applied sciences

MDPI

Article

Comprehensive Document Summarization with Refined Self-Matching Mechanism

Biqing Zeng [1], Ruyang Xu [2,*], Heng Yang [2], Zibang Gan [1] and Wu Zhou [2]

[1] School of Software, South China Normal University, Foshan 528225, China; zengbiqing@scnu.edu.cn (B.Z.); ganzibang@m.scnu.edu.cn (Z.G.)
[2] School of Computer, South China Normal University, Guangzhou 510631, China; yangheng@m.scnu.edu.cn (H.Y.); zwier@m.scnu.edu.cn (W.Z.)
* Correspondence: cs_xuruyang@m.scnu.edu.cn

Received: 13 January 2020; Accepted: 2 March 2020; Published: 9 March 2020

Abstract: Under the constraint of memory capacity of the neural network and the document length, it is difficult to generate summaries with adequate salient information. In this work, the self-matching mechanism is incorporated into the extractive summarization system at the encoder side, which allows the encoder to optimize the encoding information at the global level and effectively improves the memory capacity of conventional LSTM. Inspired by human coarse-to-fine understanding mode, localness is modeled by Gaussian bias to improve contextualization for each sentence, and merged into the self-matching energy. The refined self-matching mechanism not only establishes global document attention but perceives association with neighboring signals. At the decoder side, the pointer network is utilized to perform a two-hop attention on context and extraction state. Evaluations on the CNN/Daily Mail dataset verify that the proposed model outperforms the strong baseline models and statistical significantly.

Keywords: extractive summarization; deep learning; document summarization

1. Introduction

Automatic summarization systems have been made great progress in many applications, such as headline generation [1], single or multi-document summarization [2,3], opinion mining [4], text categorization, etc. The system aims to shorten the input and retain the salient information from the source document. Practical needs for such systems grow with the continuous increasing text sources in various fields. Text summarization methods would be divided into two categories: abstractive and extractive. The extractive methods select salient informative sentences from the source document as a summary, while the abstractive methods can generate words or sentences that are not present in the source document. The abstractive summarization is more difficult as it has to deal with factual or grammatical errors, semantic incoherence, as well as problems with the obtaining of explicit textual paraphrases and generalizations. Extractive methods relieve these problems by identifying important sentences from the document, therefore summary generated by extractive methods are generally better than that generated by abstractive methods in terms of grammaticality and factuality. However, those methods may encounter problems like the lack of core information and incomprehensive generalization. With the advantages of simpler calculation and higher generation efficiency, numerous empirical comparisons in recent years have shown that the state-of-the-art extractive methods usually have better performance than the abstractive ones [5].

Classical document extractive summarizer relies on sophisticated feature engineering that mainly based on the statistical properties of the document, such as word probability, term frequency–inverse document frequency (TF-IDF) weights, sentence position, sentence length, etc. [6]. Graph-based

methods, such as Lexrank [7], and TextRank [8], use graph weights to measure the sentence importance. In recent years, some neural network-based methods [9–12] have been proposed and applied to news datasets. Deep learning models with thousands of parameters require large annotated datasets. In the summarization field, Chen et al. [13] overcame this difficulty by creating news stories datasets from Central News Networks (CNN) and Daily Mail, which consist of 280 K documents and human writing summaries.

Deep learning models would learn hidden features of text owing to their strong generalization ability, they avoid burdensome manual feature extraction. Such advantages would be promoting to achieve end-to-end integration of key content selection and importance assessment modules in the extractive summarization system. Attention mechanism [14,15] has been broadly used in the automatic summarization task and incorporated into neural network models, in which decoders extract important information according to the weighted attention score. Despite their popularity, neural network-based approaches still have some problems when they are applied to summarization tasks. The architectures of the summarizers mostly are variants of recurrent neural networks (RNNs), such as Gated recurrent unit (GRU) and long short-term memory (LSTM). Although they can remember each past decisions within fixed-size state space in theory, practically they can only remember limited document context [16,17]. In addition, salience assessment would be harder at each time step of RNN due to the lack of guidance of the comprehensive document information. Moreover, since there are no explicit alignments between documents and its summary, the weighted score of attention usually contains noisy information and further affects the representation of the local context.

The automatic summarization system is required to hold original text information in a finite vector space, and then reproduce that expression in short form [13]. Therefore, comprehensive encoding representation has been a hot and hard issue in this field [2,18]. Some approaches based on attention mechanism attend to only limited semantic space of sentences rather than comprehensive document information. These end-to-end models attempt to make simple concatenation among forward and backward hidden states, which is hard to integrate relative information of the whole document, resulting in the suboptimal summary. Most of the previous extractive methods focused on treating extractive summarizer as a sequence labeling task. These methods firstly encode sentences in the document and specify whether the sentence should be included in the summary. The process of selecting sentences is constrained by length limitation and relies on the meaning representation [10,13]. However, such methods only estimate the current sentence importance at each time step and ignore the relative importance gain of the sentence selected in the previous steps.

