Attentive Encoder-based Extractive Text Summarization

Chong Feng
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
fengchong16@nudt.edu.cn

Fei Cai*
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
caifei@nudt.edu.cn

Honghui Chen
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
chenhonghui@nudt.edu.cn

Maarten de Rijke
Informatics Institute
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

In previous work on text summarization, encoder-decoder architectures and attention mechanisms have both been widely used. Attention-based encoder-decoder approaches typically focus on taking the sentences preceding a given sentence in a document into account for document representation, failing to capture the relationships between a sentence and sentences that follow it in a document in the encoder. We propose an attentive encoder-based summarization (AES) model to generate article summaries. AES can generate a rich document representation by considering both the global information of a document and the relationships of sentences in the document. A unidirectional recurrent neural network (RNN) and a bidirectional RNN are considered to construct the encoders, giving rise to unidirectional attentive encoder-based summarization (Uni-AES) and bidirectional attentive encoder-based summarization (Bi-AES), respectively. Our experimental results show that Bi-AES outperforms Uni-AES. We obtain substantial improvements over a relevant start-of-the-art baseline.

CCS CONCEPTS

• **Information systems** → **Summarization**;

KEYWORDS

Summarization; Attention mechanism; Encoder-decoder

ACM Reference Format:
Chong Feng, Fei Cai, Honghui Chen, and Maarten de Rijke. 2018. Attentive Encoder-based Extractive Text Summarization. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269251

*Corresponding author.

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3269251

1 INTRODUCTION

Text summarization is meant to quickly locate the key sentences of an article, which is generally done by so-called extractive and abstractive models. Extractive summarization models, which aim to generate a short text summary by extracting the salient sentences in a document [10], are able to generate more fluent summaries than abstractive approaches. In the field of extractive text summarization, encoder-decoder architectures have been used successfully [2]. Attention mechanisms in an encoder-decoder architecture allows it to automatically focus on the key region of a document [1]. Most encoder-decoder-based methods implement an attention mechanism in the decoder rather than in the encoder [2, 9], where they mainly focus on selecting the sentences that are relevant to the source document but may neglect the relationships between sentences. However, relationships of sentences may indicate whether sentences express the same meaning, which may help to make the decision of selecting them into the summary.

We propose an attentive encoder-based summarization (AES) model for text summarization. It consists of an attention-based document encoder and an attention-based sentence extractor. Specifically, we first apply a unidirectional recurrent neural network (RNN) to construct the encoder, which leads to a unidirectional attentive encoder based summarization model (Uni-AES). Furthermore, as a bidirectional RNN can read a sequence not only in the original order but also in the reverse order, we apply bidirectional RNN in the document encoder to construct a bidirectional attentive encoder-based summarization model (Bi-AES), which helps to obtain better document representation. We evaluate the performance of our models on a public dataset consisting of the CNN news articles. Our experimental results show that Bi-AES outperforms Uni-AES in terms of the ROUGE scores. In particular, Bi-AES presents a significant improvement over a relevant start-of-the-art baseline.

Our contributions are: (1) We propose an attentive encoder-based neural model to generate a rich document representation that helps to select the correct sentences into an article summary. (2) We compare the effectiveness of a unidirectional RNN and a bidirectional RNN in our model and find that document encoder composed of a bidirectional RNN produces a better performance for text summarization. (3) We investigate the performance of our models on various documents lengths and for various summary lengths, and find that Bi-AES can generate better summaries for short documents and is more effective when generating longer summaries.

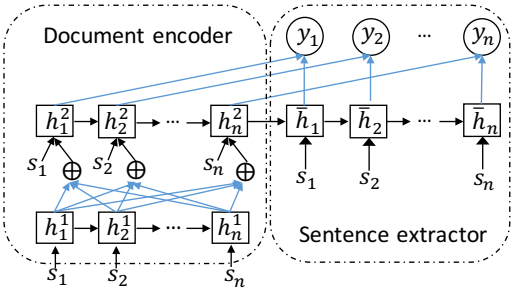


Figure 1: Structure of Uni-AES.

