Noname manuscript No.
(will be inserted by the editor)

Incorporating Word Attention with Convolutional Neural Networks for Abstractive Summarization

Chengzhe Yuan · Zhifeng Bao · Mark Sanderson · Yong Tang*

Received: date / Accepted: date

Abstract Neural sequence-to-sequence (seq2seq) models have been widely used in abstractive summarization tasks. One of the challenges of this task is redundant contents in the input document often confuses the models and leads to poor performance. An efficient way to solve this problem is to select salient information from the input document. In this paper, we propose an approach that incorporates word attention with multilayer convolutional neural networks (CNNs) to extend a standard seq2seq model for abstractive summarization. First, by concentrating on a subset of source words during encoding an input sentence, word attention is able to extract informative keywords in the input, which gives us the ability to interpret generated summaries. Second, these keywords are further distilled by multilayer CNNs to capture the coarse-grained contextual features of the input sentence. Thus, the combined word attention and multilayer CNNs modules provide a better-learned representation of the input document, which helps the model generate interpretable, coherent and informative summaries in an abstractive summarization task. We evaluate the effectiveness of our model on the English Gigaword, DUC2004 and Chinese summarization dataset LCSTS. Experimental results show the effectiveness of our approach.

Keywords Abstractive summarization · Word attention · Convolutional neural networks · Sequence-to-sequence model

1 Introduction

Text summarization is the task of generating a coherent and brief summary from an input document while keeping its main concepts. A text summarization sys-

*Corresponding author: Yong Tang
School of Computer Science, South China Normal University, Guangzhou, China
E-mail: ytang@m.scnu.edu.cn

Chengzhe Yuan
School of Computer Science, South China Normal University, Guangzhou, China

Zhifeng Bao · Mark Sanderson
School of Science, Computer Science and Information Technology, RMIT University, Australia

tem is useful in many scenarios, such as headline generation for news articles [8, 34], temporal summarization of real-world events [38,17,14,35]. In general, text summarization methods can be classified into two categories. *Extractive methods* produce summaries by extracting and concatenating important segments from the input document. On the other hand, *abstractive methods* try to build a semantic representation of the document and generate novel summary using words that are not necessarily presented in the input document. As the summary is a condensed version of the document, extractive methods usually suffer from generating less informative summaries [34]. Therefore, abstractive methods have more advantages in the summarization task. However, it is a challenge task to generate human-like summaries.

Input	russian authorities want a european body to investigate the deaths of ### seals in the caspian sea, environmental officials said wednesday.
Reference	russia wants investigation into seal deaths.
Seq2seq	russia seeks probe into <unk><unk>.
WACNNs	russia wants eu to investigate deaths of seals in caspian sea.
Input	the southern chinese city of guangzhou has set up a special zone allowing foreign consulates to build permanent offices and residences and avoid prohibitive local rents , the china daily reported tuesday.
Reference	guangzhou opens new consulate area.
Seq2seq	guangzhou sets up special zone for foreign tourists.
WACNNs	guangzhou sets up special zone for foreign consulates.

Fig. 1 Comparison of two summarization models on news articles from English Gigaword. **Reference** is the reference summary of the Input. The **Seq2seq** model makes factual errors("<unk>" represents unknown word). The **WACNNs** model is accurate with informative words, and keywords of the input are captured by the word attention module in WACNNs model.

Recent success on attentional seq2seq models have shown promising results in abstractive summarization [30,6,28,45]. These models follow the attentional encoder-decoder paradigm, which consists of two steps: (1) encoding an input document into a representation vector, (2) decoding target sentences by automatically relying on different parts of the encoded information. In [45,34], it is shown that considering too much secondary information from the input document will confuse the models and result in poor performance. Moreover, due to the exploding and vanishing gradients problem, it is difficult to encode a long document into an accurate representation vector for seq2seq-based models. Therefore, we believe that in these models, distilling salient information from the input document can improve the quality of generated summaries in an abstractive summarization task.

In order to accomplish the goal of selecting important information from the input document, we should not directly map a word-embeddings based input document to the encoder of a seq2seq model. Inspired by the work in [25] where a local and global attention architecture are explored in a neural machine translation system. In this paper, we propose to build a word attention module after the word-embeddings layer in the encoder. By concentrating on a subset of source words during target sentence generation, the word attention module is able to select in-

formative keywords from the input document. Take the first case in the Figure 1 for example, keywords ("*russian*", "*want*", "*european body*", "*investigate*", "*deaths*", "*seals*", "*caspian sea*", "*environmental*") of the input are correctly captured by the word attention module. On the other hand, these informative keywords give us an insight to understand the generated summaries.

Furthermore, owing to the capability of extracting n-gram features at different positions of a sentence, CNNs have achieved great success in many natural language processing applications, such as text classification, sentiment analysis and seq2seq learning [18,31,12,40,42]. Gehring *et al.* [12] showed that CNNs are capable of building a sequence of feature representations for fixed size contexts and multilayer CNNs can create hierarchical representations over the input sequence which benefits an abstractive summarization task. Inspired by the structure of inception in [32], we implement a similar structure of multilayer CNNs module to further extract coarse-grained contextual features of the input document to benefit an abstractive summarization task. As shown in Figure 1, the combination of word attention and multilayer CNNs modules give us the ability to generate summaries with more informative words, such as "*eu*", "*caspian sea*" and "*foreign consulates*".

In this paper, we propose a Word Attention with Convolutional Neural Networks (WACNNs) approach to extend the standard seq2seq model for an abstractive summarization task. We break down the summarization process of our model into three stages. First, by an WACNNs approach, our model select informative keywords from the input document which helps CNNs capture contextual features of the source sentence; Then, an RNN encoder maps these contextual features into a better-learned representation of the input document; Finally, a summary decoder generates the intended ouput sentences using the distlled encoded information. We make the following contributions.

- We propose an approach that incorporates word attention with multilayer CNNs modules to obtain a better-learned representation of the input for an abstractive summarization task. The word attention module offers an insight into the keywords selected from the input that helps us interpret the generated results. The multilayer CNNs are used after the word attention module to capture the coarse-grained contextual features of the input, which shall benefit the seq2seq-based models for the abstractive summarization task.
- We conduct extensive experiments on English datasets: Gigaword, DUC2004 and Chinese summarization dataset: LCSTS. Our WACNNs model significantly outperforms the standard seq2seq model and achieves improvements compared with various baselines in term of ROUGE results (see Section 5).

We organize the paper as follows. Section 2 reviews the existing related works. Section 3 describes the problem formulation and components of our model in details. Experiment setup and result analysis are presented in Section 4 and Section 5. Finally, Section 6 concludes this paper.

2 Related work

In this section, we mainly review three aspects of research related to text summarization. First, we briefly review: extractive and abstractive methods; Then,

we introduce some neural network-based frameworks for abstractive summarization task; Finally, we review some studies on attention mechanisms and CNNs architecture which are applied in sequence-to-sequence learning.

Extractive Methods usually consist of selecting the key sentences of an input document and re-arranging them as summary. Typical early work include [44, 10, 37, 41, 43]. There are several recent work to improve traditional extractive methods, such as Li *et al.* [19] who propose a compressive-based method which combines sentence compression and sentence selection, and Dasgupta *et al.* [9] who propose a graph-based summarization framework that combines submodularity and dispersion.