In this work, we designed the Refined self-matching mechanism for extractive summarization (RSME) to overcome the problems mentioned above. In particular, for the first time, the self-matching mechanism is applied to extractive summarization model, which enables the model to attend to global semantic information of the document. In order to effectively simulate human coarse-to-fine reading behavior, the Gaussian focal bias is applied to establish the localness according to signals that come from neighboring words and sentences. We integrate the Gaussian bias into the original self-matching weights. When the RSME aggregates the important information at the global level without regarding the distance barrier, the model can also recognize the semantic information near the current sentence at the local level. Finally, it establishes the long-term dependency and locality for each sentence in the document to help the model extract key information in a comprehensive way and pinpoint the important portions.

Our contributions are as follows:

(1) We propose a refined self-matching mechanism and apply it to the extractive summarization, that dynamically aggregate relative information at the local and global level for each sentence in the document, the localness and long-term dependency are modeled comprehensively.
(2) The Bidirectional Encoder Representation from Transformers (BERT) is incorporated into RSME flexibly. A hierarchical encoder is developed to effectively extracted the information at sentence-level and document-level, which helps capture the hierarchical property of the document.

(3) The pointer network is utilized to select salient sentences based on current extraction state and relative importance gain of previous selections.
(4) Extensive experiments are conducted on the CNN/Daily Mail dataset, and the experimental results showed that the proposed RSME significantly improves the ROUGE score compared with the state-of-art baseline methods.

2. Related Works

Extractive summarization has been widely studied in past researches. These methods manually define features to score sentence saliency and select most important sentences [6,13]. The vast majority of these methods score each sentence independently and then select top-scored sentences to generate a summary, but the process is not included in the learning process.

In recent years, neural network-based methods have been gaining popularity over classical methods, as they perform better in large corpus [13]. The core of neural network model is the structure of encoder and decoder. These models typically utilize convolutional neural networks (CNNs) [19], recurrent neural networks [11,20], or combination of them to create sentence and document representations, input words are represented as word embedding. These vectors are then fed into the decoder and output summary. Summary quality can be heuristically improved by maximum profit, integer linear programming. Yin and Pei et al. [21] used CNNs to map sentences to a continuous vector space, then defined diverseness and prestige to minimize loss functions. Cheng et al. [13] conceptualized extractive summarization as a sequence labeling task, which used a document encoder to score each sentence independently, and an attention-based decoder to label each sentence. Nallaptal et al. [10] proposed an RNN-based model with some interpretable features, such as saliency and content richness of sentence; the model treated the extractive summarization as one sentence classification problem and used a byte decision (0/1) to determine whether the sentence should appear in the summary. Zhou et al. [9] proposed a joint learning model for sentence scoring and selection to lead the two tasks interact simultaneously, and multiple layer perceptron (MLP) is introduced to score sentences according to both the previously selected sentences and remains. Zhang et al. [3] developed a hierarchical convolution model with an attention mechanism to extract keywords and key sentences simultaneously, and a copy mechanism was incorporated to resolve the problem of out of vocabulary (OOV) [22]. In addition, reinforcement learning (RL) has been proven to be effective in improving the performance of the summarization system [12,23] by allowing directly maximize the measure metric of summary quality, such as the ROUGE score between the generated summary and the ground truth. However, the RL-based models still have some problems, such as difficulty in optimization, adjustment and slow training.

It is worth noting that some elements of the RSME framework have been used and introduced in the earlier work [17]. The pointer network combines attention mechanism with a glimpse operation [16] to solve combinatorial optimization problems, as well as point directly to relevant sentences and words in extractive and abstractive summarizers based on previous decisions. The work most similar to us is the HSSAS [18] developed by Sabahi et al. It uses a self-attentive model to construct the hierarchy of the document and scores each sentence based on modeling of abstract features (such as content richness, saliency, and novelty in terms of the entire document). There are two main differences between our work and HSSAS. First, in order to effectively obtain embedded representations of documents and sentences, HSSAS applies a hierarchical attention mechanism to create representations of sentences and documents. However, we take advantage of the nature of CNN and RNN to represent features with different granularities and introduce pre-trained language model BERT [24] to strengthen the document representation. It completes the interaction in three levels, namely, word-sentence, sentence-sentence, and sentence-document. Second, in order to extract sentences with significant information, HSSAS adopts the weighted average of all previous states as additional input to calculate the next state, but we argue that the weighted average would suppresses communication among neighboring words and sentences. However, we develop the refined

self-matching mechanism to model the localness and long term dependency respectively, and integrate them into the propagation process of the neuron, so that the model can complete effective information extraction and weight allocation without any manual feature. The RSME follows the principle: if a sentence is salient, it should carry comprehensive representation of the document.