2 APPROACH

For extractive summarization, given a document d consisting of a sequence of n sentences $s_1, s_2, \ldots, s_n$, we aim to generate a summary for d by selecting a subset of m ($m < n$) sentences. To achieve this, we score each sentence s_t ($1 \leq t \leq n$) and predict its label $\hat{y}_t \in \{0, 1\}$, where 1 indicates that s_t should be selected into the summary and 0 indicates that it should not. In a supervised training setup, we aim to maximize the likelihood of all predicted sentence labels $\hat{y} = \langle \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n \rangle$ given the input document d and the model parameters θ:

$$p(\hat{y}|d;\theta) = \prod_{t=1}^{n} p(\hat{y}_t|d;\theta). \tag{1}$$

We propose an attentive encoder-based summarization (AES) model, where a unidirectional RNN and a bidirectional RNN are incorporated, respectively, resulting in Uni-AES and Bi-AES. The main structure of Uni-AES is illustrated in Fig. 1: an attentive RNN-based document encoder and an attention-based sentence extractor. The key difference between Uni-AES and Bi-AES lies in the part of document encoder. First of all, for Uni-AES, as shown in Fig. 1, the first layer of the document encoder reads the sentences and produces a hidden state at each time step, which is then used to calculate the relationships between sentences using an attention mechanism. The second layer reads both the sentences and the relationships between sentences to generate the document representation. After that, the final layer of the sentence extractor can predict the labels for each sentence by merging the final hidden states returned by the document encoder and the corresponding extractor hidden states. For Bi-AES, as shown in Fig. 2, a bidirectional RNN is incorporated in the document encoder.

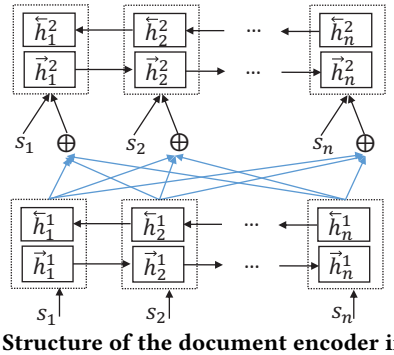


Figure 2: Structure of the document encoder in Bi-AES.

2.1 Attention-based document encoder

The document encoder in our model has two layers of recurrent neural networks (RNNs). In the first layer, a kind of self-attentive structure [7] is applied to extract different aspects of a document into a vector representation. In the second layer, each sentence is concatenated with the document representation returned by the first layer to avoid information loss and to generate the final document representation. In particular, in Uni-AES, given a document consisting of a sequence of n sentences indicated as $d = \langle s_1, s_2, \ldots, s_n \rangle$, the hidden state in the first layer of the document encoder at the time of inputting the t-th sentence, i.e., the time step t, is updated as:

$$h_t^1 = RNN(s_t, h_{t-1}^1), \tag{2}$$

where s_t is the sentence embedding calculated following [3]. The initial hidden state h_0^1 is set to a zero vector. We represent a sequence of all hidden states h_t^1 ($1 \leq t \leq n$) as H:

$$H = \langle h_1^1, h_2^1, \ldots, h_n^1 \rangle. \tag{3}$$

So far, the hidden state h_t^1 is produced only based on the the sentences preceding s_t. To further capture the relationships between sentences, we design an attention mechanism to process the outputs of the first encoder layer. Specifically, we sum the hidden states with different weights assigned to sentences:

$$\widetilde{h}_t = \sum_{j=1}^{n} a_j^t h_j^1, \tag{4}$$

where a_j^t is a normalized attention weight when inputting the t-th sentence at the j-th hidden state that is calculated as

$$a_j^t = \frac{\exp(e_j^t)}{\sum_{i=1}^{n} \exp(e_i^t)}, \tag{5}$$

where e_j^t is the initial attention weight computed based on the hidden states H:

$$e_j^t = v_j^t \tanh(WH^T), \tag{6}$$

where v_j^t and W are the trainable model parameters. Each $\widetilde{h}_t$ captures the relationships between s_t and other sentences in the document. We then input $\widetilde{h}_t$ to the second encoder layer.