Abstractive Methods tend to generate an informative summary by paraphrasing the document with novel words or phrases. Genest *et al.* [13] propose an abstractive summarization framework that selects the content of a summary from an abstract representation of the source documents. Bing *et al.* [3] apply an abstraction-based framework which constructs new sentences by exploring more fine-grained phrases. Liu *et al.* [24] apply sentence compression to extractive summaries to generate abstractive summaries. Filippova *et al.* [11] produce an informative sentence by combining several sentences in a word-graph structure. Overall, most of these methods extract words from the input document, and fusions them to produce the final summary.

Neural Network-based frameworks have achieved great success in abstractive summarization task. Rush *et al.* [30] propose a neural attention-based encoder-decoder framework which is trained on a large corpus of news documents and the corresponding headlines. Chopra *et al.* [6] further developed the work [30] by applying a CNN encoder with an RNN decoder model, besides, the model encodes the position information of the input words. Nallapati *et al.* [28] extend their work to a RNN sequence-to-sequence model which integrates some additional linguistic features (NER, POS tags) in the encoder. Zhou *et al.* [45] further develop previous work to a selective gate network which controls the sentence word vector and sentence representation vector from the encoder to the decoder. Recently, there are other studies working on the abstractive summarization task: Ayana *et al.* [1] employ a minimum risk training strategy for model optimization; Ma *et al.* [26] propose a model to improve semantic relevance of generated summaries and source texts; Li *et al.* [20] extend the standard seq2seq framework with a deep recurrent generative decoder model.

Attention Mechanisms have been widely applied and proved to be important and effective in the field of natural language processing (NLP), such as machine translation task [2], dialogue system [27] and abstractive summarization task [7, 33, 29]. The attention mechanism is first used in [2], which allows the model to automatically focus on the most relevant part of an input document instead of using a fixed-length vector to represent the entire input document when generating a target word. After that, there are some studies working on improving the performance of attention mechanism. For example, a self-attention mechanism that tries to relate different positions of a single sequence to compute a representation of the sequence [4, 23, 36], hierarchical attention networks that have two levels of attention mechanisms applied at the word and sentence-level to pay different attention to individual words and sentences accordingly [7, 33, 29].

As mentioned above, most of the approaches fail to consider word level information when calculating the attention weights at each decoder step. In hierarchical

attention network [28,34,7,33,29], they directly incorporate word-level attention into a hierarchical attention computation. Different from these efforts, we focus on the full use of specific source word embedding for word attention.

Convolutional Neural Networks architecture has also achieved great success in NLP tasks, for example, text classification, sentiment analysis [18,31,40,42]. Gehring *et al.* [12] stands out as one of the notable landmarks in the sequence to sequence learning task. It takes advantage of building a sequence of feature representations over the input document and multilayer CNNs can create hierarchical representations over the input sequence which benefits abstractive summarization task. Wang *et al.* [39] further developed the work [12] by incorporating topic information and using self-critical sequence for training optimization.

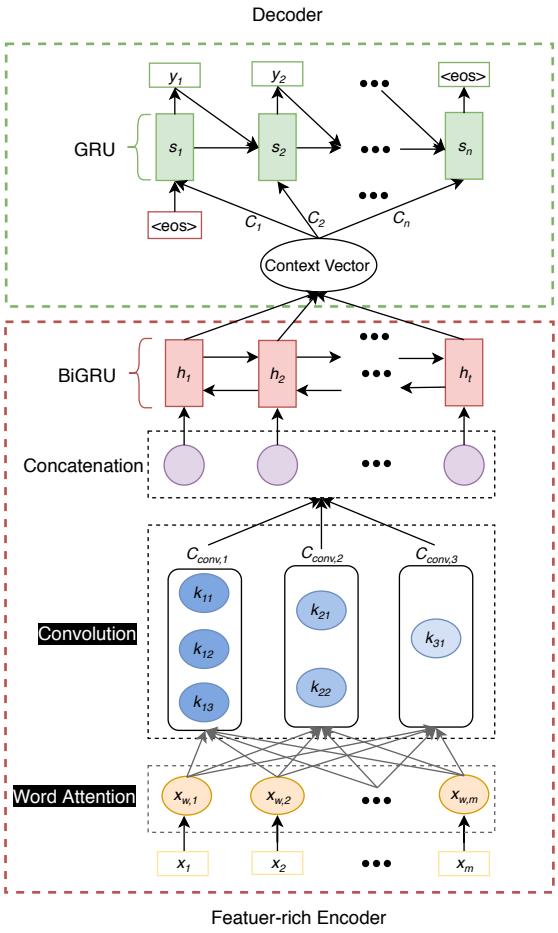


Fig. 2 Overview of the Word Attention with Convolutional Neural Networks (WACNNs), where $k_{i,j}$ in convolution refers to the j-th kernel size in i-th layer of convolution.

However, different from these approaches that directly apply CNNs as the encoder in seq2seq model. We use a word attention module to help multilayer CNNs

extract contextual features of the source sentence, and these contextual features will be further interpreted by the RNN encoder to a better-learned representation of the input document. Besides, different from Wang *et al.* [39] who only concentrate on the results of the summaries, we focus on generating interpretable summaries. Moreover, our work is complementary to abstractive summarization task as our model can be extended to other sequence-to-sequence based frameworks to interpret the generated summaries.

3 Problem formulation & our model

3.1 Problem formulation

We define the task of abstractive summarization as follows: Given a variable-length source document $X = \{x_1, x_2, ..., x_m\}$, in which each word x_t comes from a fixed-length source vocabulary V_s. The system takes X as input, and generates a summary $Y = \{y_1, y_2, ..., y_n\}$, where $n \leq m$ and y_t is also from a fixed-length target vocabulary V_t. We denote this task as abstractive summarization, when not all words in the generated summary are collected from source document. The goal of the task of abstractive summarization is to maximize the conditional probability of Y given input document X and model parameters θ, i.e., arg max $P(Y|X;\theta)$.

3.2 Overview of our model

We describe our model (as shown in Figure 2) in this section. The framework of our model which is based on sequence-to-sequence architecture [2]. In particular, for the encoder, in order to obtain a better-learned representation of the input document, we use a feature-rich encoder which consists of three components: (1) A word attention module; (2) A multilayer convolutional neural networks module; (3) A Recurrent Neural Network (RNN) encoder. For the decoder, we use the standard attention-based RNN decoder to generate the output summary based on the distilled representation. We now describe each component of our model in detail.

3.3 Feature-rich encoder

In seq2seq-based summarization models, the encoder is responsible for mapping the input sentences to a sequence of sentence representations and the decoder is to generate output sentences according to the corresponding sentence representation. The quality of the output summary heavily relies on the encoder hidden state which is regarded as the sentence representation. In addition, some previous work [45, 34] implies that in an abstractive text summarization task, the system does not need to keep all information from the source document, and too much unnecessary information will confuse the model and generate low-quality summaries.