3. Method

In this section we will describe the RSME in terms of the following details: (I) problem description of extractive method. (II) Hierarchical neural network-based document encoder. (III) Self-matching mechanism. (IV) Localness modeling. (V) Pointer network-based decoder. The overall framework of RSME is shown in Figure 1.

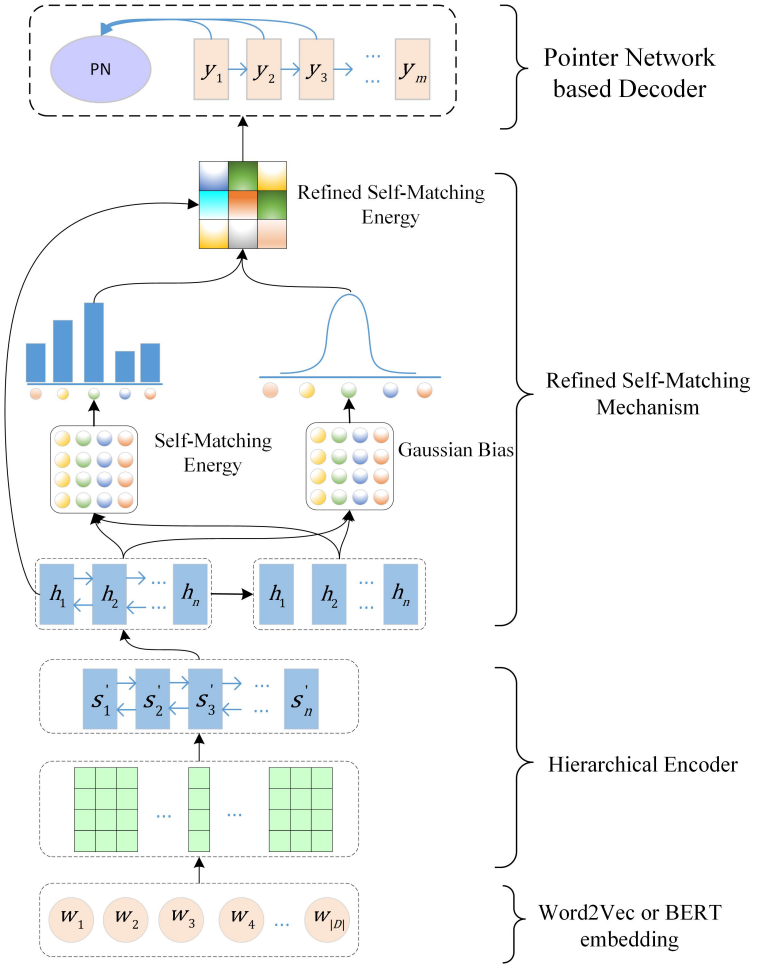

Figure 1. The model architecture of refined self-matching mechanism for extractive summarization (RSME), the embedding layer is responsible for transforming input text into continuous value vector; the hierarchical encoder is composed of convolutional neural networks (CNNs) and long short-term memory (LSTM), which are respectively used to construct state representation of the sentence and document; the self-matching module is responsible for establishing long-term dependency and locality for current sentence representation; the decoder is responsible for selecting sentences based on the current extraction status and the previously selected relative importance gain.

3.1. Problem Description of Extractive Summarizer

Given an input document consists of n sentences $\boldsymbol{D} = \{s_1, s_2, \ldots, s_n\}$, each sentence $s_i = \left\{w_1, w_2, \ldots, w_{|s_i|}\right\}$ contains $|s_i|$ words. The objective of summarization system is to produce a summary $\boldsymbol{Y}$ by selecting m (m < n) sentences from the source document. We predict the label of the ith sentence y_i as (0,1). The labels of sentences in the summary are set as

$y_i = 1$. The goal of training is to learn a set of model parameters θ to maximize probability $p(Y|D,\theta) = \prod_i p(y_i|y_{i-1},\ldots,y_2,y_1,D,\theta) = \prod_i p(y_i|y_{<i},D,\theta)$.