To avoid information loss, we concatenate the sentence embedding s_t with $\widetilde{h}_t$, the document representation. Thus, the hidden state of the second layer for sentence s_t can be updated as:

$$h_t^2 = RNN([s_t, \widetilde{h}_t], h_{t-1}^2), \tag{7}$$

where $[s_t, \widetilde{h}_t]$ means the concatenation of s_t and $\widetilde{h}_t$.

In bidirectional RNNs, the sentences are input following the original order in the document as well as following the reverse order. Hence, bidirectional RNNs can capture the relationships between a sentence and its surrounding sentences [1]. Thus, we apply a bidirectional RNN in the document encoder to better capture the relationships between sentences, resulting in the Bi-AES model. Corresponding to h_t^1 in (2), a bidirectional RNN produces two hidden states in the first layer at time step t:

$$\overrightarrow{h_t^1} = \overrightarrow{RNN}(s_t, \overrightarrow{h_{t-1}^1}) \text{ and } \overleftarrow{h_t^1} = \overleftarrow{RNN}(s_t, \overleftarrow{h_{t+1}^1}), \tag{8}$$

where $\overrightarrow{h_t^1}$ and $\overleftarrow{h_t^1}$ are the forward and backward hidden state, respectively. The initial states $\overrightarrow{h_0^1}$ and $\overleftarrow{h_{n+1}^1}$ are set to zero vectors. We concatenate $\overrightarrow{h_t^1}$ with $\overleftarrow{h_t^1}$ to produce the corresponding hidden state h_t^1 as in (2), so that we can get $\widetilde{h}_t$ as in (4). Similarly, for generating h_t^2, Bi-AES first computes $\overrightarrow{h_t^2}$ and $\overleftarrow{h_t^2}$ as

$$\overrightarrow{h_t^2} = \overrightarrow{RNN}([s_t, \widetilde{h}_t], \overrightarrow{h_{t-1}^2}) \text{ and } \overleftarrow{h_t^2} = \overleftarrow{RNN}([s_t, \widetilde{h}_t], \overleftarrow{h_{t+1}^2}), \tag{9}$$

and then concatenate $\overrightarrow{h_t^2}$ with $\overleftarrow{h_t^2}$ to get h_t^2.

2.2 Attention-based sentence extractor

The sentence extractor in the AES model is an RNN-based estimation that computes the salience of each sentence to be labeled. In particular, an attention mechanism is applied in the sentence extractor. Given a document d and the encoder hidden states $\{h_1^2, h_2^2, \ldots, h_n^2\}$, our extractor predicts the probability of selecting sentence s_t into the summary at time step t by considering its encoder hidden state h_t^2 and its corresponding extractor hidden state $\overline{h}_t$ as:

$$p(y_t|s_t, d) = \text{softmax}(\sigma(h_t^2, \overline{h}_t)), \tag{10}$$

where $\sigma(h_t^2, \overline{h}_t)$ is a multi-layer neural network produced by

$$\sigma(h_t^2, \overline{h}_t) = V \tanh(U_1 h_t^2 + U_2, \overline{h}_t), \tag{11}$$

where U_1, U_2 and V are the trainable neural network parameters. The extractor hidden state $\overline{h}_t$ is produced by

$$\overline{h}_t = RNN(s_t, \overline{h}_{t-1}), \tag{12}$$

while inputting s_t and $\overline{h}_0$ is equal to the last hidden state of the document encoder, i.e., h_n^2. In addition, a negative likelihood based loss function is applied when computing $p(y_t|s_t, d)$ in (10)