Therefore, we propose a feature-rich encoder architecture which is able to capture contextual features of the source sentence and build a better-leaned representation of the input document for abstractive summarization task. The feature-rich

encoder extends the standard RNN encoder by incorporating word attention with multilayer CNNs modules.

Word attention The role of this module is to learn which words in source sentences are more informative under a given length of a sliding window of input. Before the word attention module, we use word embeddings for the input document, the word embedding layer is utilized for a lookup operation which maps the vectors $X = \{x_1, x_2, \ldots, x_m\}$ to the dense d-dimensional vectors $X_e = \{x_{e,1}, x_{e,2}, \ldots, x_{e,m}\}$:

$$x_{e,i} = W_e x_i \tag{1}$$

Where $x_i \in \mathbb{R}^{|V_s|}$ is a one-hot vector, $x_{e,i} \in \mathbb{R}^d$ is the embedded vector, and $W_e \in \mathbb{R}^{d \times |V_s|}$ denotes the weight of the embedding layer.

As shown in Figure 3, we observe that the word attention module is an n-gram language model, and the mechanism of this module is to apply attention to a fixed-length of sliding windows to the embedded vectors X_e. Assume that $x_{e,i}$ is the center word and the length of the sliding window is L. The attention scores for the word $x_{e,i}$ is defined as follows:

$$X_{e,i}^{w-att} = \{x_{e,i+\frac{-L+1}{2}}, \ldots, x_{e,i}, \ldots, x_{e,i+\frac{L-1}{2}}\}^{\mathrm{T}} \tag{2}$$

$$\alpha_i = h(X_{e,i}^{w-att} \odot \boldsymbol{W_{w-att}} + b_{w-att}), i \in [1, m] \tag{3}$$

Where $\boldsymbol{W_{w-att}} \in \mathbb{R}^{L \times d}$ is the parameter matrix, b_{w-att} is the bias vector. $h(\cdot)$ denotes the sigmoid function, $\odot$ indicates element-wise multiplication, and α_i is the attention score used as a weight for $x_{e,i}$.

In a word attention module, the higher attention score of a given word indicates the more importance of this word. By using attention scores α_i as the weight for word embedding vector $x_{e,i}$, we can obtain the context vectors $X_{att} = \{x_{att,1}, x_{att,2}, \ldots, x_{att,m}\}$, where $x_{att,i}$ is defined as follows:

$$x_{att,i} = \alpha_i * x_{e,i}, i \in [1, m] \tag{4}$$

Convolutional neural networks The context vectors are taken as the input to the multilayer CNNs to further learn coarse-grained local features from the source sentence.

For the convolutional layer, we assume that $x_{e,i} \in \mathbb{R}^d$ denotes the i-th element in the context vector $X_{1:m}^{conv} = \{x_{e,1} \oplus x_{e,2} \oplus \cdots \oplus x_{e,m}\}$, where $\oplus$ is the concatenation operator, and $X_{i:i+j}^{conv}$ represents the concatenation of j consecutive elements $[x_{e,i}, x_{e,i+1}, \ldots, x_{e,i+j}]$. A convolution operation involves applying a *filter* $\boldsymbol{W_{conv}} \in \mathbb{R}^{h \times d}$ sliding over a window of h elements to learn coarse-grained features at different positions. For example, a local feature c_i^{conv} is generated from a window of elements $X_{i:i+h-1}^{conv}$ as follows:

$$c_i^{conv} = f(\boldsymbol{W_{conv}} \cdot X_{i:i+h-1}^{conv} + b_{conv}) \tag{5}$$

Where b_{conv} is a bias vector and $f(\cdot)$ denotes a non-linear activation function. This *filter* $\boldsymbol{W_{conv}}$ is applied to each possible windows of elements in the context vector $[X_{1:h}^{conv}, X_{2:h-1}^{conv}, \ldots, X_{m-h+1:h}^{conv}]$ to generate a feature map:

$$C_{conv} = [c_1^{conv}, c_2^{conv}, \ldots, c_{m-h+1}^{conv}] \tag{6}$$

Where $C_{conv} \in \mathbb{R}^{m-h+1}$. Similar to the structure of inception in [32], we implement a multilayer CNNs to get the hiererchical feature map $\hat{\boldsymbol{C}}_{\boldsymbol{conv}} = \{C_{conv,1} \oplus C_{conv,2} \oplus C_{conv,3}\}$. The new coarse-grained features from source document then will be fed into the recurrent encoder.

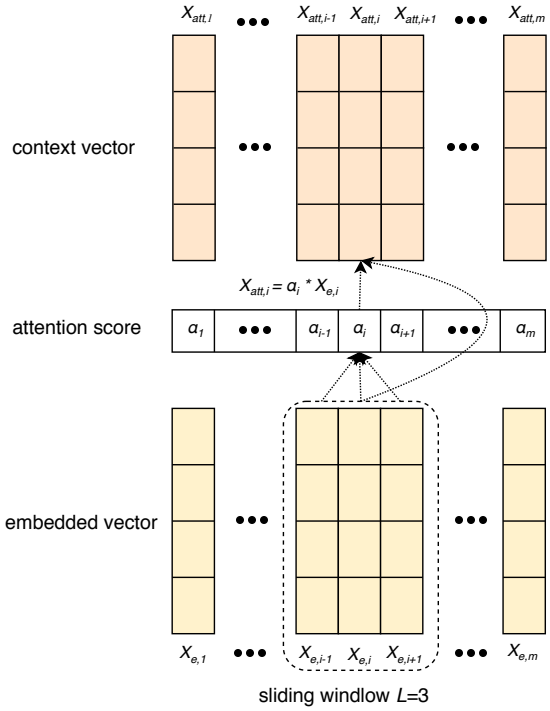


Fig. 3 Illustration of word attention. Context vector is the production between embedded vector and corresponding attention score.

Recurrent encoder In our model, we use a bidirectional gated recurrent unit (GRU) [5] to implement a recurrent encoder, which builds the hidden states h_i as the higher-level of the sentence representation of the input document. Specifically, the GRU is defined by:

$$z_i = \sigma(\boldsymbol{W_z}[x_{enc,i}, h_{i-1}]) \tag{7}$$
$$r_i = \sigma(\boldsymbol{W_r}[x_{enc,i}, h_{i-1}]) \tag{8}$$
$$g_i = tanh(\boldsymbol{W_h}[x_{enc,i}, r_i \odot h_{i-1}]) \tag{9}$$
$$h_i = (1 - z_i) \odot h_{i-1} + z_i \odot g_i \tag{10}$$

Where z_i and r_i are the update gate and reset gate, g_i and h_i are the candidate activation and generated hidden state, $\sigma(\cdot)$ and $tanh(\cdot)$ are the sigmoid function and the hyperbolic tangent activation function.

Let the i-th feature vector $x_{enc,i}$ be the input to the recurrent encoder. Bidirectional GRU (BiGRU) processes the input element $x_{enc,i}$ and the previous hidden

state h_{i-1} in both forward and backward directions respectively:

$$\overrightarrow{h}_i = \text{GRU}(x_{enc,i}, \overrightarrow{h}_{i-1}) \tag{11}$$

$$\overleftarrow{h}_i = \text{GRU}(x_{enc,i}, \overleftarrow{h}_{i-1}) \tag{12}$$

Then, the forward and backward hidden states are concatenated as the final hidden state of the encoder:

$$h_i = \overrightarrow{h}_i \oplus \overleftarrow{h}_i \tag{13}$$

3.4 Summary decoder

We also use a GRU with attention as the decoder to generate the output summary word by word.