Existing summarizers usually contain the following modules: sentence encoder, document encoder, and decoder. Firstly the sentence encoder will be utilized to encode each word in the sentence and then assemble them into a sequential representation. Secondly, the document encoder contextualizes the sentential representation into document representation. Finally, the decoder selects the sentence according to document meaning representation until reaching the length limit.

3.2. Hierarchical Neural Network Based Encoder

In this work, a hierarchical encoder was developed to capture the sentential representation. For each words in a sentence, their word embedding representations x could be projected from word2vec [25] or BERT [24]. A temporal CNN was exploited to encode all the words in the sentence to obtain the sentential representation; the filter component could map new features within a fixed-size window.

$$x_t = fconv(W_c[w_{t-\frac{k}{2}},\ldots,w_{t+\frac{k}{2}}] + b_c) \tag{1}$$

In which, the $fconv$ is a nonlinear function, $W_c \in \mathbb{R}^{d_w \times k}$ are training parameters, d_w is the embedding dimension, $b_c \in \mathbb{R}^k$ is the bias term, k is the kernel size. The hidden state of the whole sentence can be expressed as:

$$s_t^{'} = (x_1, x_2, \ldots, x_{|s_t|}) \tag{2}$$

when the continuous representation of the sentence is obtained, they are further fed into bi-directional LSTM (BiLSTM), which contains a forward LSTM that reads document from s_1 to s_n as Equation (3), and a backward LSTM that reads document from s_n to s_1 as Equation (4). The BiLSTM captures the temporal dependency among the context. For each sentence s_t, bidirectional hidden space will be concatenated together to denote current state h_t.

$$\overrightarrow{h_t} = LSTM(s_t^{'}, \overrightarrow{h_{t-1}}) \tag{3}$$

$$\overleftarrow{h_t} = LSTM(s_t^{'}, \overleftarrow{h_{t-1}}) \tag{4}$$

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]. \tag{5}$$

Let d_n denotes the number of hidden units of the BiLSTM, n is the number of sentences contained in each document, H_D is the hidden state of the whole LSTM calculated by Equation (6), the dimension of H_D is $\mathbb{R}^{n \times d_n}$.

$$H_D = (h_1, h_2, \ldots, h_n) \tag{6}$$

3.3. Self-Matching Mechanism

The basic requirement of automatic summarizer is that the system can retrieve the sentences with salient information from the whole document semantic space. A large number of LSTM-based methods [10,11] have been proposed to solve the problem: how to model abstractive salience to guide sentence representation. It is well known that if a sentence is salient, the meaning it represents would be retrieved in multiple parts from the document [5]. Although in theory, the RNN-based models are capable to remember all previous decisions, in practice it can only remember limited knowledge of context. For the above observation, we hope to emphasize document-awareness to each word and sentence to guide the salient feature representation, and compensate for the model memory capability in the global context. In past researches, Jiang and Wang et al. [26] established matching relationships between hypotheses and premises word by word in the natural language inference

tasks. The global information generated by the matching result of each word serve as additional input of the LSTM to guide the encoding process.Thus the hidden state of each word is enriched with global information, and improves the original LSTM component 'remember gate' and 'forgotten gate'. Wang et al. [27] proposed a self-matching attention mechanism to deal with the different importance of each word to inference, and applied a gate unit to adaptively control global information from itself or self-matching results.

Inspired by their work, we adapted the self-matching mechanism to the extractive summarizer system. In this work, the self-matching mechanism would match the document against itself, which would dynamically aggregate relevant information from the whole document for each word and sentence, this information is related to the matching degree of each sentence representation with the whole document information. The similar gate unit [27] is employed which is based on current sentence representation and its attention-pooling vector. The matching global information is further merged into the final hidden representation to make the recurrent neural network dynamically incorporates the obtained relative matching information. Intuitively, the document pair can be viewed as the document evidence and the question to be answered. Furthermore, sentence representation containing limited knowledge context is improved, and the sentence representation is expanded.

The dot-product is applied to calculate the matching degree q_t between the semantic representation of each sentence and the whole document, namely, the matching attention-weight matrix. The matching weight α_t would measure the relevance of each sentence to the global document information, then the context c_t can be enriched with the above matching information. A joint representation of context information and hidden state of LSTM are defined as intermediate representation m_t. The share of global information contained in the sentence s_t can be adjusted adaptively by gate unit p_t, in intuition, masking out the irrelevant parts and emphasizing the important ones. Global information glo_t dynamically extracts knowledge from itself or the matched relevant information of document. Finally, glo_t is taken as the extra input of RNN to establish global document awareness for the recurrent neural network.