$$loss = -\frac{1}{n} \prod_{t=1}^{n} p(y_t = \overline{y}_t|s_t, d). \tag{13}$$

Finally, the label of sentence s_t indicating whether s_t should be selected into the summary can be predicted by

$$\hat{y}_t = \arg\max_{y_t \in \{0,1\}} p(y_t|s_t, d). \tag{14}$$

3 EXPERIMENTS

Research questions. (**RQ1**) How does the performance of our proposal, attentive encoder-based summarization (AES) (Uni-AES and Bi-AES), compare to start-of-the-art baselines? (**RQ2**) How do our models perform on different length of documents? (**RQ3**) What is the impact on summarization performance of our models when varying the length of the generated summary, i.e., 75 bytes vs. 275 bytes vs. full length (three sentences)?

Model summary. As our AES models generate a summary by selecting the salient sentences from an original document, we compare our models with baselines for extractive text summarization: (1) **LEAD**: a standard text summarization baseline selecting the *leading* three sentences from each document as the summary [2, 8, 10]; (2) **NN-SE**: the state-of-the-art neural extractive text summarization model composed of a hierarchical document encoder and an attention based sentence extractor [2]. Other encoder-decoder based models [5, 9] are not included for comparisons as they are mainly used for abstractive summarization. The models we propose in this paper are: (1) **Uni-AES**, an attentive encoder-based model with a unidirectional RNN; and (2) **Bi-AES**, an attentive encoder-based model with a bidirectional RNN.

Datasets and experimental setup. We follow the experimental setup of [2]. We train and evaluate our summarization models on a publicly available dataset created from the CNN news [2]. Each document in the dataset contains its highlights which are genuinely created by the news editors, so we can use the highlights as the ground truth summary. Every sentence of a document has been labeled with 1 (selected into the summary) or 0 (otherwise). Details of the dataset are shown in Table 1. All sentences are padded to a fixed length of 50 words and documents are padded to a fixed length of 60 sentences because more than 95% sentences have 50 words or less and more than 95% documents contain 60 sentences or less. For the recurrent document encoder and sentence extractor, we use LSTM cells with a size of 650 and regularization dropout with probability 0.5 on the scoring layer as well as the LSTM input-to-hidden layers. We train our models with a batch size of 20 documents and the Adam optimizer [4] with initial learning rate 0.001 to achieve a global optimum. These parameters are regulated by promoting the performance of the validation dataset after each epoch and determined after repeated validation.

Table 1: Dataset statistics [2].

Variables	Train	Test
# Documents	83,568	1,093
Maximal # words in a sentence	1,341	1,426
Maximal # sentences in a document	125	105
Average # words per sentence	23.6	23.1
Average # sentences per document	29.8	30.2
Average # highlights per document	3.5	2.6

Table 2: ROUGE evaluations (%) on the CNN dataset. The results produced by the best baseline and the best performer in each column are underlined and boldfaced, respectively. Statistical significance of differences between Uni-AES or Bi-AES and the best baseline ($^{\blacktriangle}$) and between Bi-AES and Uni-AES ($_{\triangle}$) is determined by a t-test ($p < 0.05$).

Models	R-1	R-2	R-3	R-4	R-L
LEAD	43.50	13.30	8.09	5.32	<u>40.13</u>
NN-SE	<u>44.53</u>	<u>13.68</u>	<u>8.21</u>	<u>5.62</u>	39.37
Uni-AES	44.81	13.87	8.26	5.71	39.64
Bi-AES	**47.83**$^{\blacktriangle}_{\triangle}$	**16.92**$^{\blacktriangle}_{\triangle}$	**9.28**$^{\blacktriangle}$	**5.98**	**42.36**$^{\blacktriangle}_{\triangle}$