At the t-th decoding step, the GRU takes the previous hidden state s_{t-1}, the previous generated summary word y_{t-1}, and the current context vector c_t as input to update the hidden state s_t as:

$$s_t = \text{GRU}(s_{t-1}, y_{t-1}, c_t) \tag{14}$$

Where the context vector c_t is computed by the attention mechanism, which matches the current decoder state s_{t_1} with each encoder hidden state h_i to get an attention score $\alpha_{t,i}$, and then the attention scores are normalized to get the context vector c_t. It is defined as:

$$e_{t,i} = v^T tanh(\boldsymbol{W_{d-att}} s_{t-1} + U_{d-att} h_i) \tag{15}$$

$$\alpha_{t,i} = \frac{exp(e_{t,i})}{\sum_{i=1}^{k} exp(e_{t,i})} \tag{16}$$

$$c_t = \sum_{i=1}^{k} \alpha_{t,i} h_i \tag{17}$$

Where $\boldsymbol{W_{d-att}}$, U_{d-att} and v^T are learnable matrices in the decoder. $\alpha_{t,i}$ indicates how much the i-th element from the encoder hidden states contributes to generating the t-th word in summary and k refers to the number of the encoder hidden states.

When generating the t-th summary word, the decoder uses c_t and s_t to produce the probability distribution of the candidate words as follows:

$$P_{vob} = \varsigma(\boldsymbol{W_y} s_t + b_y) \tag{18}$$

Where $\varsigma(\cdot)$ is a softmax function, $\boldsymbol{W_y}$ is weight matrix, and P_{vob} is a distribution of fixed target vocabulary vector. At each decoding step, a final summary word y_t is generated by sampling from P_{vob} until sampling the end-of-sentence (EOS) tag. The lenght of the generated summaries could be various as the input documents are different.

3.5 Model learning

Given the model parameters θ and training set $\mathcal{D}$, our goal is to maximize the conditional probability of the output summary. Therefore, the learning process is to minimize the negative log likelihood loss function:

$$\mathcal{L} = \sum_{(X,Y)\in\mathcal{D}} -\log\, p(Y|X;\theta) \tag{19}$$

Where $\mathcal{D}$ is the set of input document and corresponding summaries. We use a stochastic gradient descent with mini-batch strategy to optimize the model parameters θ.

4 Experiment setup

We conducted a series of experiments to evaluate performance of different models in the abstractive summarization task. In this section, we describe the datasets, experimental implementation, and baseline models in details.

4.1 Datasets

We use two benchmark datasets that contain both English and Chinese text summarization corpora to train and evaluate the models.

English Gigaword is an English text summarization dataset constructed from the Annotated English Gigaword dataset[1], which is currently the largest static English news corpus, containing nearly ten million news articles from seven news sources over the last two decades. We use the script released by *Rush et al.* [30] to obtain the same version of the dataset, which is constructed by extracting the first sentence and the headline of each news article to form sentence-summary pairs. To filter noisy data in news articles, the script also contains various pre-processing steps, including PTB tokenization, lower-casing, replacing all digit characters with the"#" symbol, and replacing of word types seen less than 5 times with the "$\langle unk \rangle$" symbol. The filtered dataset contains about 3.8M training pairs, 190K development pairs, and 2,000 test pairs.

To make the evaluation on the English Gigaword corpus more objective, we take the **DUC-2004** Task-1[2] as our English test set. This test set has 500 news articles collected from the Associated Press Wire services and the New York Times. Each article contains four human-generated reference summaries, with a maximum length of 75 characters.

LCSTS is a large-scale Chinese short text summarization dataset, consisting of article-summary pairs collected from a popular Chinese social media website named Sina Weibo[3][16]. The articles posted are shorter than 140 Chinese characters, and with corresponding human-written reference summaries. The LCSTS is split into three parts. Part-I contains 2.4M article-headline pairs, Part-II and

[1] https://catalog.ldc.upenn.edu/ldc2012t21
[2] https://duc.nist.gov/duc2004/
[3] http://www.weibo.com

Part-III consist of 10.7k and 1.1k pairs with human-labeled scores from 1 to 5, the scores show how relevant of an article is to its summary. We only keep pairs with scores no less than 3. We use Part-I, Part-II, and Part-III as a training set, development set, and test set respectively.

4.2 Evaluation metric

We use ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [21], which is a measure to evaluate performance on the task of automatic summarization as our evaluation metric. The essence of ROUGE is a count of the number of overlapping units between candidate summary and a set of reference summaries. In our work, we choose ROUGE-N and ROUGE-L. ROUGE-N the n-grams recall defined as follows:

$$\text{ROUGE-N} = \frac{\sum_{S \in Ref} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in Ref} \sum_{gram_n \in S} Count(gram_n)} \tag{20}$$

where n represents the length of n-grams, Ref is the set of the reference summaries. $Count(gram_n)$ is the number of n-grams in the reference summaries, $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in the candidate summary and a set of reference summaries. ROUGE-L is a longest common subsequence (LCS) based measures.

Following previous work [30, 16], we implement ROUGE-1 (unigram), ROUGE-2 (bigram), ROUGE-L (LCS) respectively in the reported experimental result.

4.3 Implementation

Our experiments are implemented in TensorFlow[4] on a single NVIDIA Tesla P100 GPU. Our model is developed based on the framework of an attentional sequence-to-sequence model [2].

For the experiments on the English Gigaword dataset, we set the word embedding dimension to 300; the dimension of GRU hidden states in both encoder and decoder to 512, the input and output vocabulary size are both constrained to 35,000; we use $L = 5$ as the length of sliding window for word attention; and the filter size in multilayer CNNs module is $[1, 3, 3; 1, 3; 1]$. For the experiments on the LCSTS dataset, the word embedding dimension is set to 256, and the dimension of all GRU hidden states are also set to 400. The input and output vocabulary sizes are constrained to 50,000 and 35,000 respectively, and we use $L = 9$ as the length of the sliding window for word attention, and the filter size in multilayer CNNs is $[1, 5, 5; 1, 5; 1]$

In both experiments, the batch size is set to 64, the initial weight parameters are uniformly sampled in the range [-0.1,0.1], we use the Adadelta optimizer with the learning rate of 1.0, $\rho = 0.95$ and $\epsilon = 1e - 6$. The gradient clipping is applied in the range [-1.0,1.0].

[4] https://www.tensorflow.org/

4.4 Baseline

As we use the standard datasets, we evaluate our experiments with the results of the baselines reported in their papers. The two datasets have different baselines. Therefore, we introduce the baselines for English Gigaword and LCSTS respectively, besides, we also implement a standard seq2seq model (denote it for "seq2seq") and seq2seq+CGU model [22] which is a global encoding framework for abstractive summarization as baselines for both datasets.

