. Repetition segments are marked in [blue] .

even contains repetitions at the end of the summary. Abstractive summary generated by baseline **Bottom-Up Summarization** is much more better, which indicates the effectiveness of modifications. However, the generated summary only contains partial salient information of the document. **ETADS** achieves the best performance in this case due to two of the generated sentences containing salient information and without repetitions. The above results verify that the extensions in our model improve the capability of document summarization from not only quantitative but also qualitative perspectives.

4.5 Discussion

In this section, we first validate the robustness of our model with different encoder/decoder architectures and then discuss the different deploy strategies for our extensions.

4.5.1 Architecture Robustness

We conduct experiments to see how the model's performance is affected by the stacked architecture. We perform a set of experiments which adjust the structures of the encoder and decoder to

Encoder	RG-1	RG-2	RG-L	# of paras
2 layers	35.12	14.05	32.41	3190K *
4 layers	39.45	17.20	**36.49**	3821K
6 layers	**39.67**	**17.47**	35.71	4451K
Decoder	**RG-1**	**RG-2**	**RG-L**	**# of paras**
2 layers	31.10	12.93	27.04	3406K
4 layers	**39.45**	17.20	**36.49**	4246K
6 layers	39.35	**18.01**	36. 21	5087K

* 1K equals to 1000

Table 4: ROUGE scores on the CNN/Daily Mail test set. "# of paras" denotes the number of training parameters. We fix the decoder to 4 layers when adjust structure of the encoder and vice versa.

Layers	RG-1	RG-2	RG-L
-	40.87	17.78	37.73
[1-2]	42.81	20.12	39.68
[3-4]	41.91	19.65	39.32
[1-4]	**43.06**	**20.85**	**40.12**

Table 5: ROUGE precision scores on the CNN/Daliy Mail test set. We use the token "-" to indicate the basic model which does not contain saliency-selection network. "[1-2]" indicates we deploy saliency-selection network on the first and second layer of basic model, "[3-4]" and "[1-4]" are similar.

2, 4 and 6 layers respectively. Experimental results on the test set in Table 4 show that there is no notable difference between 4 layers or 6 layers for encoder or decoder. However, the number of parameters is significantly increased nearly 1/4 for 6 layers, which means more time is needed for convergence. Employing 2 layers for either the encoder or decoder leads to rapid performance degradation. From the aspect of efficiency and effectiveness, we decide to equip 4 layers for the encoder and decoder eventually.

4.5.2 Deployment Strategies

In this section, we discuss the different deployment strategies for our extensions on the encoder-decoder framework.

Firstly, we deploy the saliency-selection network on different layers to discuss strategies of saliency-selection deployment. As we mentioned before, the major difficulty of this salient information selection procedure is to comprehend the relative semantic meanings and make the correct selection, which significantly affects the precision scores. Therefore, it is proper to use precision

Layers	RG-1	RG-2	RG-L
-	41.10	17.82	37.91
[1-2]	40.92	18.61	38.22
[3-4]	40.57	18.20	38.19
[1-4]	**41.31**	**18.72**	**38.93**

Table 6: ROUGE recall scores on the CNN/Daliy Mail test set. "-" to indicate the basic model which does not contain focus-attention mechanism. Other symbols express same meaning with Table 5

scores to measure effectiveness. From Table 5, it can be observed that the improvements brought by our saliency-selection network do not increase with layers linearly. In the shallow layers, the saliency-selection network contributes to notable improvement which is close to the best results we achieved. However, for the deeper layers, the improvement brought by the saliency-selection network is limited. We believe it can be attributed to the characteristics of our encoder-decoder framework. Self-attention sub-layer effectively reduces the cost of long-term information fusion, which leads to difficult to comprehend the original semantic information. The saliency-selection network we proposed is not competent to distinguish noise information when the original semantic information becoming confusing.

Furthermore, we discuss the strategies for focus-attention mechanism with ROUGE recall scores. The results of Table 6 demonstrate a similar phenomenon to Table 5 where improvements mainly come from shallow layers. We believe it is a trade-off between local context and global information. Focus-attention mechanism aims to gather attention to the local context around a center which deviates from the original goal. (Vaswani et al., 2017; Shi et al., 2016) indicate that there exists a consensus in the NLP community that shallow layers of a stacked model are sensitive to local context and deeper layers modeling global semantics. Therefore, as the module designed to capture local context, we believe it is reasonable to obtain more promotion where it is equipped on shallower layers which is also a side proof of effectiveness.

5 Conclusion

In this paper, we propose a novel framework for abstractive document summarization with extended Transformer model. The proposed model consists of a concise pipeline. First, the stacked

encoder with focus-attention mechanism captures long-term dependencies and local context of input document comprehensively. Then the decoder with saliency-selection network distinguishes and condenses the salient information into the output. Finally, an inference algorithm produces the abstractive summaries. Our experiments show that the proposed model achieves a significant improvement for abstractive document summarization over previous state-of-the-art baselines.

Acknowledgments

This work is supported by Chinese National Research Fund (NSFC) Key Project No. 61532013 and No. 61872239. FDCT/0007/2018/A1, DCT-MoST Joint-project No. (025/2015/AMJ), University of Macau Grant Nos: MYRG2018-00237-RTO, CPG2018-00032-FST and SRG2018-00111-FST of SAR Macau, China.

References

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1662–1675.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for document summarization. *arXiv preprint arXiv:1610.08462*.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 484–494.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Angela Fan, David Grangier, and Michael Auli. 2017. Controllable abstractive summarization. *arXiv preprint arXiv:1711.05217*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1631–1640.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817.

Piji Li, Lidong Bing, and Wai Lam. 2018a. Actor-critic based training framework for abstractive summarization. *arXiv preprint arXiv:1803.11070*.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100.

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018b. Improving neural abstractive document summarization with explicit information selection modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796.

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018c. Improving neural abstractive document summarization with structural regularization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4078–4087.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Jeffrey Ling and Alexander Rush. 2017. Coarse-to-fine attention models for document summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 33–42.

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

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

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 646–653.

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

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1073–1083.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural mt learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1171–1181.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 76–85.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

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

Baosong Yang, Zhaopeng Tu, Derek F Wong, Fandong Meng, Lidia S Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4449–4458.