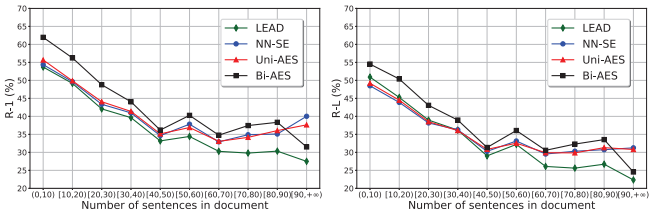
Evaluation metric. We evaluate the performance of the models using the ROUGE metrics, which count the number of over-lapping units such as n-grams, word sequences, and word pairs between the machine-generated summary to be evaluated and the ideal summaries [6]; ROUGE-n (R-n, for $n = 1, \ldots 4$) indicate the informativeness of a summary, and ROUGE-L (R-L) is based on the longest common subsequence to capture its fluency.

4 RESULTS AND DISCUSSION

4.1 Performance of summarization models

For **RQ1**, we examine the ROUGE scores of full length summaries produced by our models as well as the baselines. The results are shown in Table 2. In the baseline group, NN-SE has a higher R-N ($N = 1, 2, 3, 4$) score than LEAD but a lower R-L score: NN-SE can generate more informative but less fluent summaries than LEAD.

Uni-AES shows a slight improvement over the baselines in terms of R-N ($N = 1, 2, 3, 4$), presenting an improvement of around 1% against NN-SE, but it loses against LEAD in terms of R-L. Regarding Bi-AES, the improvements over NN-SE are obvious. In particular, compared to NN-SE, Bi-AES presents an improvement of 7.41%, 23.68%, 13.03%, 6.41% and 7.59% in terms of R-1, R-2, R-3, R-4 and R-L, respectively. These improvements are mostly significant. These results confirm the effectiveness of Bi-AES. Our attention mechanism combined with a bidirectional RNN helps to obtain the main gists of articles and select salient sentences for text summarization.



(a) Performance in terms of R-1. (b) Performance in terms of R-L.

Figure 3: Performance for different document lengths.

In addition, we compare the performance of Bi-AES and Uni-AES. Bi-AES receives higher ROUGE scores than Uni-AES, with an improvement of 6.74%, 21.99%,12.35%, 4.73% and 6.86% in terms of R-1, R-2, R-3, R-4 and R-L, respectively. We see a significant improvement of Bi-AES over Uni-AES in terms of R-1, R-2 and R-L. Thus, the use of *bidirectional* RNNs is beneficial to capturing sentence relationships by considering the surrounding sentences.

4.2 Impact of document length

We move to **RQ2** and group the test documents by length, i.e., the number of sentences they contain. For simplicity, we examine the performance of discussed models in terms of R-1 and R-L for generating the full-length summaries. We plot the results in Fig. 3. The overall R-1 and R-L scores of the four summarization models go down as the length of documents increases. This could be explained by the fact that for long documents, the key sentences may be more dispersed, which increases the difficulty for text summarization. In addition, the Bi-AES model may be more sensitive to document length than other models as the ROUGE scores present an obvious change when the number of sentences varies. The R-1 and R-L scores of the Uni-AES model are close to those of the NN-SE model in different intervals, but the improvements of the Bi-AES model over the baselines are obvious. Generally, for short documents, e.g., fewer than 50 sentences, the relative improvements of our proposals over the baselines present a monotonous decrease. In addition, the improvements of Bi-AES over the baselines are more remarkable when the number of sentences is less than 40 than the improvements observed when the number of sentences exceeds 40. The reason may be that less than 5% documents contain 50 sentences or more may make the training parameters not adequately optimized for long documents. For the Bi-AES model, as bidirectional RNNs capture more information, they may also pick up more noise or redundant information when incorporated in the AES model.