We compare the WACNNs model with the following baselines for English Gigaword dataset.

- **ABS** [30] is a neural network based model with an attentive CNN encoder and Neural Network Language Model decoder for abstractive sentence summarization.
- **ABS+** [30] uses a further extractive tuning step based on ABS model with an additional handcrafed features.
- **lvt2k-1sent** and **lvt5k-1sent** [28] are RNN seq2seq models using large vocabulary trick. Besides, both encoders are integrated with additional linguistic features, such as part-of-speech tags and named-entity tags.
- **RAS-Elman** and **RAS-LSTM** [6] are extensions of the ABS model with an attentive CNN encoder with Elman RNN decoder and LSTM decoder respectively.
- **RNN+MLE** and **RNN+MRT** [1] employ minimum risk training for model optimization, which directly optimizes model parameters in sentence level with respect to the evaluation metrics.
- **SEASS** [45] extends the sequence-to-sequence framework with a selective encoding model, which consists of a sentence encoder and a selective gate network.
- **ConvS2S** [12] is a fully convolutional model for sequence-to-sequence learning.
- **SeqCopyNet** [46] is a sequential copying networks to model the sequential copying phenomenon in seq2seq generation.

For the LCSTS dataset, the compared baselines are introduced as follows.

- **RNN** and **RNN-context** [16] are both RNN-based seq2seq models, the difference is RNN-context uses an attention mechanism.
- **CopyNet** [15] is a sequence-to-sequence model integrated with a copying mechanism.
- **SRB** [26] is a semantic relevance based neural model to improve semantic similarity between texts and summaries.
- **DRGD** [20] is a standard seq2seq framework with a deep recurrent generative decoder model.

5 Results and discussions

In this section, we report and analyze the results of our experiments with ROUGE metrics[5]. We obtain our ROUGE scores using the pyrouge package[6], and we classify the models according to two types of encoder architecture: CNNs and RNN. In

[5] We use RG-1, RG-2, and RG-L denote ROUGE-1, ROUGE-2, and ROUGE-L.

[6] https://pypi.org/project/pyrouge/0.1.3/

addition, we explain the effectiveness of the WACNNs model by a set of generated summaries.

5.1 ROUGE evaluation

For both the English Gigaword test set and the DUC-2004 test set, we train our model using the English Gigaword train set. In Table 1, we report the ROUGE F1 results of our model and baseline models on English Gigaword test set, and the ROUGE Recall results on DUC-2004 dataset are shown in Table 2. Our model outperforms most of the baseline models on the ROUGE metrics in both test sets, and has comparable performance with seq2seq+CGU model [22] on Gigaword test set.

Table 1 ROUGE-F1 evaluation results on English Gigaword test set.

Models	Encoder	RG-1	RG-2	RG-L
ABS		29.55	11.32	26.42
ABS+		29.78	11.89	26.97
RAS-LSTM	CNNs	32.55	14.70	30.03
RAS-Elman		33.78	15.97	31.15
ConS2S		35.88	17.48	33.29
lvt2k-1sent		32.67	15.59	30.64
lvt5k-1sent		35.30	16.64	32.62
RNN+MLE		32.67	15.23	30.56
RNN+MRT		36.54	16.59	33.44
SEASS	RNN	36.15	17.54	33.63
SeqCopyNet		35.93	17.51	33.35
seq2seq+CGU		36.30	**18.00**	33.80
seq2seq (our impl.)		33.64	15.94	31.66
WACNNs		**36.55**	17.74	**33.83**

English Gigaword. In Table 1, it is clear that our WACNNs model achieves full-length F1 scores of 36.55 ROUGE-1, 17.74 ROUGE-2 and 33.83 ROUGE-L, and WACNNs model performs better than most of the competitor models and has comparable performance with seq2seq+CGU model.

Among baseline models, we first compare our model with the seq2seq model, it shows that our WACNNs model has a gain over the se2seq model by 2.91 ROUGE-1, 1.8 ROUGE-2, and 2.17 ROUGE-L scores relatively, which demonstrates the efficiency of our model. Moreover, by using the coarse-grained contextual features distilled from the input by the word attention and multilayer CNNs modules, our model has better performance than the best baseline seq2seq+CGU in ROUGE-1 and ROUGE-L results in RNN Encoder and achieves a higher ROUGE scores over the best baseline, ConS2S, in CNN Encoder. By controlling the information of the source context from encoder to decoer, which is similar to the idea of our model, seq2seq+CGU model has comparable performance with our model. In addition, different from integrating additional handcraft linguistic features, such as POS tag (lvt2k-1sent and lvt5k-1sent models), word position (RAS-Elman and RAS-LSTM models) to enrich encoding inform, and different from other mechanisms such as selective gate mechanism (SEASS) and sequential copying networks (SeqCopyNet), our model outperforms these baseline models by capturing semantic information of

the source sentence through the combination of word attention and CNNs modules. Moreover, instead of directly using CNNs as the encoder in the ABS and ABS+ models or optimizing the evaluation metric in the RNN+MRT and RNN+MLE models, our model also performs better than these models by selecting information keywords from the input document through the word attention module and long-distance dependencies learned by the RNN encoder.

Table 2 ROUGE-Recall evaluation results at 75 bytes on DUC-2004 test set.

Models	Encoder	RG-1	RG-2	RG-L
ABS	CNNs	26.55	7.06	22.05
ABS+		28.18	8.49	23.82
RAS-LSTM		27.41	7.69	23.06
RAS-Elman		28.97	8.26	24.06
ConS2S		30.44	10.84	26.90
lvt2k-1sent	RNN	28.35	9.46	24.59
lvt5k-1sent		28.61	9.42	25.24
RNN+MLE		24.92	8.60	22.25
RNN+MRT		30.41	**10.87**	26.79
SEASS		29.12	9.56	25.51
seq2seq+CGU[7]		29.62	10.08	26.17
seq2seq (our impl.)		28.02	9.37	24.85
WACNNs		**30.54**	**10.87**	**26.94**

When **DUC-2004** is the evaluation dataset, we train our model on the same Gigaword dataset. Table 2 summaries the ROUGE recall scores of our WACNNs model and the baseline methods. We can notice that our model achieves the best performance on the ROUGE-1 and ROUGE-L scores, and is comparable to the ROUGE-2 score. Besides, our model also has better performance than seq2seq+CGU which is one of the best baseline model in the English datasets. Due to the similarity of the DUC-2004 and Gigaword test set, we do not provide qualitative analysis in this experiment.

Table 3 ROUGE-F1 evaluation results on LCSTS, the word-level ROUGE scores are presented as WACNNs/w and the character-level as WACNNs/c.

Models	Encoder	RG-1	RG-2	RG-L
RNN	RNN	21.50	8.90	18.60
RNN-context		29.90	17.40	27.20
SRB		33.30	20.00	30.10
CopyNet		34.40	21.60	31.30
DRGD		37.00	24.20	34.20
seq2seq+CGU		**39.40**	**26.90**	**36.50**
seq2seq (our impl.)		33.20	22.50	31.80
WACNNs/w		38.80	24.40	34.50
WACNNs/c		**39.40**	25.90	35.90

For **LCSTS dataset**, we use two approaches to preprocess it: character-based and word-based. As shown in Table 3, we observe little differences in the performances of our model from LCSTS dataset. Compared to the best baseline,

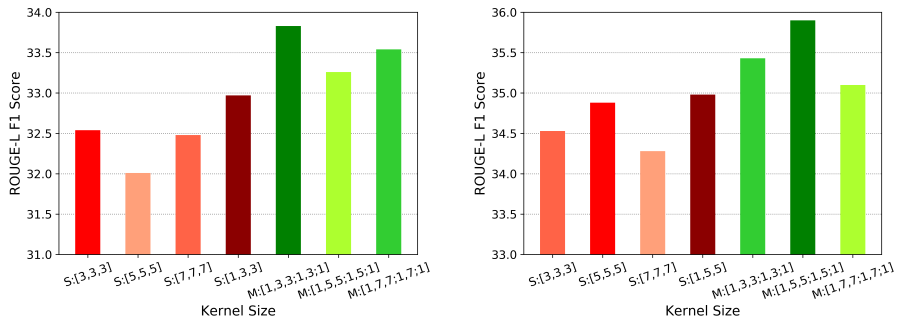
[7] We implement the code: https://github.com/lancopku/Global-Encoding

seq2seq+CGU, our model achieves comparable performance on ROUGE-1 result but has slightly worse performance on ROUGE-2 and ROUGE-L results. For the different performances in English datasets (English Gigaword and DUC-2004) and Chinese dataset LCSTS. One reason could be that the seq2seq+CGU model provides better representations of the source information when the scale of train data is relatively small (LCSTS dataset contains 2.4M sentence pairs), but our model is able to capture better representations of the input document when the scale of train data is relatively large (Gigaword corpus contains 3.8M sentence pairs). In addition, even though DRGD equips with a deep recurrent generative decoder, CopyNet employs a copying mechanism that copies words from the input document, SRB uses a semantic relevance based neural model and RNN-context utilizes attention mechanism in seq2seq model, our model still performs better than these models.

The advantages of ROUGE scores on both English and Chinese suggests that, our model is able to build a better-learned representation of the input document to improve the quality of summary.

5.2 Kernel size analysis

We analyze the impact of different kernel sizes in the convolutional layer on the model's performance. Convolutional neural networks enable the model to extract n-gram features from the sentence. Intuitively, we believe that multiple convolutional layers are able to perform better than a single convolutional layer, as multilayer CNNs can create hierarchical representations over the input sequence [12]. Inspired by the inception architecture [32], we follow the design principle of inception and take the kernel size from $k = [3, 5, 7]$.



(a) English Gigaword (b) LCSTS

Fig. 4 ROUGE-L F1 scores on English Gigaword and LCSTS datasets with different kernel size strategies. For x-axis, S indicates a single convolutional layer (shades of red), and M indicates multiple convolutional layers (shades of green) with different kernel size (3,5,7).

Figure 4(a) and Figure 4(b) show the ROUGE-L F1 performance with different a kernel size (3,5,7) on English Gigaword and LCSTS datasets in the convolutional layer respectively. In English Gigaword, we find that the multilayer CNNs with

kernel size $[1, 3, 3; 1, 3; 1]$ achieves the best performance. In a single convolutional layer, the model with different kernel size $[1, 3, 3]$ outperforms models with the same kernel size $[3, 3, 3]$, $[5, 5, 5]$ and $[7, 7, 7]$ respectively. A similar performance can be noticed in the LCSTS dataset, it is observed that model with the kernel size $[1, 5, 5; 1, 5; 1]$ performs better than a kernel size with $[1, 3, 3; 1, 3; 1]$ and $[1, 7, 7; 1, 7; 1]$, besides, in single convolutional layer, the model with different kernel sizes obtains improvements compared to the models with the same kernel size. We assume that the different length of the input document in English Gigaword and LCSTS datasets leads to the different model performances in kernel size. Overall, we can learn that the abstractive summarization task shall benefit from the model with multiple convolutional layers.

5.3 Case study

To investigate the effectiveness of WACNNs approach in abstractive summarization task, we show some generated summaries by our model and the standard seq2seq model. In Table 4, we report some examples of generated summaries from the English Gigaword and LCSTS test sets.

We can learn that in S(1), the source document introduces northeast china province fights drought, however, the seq2seq model generates a summary which contains the word "*drought*" twice and incorrectly describes the source document. In the example of S(2), the summary "*portugal agrees to take two syrian detainees from guantanamo*" generated by our model concisely conveys the idea of S(2), whereas, the output of the seq2seq model misinterprets its meaning by word "*prisoners*", besides, the seq2seq model also lacks the keyword "*syrian*". Similar results can be noticed in LCSTS test set. In the example of S(3), the main idea in this sentence is about a suspect, who hurled a baby girl to her death in Beijing, was arrested. However, the seq2seq model result is "The suspect who hurled a baby girl to her death in Beijing was *not satisfied*." which fails to express this idea because of an irrelevant phrase "*not satisfied*". Our model is able to generate a more informative summary than the seq2seq model with extra words (*Daxing*) from the input document. Another evidence of the ability in generating a better summary by our model can be found in S(4), as the seq2seq model generates a less accurate summary "The State Council speeds up export tax rebate policies".

Overall, the examples in Table 4 demonstrate that the effectiveness of our WACNNs model in generating high-quality summaries with additional informative words.

5.4 Word attention with CNNs modules analysis

In order to illustrate the effectivenss of word attention with CNNs modules in bulding a better-learned representation of input document and the ability to interpret generated summaries, we compare our model with basic seq2seq baseline and visualize the word attention mechanism.

Effectiveness of word attention with CNNs modules. Similar to the settings in SEASS model [45], we compare our model with basic seq2seq baseline on English Gigaword and LCSTS test sets with the length of sentence ranging

Table 4 Examples of the generated summaries on the English gigaword (S(1), S(2)) and LCSTS test sets (S(3), S(4)). **S**: source document, **Reference**: reference summary, **seq2seq**: output of the seq2seq model, **WACNNs**: output of our model.