4.3 Impact of the length of summaries

Finally, to answer **RQ3**, we compare the summarization results when summaries of varying length: 75 bytes vs. 275 bytes vs. full length (three sentences). The results for full-length summaries have been reported in Table 2; we show the ROUGE scores for 75 and 275 bytes in Table 3. Again, the best results are produced by our Uni-AES and Bi-AES, and in general NN-SE is superior to LEAD. For generating the 75 bytes summaries, Uni-AES and Bi-AES achieve a slight improvement over the baselines in terms of all ROUGE scores. The differences in performance between Uni-AES and Bi-AES are relatively small. Interestingly, for generating the 275 bytes summaries, Uni-AES and especially Bi-AES achieve a clear improvement over the baselines. For instance, in terms of R-1, Bi-AES achieves a significant improvement of near 9.26% over NN-SE; in terms of R-L, an improvement of nearly 4.57% over LEAD is achieved. When comparing Bi-AES and Uni-AES, we observe significant improvements in terms of R-1 and R-L. In summary, the proposed AES model works better for generating long summaries than short ones, especially when it is incorporated with a bidirectional RNN.

Table 3: ROUGE evaluations (%) on the CNN dataset with various length limits, i.e., 75 and 275 bytes. The best baseline and the best performer in each column are underlined and boldfaced, respectively. Statistical significance of pairwise differences between Uni-AES or Bi-AES and the best baseline (▲) and between Bi-AES and Uni-AES (△) is determined by a t-test at the level of $p < 0.05$.

Models		R-1	R-2	R-3	R-4	R-L
(*75 bytes*)	LEAD	12.24	2.67	1.13	0.57	11.19
	NN-SE	12.49	2.88	1.34	0.78	11.23
	Uni-AES	12.81	3.00	**1.40**	**0.79**	11.39
	Bi-AES	**13.13**	**3.01**	1.39	**0.79**	**11.65**
(*275 bytes*)	LEAD	34.63	11.02	5.15	3.23	32.37
	NN-SE	35.51	11.24	5.30	3.31	31.10
	Uni-AES	35.96	11.43	5.30	3.35	31.34
	Bi-AES	**38.80**▲△	**12.61**	**6.68**▲	**4.14**	**33.85**△

5 CONCLUSIONS AND FUTURE WORK

We have proposed AES, an attentive encoder-based summarization model for extractive text summarization, where a unidirectional recurrent neural network (RNN) and a bidirectional RNN are considered in the document encoder, respectively. The bidirectional model Bi-AES clearly outperforms the baselines as well as the unidirectional model Uni-AES in terms of ROUGE scores, especially on short documents and when generating long summaries.

As to future work, we would like to test our models on different datasets to examine if they are suitable for text summarization for other genres. We also intend to incorporate other features of a document, e.g., topic, title and relevance between paragraphs, to capture richer information and generate better summaries.

Acknowledgments. This research was supported by the National Natural Science Foundation of China under No. 61702526, the Defense Industrial Technology Development Program under No. JCKY2017204B064, and the National Advanced Research Project under No. 6141B0801010b.

REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.
[2] J. Cheng and M. Lapata. 2016. Neural summarization by extracting sentences and words. In *ACL*. 484–494.
[3] Y. Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*. 1746–1751.
[4] D.P. Kingma and J. Ba. 2015. A method for stochastic optimization. In *ICLR*.
[5] P. Li, W. Lam, L. Bing, and Z. Wang. 2017. Deep recurrent generative decoder for abstractive text summarization. In *EMNLP*. 2091–2100.
[6] C.-Y. Lin and E. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *HLT-NAACL*. 71–78.
[7] Z. Lin, M. Feng, C.N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. 2017. A structured self-attentive sentence embedding. In *ICLR*.
[8] R. Nallapati, F. Zhai, and B. Zhou. 2017. SummaRuNNer: A recurrent neural Network based sequence model for extractive summarization of documents. In *AAAI*. 3075–3081.
[9] R. Nallapati, B. Zhou, C.N. dos Santos, Ç. Gülçehre, and B. Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *CoNLL*. 280–290.
[10] P. Ren, Z. Chen, Z. Ren, F. Wei, J. Ma, and M. de Rijke. 2017. Leveraging contextual sentence relations for extractive summarization using a neural attention model. In *SIGIR*. 95–104.