S(1): a series of emergency measures have been taken in northeast china's liaoning province in a massive campaign against drought believed to be the most serious since ####. **Reference**: northeast china province fights drought. **seq2seq**: drought measures against drought in northeast china. **WACNNs**: northeast china province launches **emergency measures** against drought.
S(2): portugal has agreed to take two syrian detainees from guantanamo on humanitarian grounds , the government said friday – becoming the third eu nation to accept inmates from the u.s. military prison . **Reference**: portugal taking # syrian guantanamo detainees. **seq2seq**: portugal to accept # guantanamo prisoners. **WACNNs**: portugal agrees to take **two syrian detainees** from guantanamo.
S(3): #月#日，北京市大兴区摔死女童案犯罪嫌疑人韩某、李某分别因涉嫌故意杀人罪和窝藏罪，被依法批准逮捕。#月#日，两名驾车男子因不满一名推着婴儿车的女士挡道与该女士发生争执，过程中一名男子将婴儿车内的女童摔在地上，导致女童死亡。 On #/#(date), two suspects Han and Li, who were accused of killing a baby girl by hurling her to the ground in Daxing district of Beijing, were formally arrested on suspicion of intentional homicide and harbouring a criminal respectively. On #/#(date), two-man had an altercation with a woman for blocking the car's path with the pushchair, during the argument, one man snatched the baby from a pram and hurled it to its death. **Reference**: 北京摔死女童案疑犯被逮捕 The suspect who hurled a baby girl to her death in Beijing was arrested. **seq2seq**: 北京摔死女童案嫌犯不满 The suspect who hurled a baby girl to her death in Beijing was not satisfied. **WACNNs**: 北京**大兴**摔死女童案嫌疑人被批捕 The suspect who hurled a baby girl to her death in ***Daxing*** of Beijing was arrested.
S(4): 国务院近日发布关于促进外贸稳定增长若干意见的正式文件，确定加快出口退税、扩大贸易融资规模等政策。新华分析师认为，接下来促外贸的各项措施会进入密集的落实期。在出口退税、金融等方面出台更细化的措施，此次政策着力点集中于降低出口企业成本。 Recently, the State Council issued a document "Several Opinions on Promoting the Steady Growth of International Trade" and confirmed policies such as speeding up export tax rebates and expanding the scale of trade finance. Analysts at Xinhua News Agency believe the implementation of various measures to promote international trade. More detailed measures will be released in the fields of export tax rebates and finance, the focus of the policies is on reducing the cost of export enterprises. **Reference**: 国务院下发促外贸稳增长意见 The State Council issued opinions on promoting the steady growth of international trade. **seq2seq**: 国务院加快出口退税政策 The State Council speeds up export tax rebate policies. **WACNNs**: 国务院出台促外贸稳增长意见 The State Council introduced opinions on promoting the steady growth of international trade.

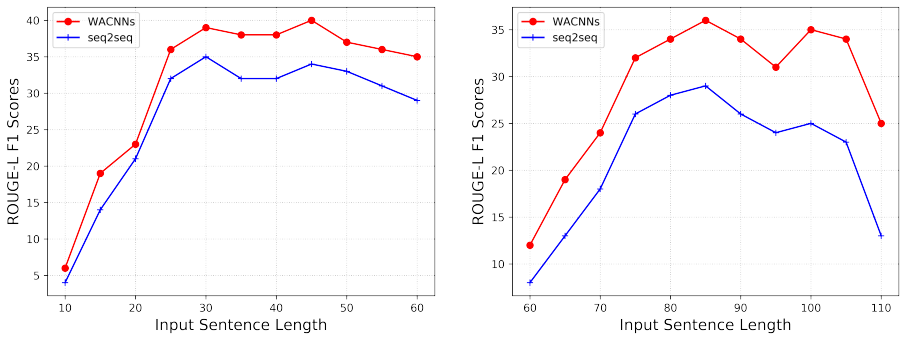
from 10 to 60 and 60 to 110 respectively. Besides, we group both test sets into 12 groups with an interval of 5.

Figure 5(a) and Figure 5(b) show the ROUGE-L F1 scores on English Gigaoword and LCSTS test sets with different input sentence length for WACNNs model and seq2seq baseline respectively. We find that our WACNNs model consistently outperofmrs the seq2seq model across all ranges of input sentence lengths on Gigaword and LCSTS test sets. In the English Gigaword test set, we notice that when the input sentence length is relatively small (from 10 to 25), both models have similar performance in ROUGE-L results, but our model outperforms

Table 5 Highlighted words by word attention module for the examples in Table 4. Colored words are considered as informative words. Darker color indicates higher word attention scores.

S(1): a series of emergency measures have been taken in northeast china 's liaoning province in a massive campaign against drought believed to be the most serious since ####.
WACNNs: northeast china province launches emergency measures against drought.
S(3): #月#日，北京市 大兴区 摔 死 女童案 犯 罪 嫌疑人 韩 某 、 李 某 分 别 因 涉 嫌 故意 杀人罪 和 窝藏罪，被依法批准 逮捕 。#月#日，两名驾车男子因不满一名推着婴儿车的女士 挡道 与该女士发生 争执，过程中一名男子将婴儿车内的女童摔在地上，导致女童死亡。 On #/#(date), two suspects Han and Li, who were accused of killing a baby girl by hurling her to the ground in Daxing district of Beijing, were formally arrested on suspicion of intentional homicide and harboring a criminal respectively. On #/#(date), two man had an altercation with a woman for blocking the car's path with the pushchair, during the argument, one man snatched the baby from a pram and hurled it to its death.
WACNNs: 北京大兴摔死女童案嫌疑人被批捕 The suspect who hurled a baby girl to her death in Daxing of Beijing was arrested.

seq2seq model when the input senten length is greater than 30, especially, for the sentence length of 30, 35, 40, 45 and 60, the WACNNs model obtains signficant improvements compared to the seq2seq model. Compare to English Gigaword, we find little different performance in the LCSTS. In specific, both models have big drop in performance for the sentence length of 90, 95 and 110. We assume that the different size of English Gigaword (1.9K) and LCSTS (0.8K) test sets leads to the different model performance in input sentence length. Overall, these improvements on both test sets verify the effectiveness of the word attention with CNNs modules in abstractive summarization task.



(a) English Gigaword (b) LCSTS

Fig. 5 ROUGE-L F1 scores on English Gigaoword and LCSTS test sets with different input sentence length for WACNNs model and seq2seq baseline.

Visualization of word attention. Since the hierarchical representations by CNNs module is a high dimensional vector, which makes it hard to visualize all the vector values, we discuss our findings about the word attention module helping to interpret the generated summaries. We highlight words that have high word attention scores by equation (3). These words are considered as informative in the source document. Two examples from the Gigaword and LCSTS test data are shown in Table 5. Words with different level of scores are colored different shades of blue: darker blue indicates higher word attention scores, and words with low scores are not colored.

As we notice from S(1), the word attention module tends to assign the highest attention to informative noun: For example "*emergency measures*", "*northeast china*", and "*drought*". These important keywords precisely describe the idea of this sentence. In S(3) we also find that the word attention module successfully picks informative keywords from the sentence, such as: "*Beijing*","*suspects*", "*baby girl*","*hurling*", "*arrested*", and "*killing*". Furthermore, other important words such as "*Daxing district*", "*intentional homicide*", and "*harboring a criminal*" are given second-level scores. Hence, these two examples show that the proposed word attention module is able to identify more important words from the source sentence, which helps to provide a tailored sentence encoding for the abstractive summarization task, and enables us to explain the generated summaries.

6 Conclusion and future work

In this paper, we proposed a novel approach that combines word attention with a multilayer convolutional neural networks to extend the standard seq2seq model for abstractive summarization. The proposed model consists of three steps: First, we use a word attention module to select informative keywords from input document and multilayer CNNs to capture contextual features of the input. Second, the model encodes the contextual features into a better-learned representation of the input document. Last, the model decodes the target summary with the distilled encoded information. Experimental results on three benchmark datasets in different languages showed that our model outperforms a variety of existing models. In the future, we will also try to analyze the effect of the hierachical representations of the input documents and expand the application of our model to evaluate summaries of multi-documents.

Acknowledgements This work was supported by the China Scholarship Council (CSC) for award No. 201706750031, NSFC (No.61772211, 61728204, 91646204), ARC (DP170102726, DP180102050), Special Project on the Integration of Industry & Academia and Synergy of Research of Guangzhou, China (No. 201704020203), Special Fund for Applied Program of Science and Technology of Guangdong Province, China (No. 2016B010124008), the Innovation Project of Graduate School of South China Normal University. Zhifeng Bao is a recipient of Google Faculty Award.

References

1. Ayana, Shen, S., Liu, Z., Sun, M.: Neural headline generation with minimum risk training. arXiv:1604.01904 (2016)

2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv:1409.0473 (2014)
3. Bing, L., Li, P., Liao, Y., Lam, W., Guo, W., Passonneau, R.J.: Abstractive multi-document summarization via phrase selection and merging. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, pp. 1587–1597 (2015)
4. Cheng, J., Dong, L., Lapata, M.: Long short-term memory-networks for machine reading. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 551–561 (2016)
5. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1724–1734 (2014)
6. Chopra, S., Auli, M., Rush, A.M.: Abstractive sentence summarization with attentive recurrent neural networks. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 93–98 (2016)
7. Cohan, A., Dernoncourt, F., Kim, D.S., Bui, T., Kim, S., Chang, W., Goharian, N.: A discourse-aware attention model for abstractive summarization of long documents. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 615–621 (2018)
8. Colmenares, C.A., Litvak, M., Mantrach, A., Silvestri, F.: HEADS: headline generation as sequence prediction using an abstract feature-rich space. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 133–142 (2015)
9. Dasgupta, A., Kumar, R., Ravi, S.: Summarization through submodularity and dispersion. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 1014–1022 (2013)
10. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. **22**, 457–479 (2004)
11. Filippova, K.: Multi-sentence compression: Finding shortest paths in word graphs. In: Proceedings of the Conference COLING 2010, 23rd International Conference on Computational Linguistics, pp. 322–330 (2010)
12. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: Proceedings of the 34th International Conference on Machine Learning, pp. 1243–1252 (2017)
13. Genest, P., Lapalme, G.: Framework for abstractive summarization using text-to-text generation. In: Proceedings of the Workshop on Monolingual Text-To-Text Generation@ACL, pp. 64–73 (2011)
14. Georgescu, M., Pham, D.D., Kanhabua, N., Zerr, S., Siersdorfer, S., Nejdl, W.: Temporal summarization of event-related updates in wikipedia. In: Proceedings of the 22nd International World Wide Web Conference, pp. 281–284 (2013)
15. Gu, J., Lu, Z., Li, H., Li, V.O.K.: Incorporating copying mechanism in sequence-to-sequence learning. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 1631–1640 (2016)
16. Hu, B., Chen, Q., Zhu, F.: LCSTS: A large scale chinese short text summarization dataset. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1967–1972 (2015)
17. Huang, Y., Shen, C., Li, T.: Event summarization for sports games using twitter streams. World Wide Web **21**(3), 609–627 (2018)
18. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2267–2273 (2015)
19. Li, C., Liu, F., Weng, F., Liu, Y.: Document summarization via guided sentence compression. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 490–500 (2013)
20. Li, P., Lam, W., Bing, L., Wang, Z.: Deep recurrent generative decoder for abstractive text summarization. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2091–2100 (2017)

21. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, pp. 74–81 (2004)
22. Lin, J., Sun, X., Ma, S., Su, Q.: Global encoding for abstractive summarization. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers, pp. 163–169 (2018)
23. Lin, Z., Feng, M., dos Santos, C.N., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. arXiv:1703.03130 (2017)
24. Liu, F., Liu, Y.: From extractive to abstractive meeting summaries: Can it be done by sentence compression? In: Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp. 261–264 (2009)
25. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421 (2015)
26. Ma, S., Sun, X., Xu, J., Wang, H., Li, W., Su, Q.: Improving semantic relevance for sequence-to-sequence learning of chinese social media text summarization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 635–640 (2017)
27. Mei, H., Bansal, M., Walter, M.R.: Coherent dialogue with attention-based language models. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp. 3252–3258 (2017)
28. Nallapati, R., Zhou, B., dos Santos, C.N., Gülçehre, Ç., Xiang, B.: Abstractive text summarization using sequence-to-sequence rnns and beyond. In: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, pp. 280–290 (2016)
29. Nema, P., Khapra, M.M., Laha, A., Ravindran, B.: Diversity driven attention model for query-based abstractive summarization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 1063–1072 (2017)
30. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 379–389 (2015)
31. Severyn, A., Moschitti, A.: Twitter sentiment analysis with deep convolutional neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 959–962 (2015)
32. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
33. Tan, J., Wan, X., Xiao, J.: Abstractive document summarization with a graph-based attentional neural model. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 1171–1181 (2017)
34. Tan, J., Wan, X., Xiao, J.: From neural sentence summarization to headline generation: A coarse-to-fine approach. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, pp. 4109–4115 (2017)
35. Tran, T.A., Niederée, C., Kanhabua, N., Gadiraju, U., Anand, A.: Balancing novelty and salience: Adaptive learning to rank entities for timeline summarization of high-impact events. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, pp. 1201–1210 (2015)
36. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, pp. 6000–6010 (2017)
37. Wan, X., Yang, J., Xiao, J.: Manifold-ranking based topic-focused multi-document summarization. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 2903–2908 (2007)
38. Wang, C., He, X., Zhou, A.: Event phase oriented news summarization. World Wide Web **21**(4), 1069–1092 (2018)
39. Wang, L., Yao, J., Tao, Y., Zhong, L., Liu, W., Du, Q.: A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden., pp. 4453–4460 (2018)

40. Wang, S., Huang, M., Deng, Z.: Densely connected CNN with multi-scale feature attention for text classification. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, pp. 4468–4474 (2018)
41. Wong, K., Wu, M., Li, W.: Extractive summarization using supervised and semi-supervised learning. In: Proceedings of the 22nd International Conference on Computational Linguistics, pp. 985–992 (2008)
42. Yoon, J., Kim, H.: Multi-channel lexicon integrated cnn-bilstm models for sentiment analysis. In: Proceedings of the 29th Conference on Computational Linguistics and Speech Processing, pp. 244–253 (2017)
43. Yuan, C., Li, D., Zhu, J., Tang, Y., Wasti, S., He, C., Liu, H., Lin, R.: Citation based collaborative summarization of scientific publications by a new sentence similarity measure. In: Collaborative Computing: Networking, Applications and Worksharing - 13th International Conference, CollaborateCom 2017, Edinburgh, UK, December 11-13, 2017, Proceedings, pp. 680–689 (2017)
44. Zajic, D., Dorr, B., Schwartz, R.: Automatic headline generation for newspaper stories. In: Workshop on Text Summarization, pp. 78–85 (2002)
45. Zhou, Q., Yang, N., Wei, F., Zhou, M.: Selective encoding for abstractive sentence summarization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 1095–1104 (2017)
46. Zhou, Q., Yang, N., Wei, F., Zhou, M.: Sequential copying networks. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pp. 4987–4995 (2018)