Formally, giving the current state of sentence h_t, all hidden states of the encoder h^c. Matching attention weight α_t can be calculated:

$$q_{t,i} = h_t \bigodot (h_i^c)^T \tag{7}$$

$$\alpha_{t,i} = \frac{exp(q_{t,i})}{\sum_{j=1}^{n} exp(q_{t,j})} \tag{8}$$

c_t is the context representation based on matched attention:

$$c_t = \sum_{i=1}^{n} \alpha_{t,i} h_t. \tag{9}$$

The final hidden state for sentence s_t can be calculated as:

$$h_t^{'} = LSTM(h_{t-1}^{'}, [h_t; glo_t]) \tag{10}$$

$$glo_t = p_t * h_t + (1 - p_t) * m_t \tag{11}$$

$$p_t = \sigma(W_g[h_t; c_t]) \tag{12}$$

$$m_t = U_g^T tanh(W_m[h_t; c_t]) \tag{13}$$

where W_g and W_m are two learnable weight matrices, σ is the sigmoid activation function.

3.4. Localness Modeling

The self-matching mechanism can establish long-term dependencies for each sentence without concerning distance. Such an operation will disperse the attention distribution and result in overlooking neighboring signals. We argue that the self-matching mechanism may be further enhanced by local information modeling [28]. Conventional self-matching mechanisms focus on all sentences when collecting document information, but some secondary information may confuse the model and lead to suboptimal performance [29]. Moreover, the usage of weighted averages can inhibit the expression of relationships among neighboring words or sentences. In linguistic intuition, if a word x_i is aligned to x_j on the semantic matching, we also hope that words neighboring x_j can be perceived, which can capture phrases patterns or sentence segments that contain more explicit local context. Take the word-level local information as an example. In Figure 2, if "children" is aligned to "make", we hope to pay more attention to nearby "a pray". Eventually, "children" can aggregate the phrase information "make a pray" in the matching process. It is obvious that such neighboring information can contribute to expanding the relationship expression among words and the local context. Similar principles can be extended to sentences and related segments.

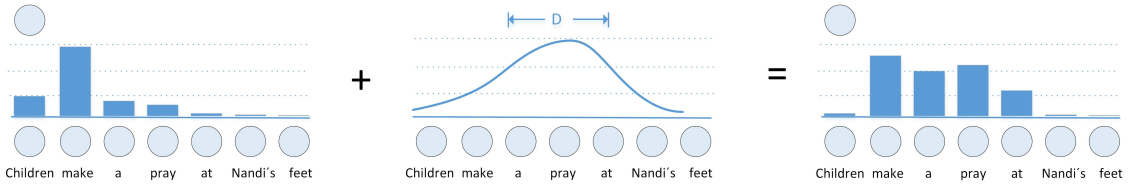

Figure 2. Case for localness modeling.

A learnable Gaussian distribution is utilized to model the local context [30,31] which contains valuable fine-grained information, whose scope is D as Figure 2. Similar to reference [30], the setting of scope size relies on the matching result itself. The Gaussian distribution will serve as the regularization term of attention and added to the original distribution.

$$\alpha_t = softmax(q_t + G) \tag{14}$$

The first term in the equation is the original dotted product distribution, and G is the local Gaussian bias term. The improved mechanism can model both the long term dependency and the localness, better simulating human coarse-to-fine understanding behavior. Since the prediction of each central position and window depend on their corresponding context representation, we apply a feed-forward network to transform q_t into the hidden state of center position and window. The motivation of this design is that the central position and window size interdependently locate local scope, hence condition on the similar hidden state. The center position scalar μ_t and dynamic coverage scalar σ_t are predicted by the matching energy q_t. They can be view as the center and the scope of the locality to be paid attention, intuitively they correspond to "pray" and Gaussian scope "D" in Figure 2. The center position scalar and the coverage scalar are calculated as follows:

$$\mu_t = U_p^T tanh(W_p q_t) \tag{15}$$

$$\sigma_t = U_c^T tanh(W_g q_t) \tag{16}$$

where $W_p \in \mathbb{R}^{d_n \times d_n}$ and $W_g \in \mathbb{R}^{d_n \times d_n}$ are shared parameters, $U_c \in \mathbb{R}^{d_n}$ and $U_p \in \mathbb{R}^{d_n}$ are two different linear projection weighted vectors. μ_t and σ_t are further normalized to the interval $[0, I]$, where I represents the number of input sentences.

$$\begin{pmatrix} \tilde{\mu}_t \\ \tilde{\sigma}_t \end{pmatrix} = I * sigmoid \begin{pmatrix} \mu_t \\ \sigma_t \end{pmatrix} \tag{17}$$

According to the definition of Gaussian distribution, local bias for the t encoding step is calculated with $\tilde{\mu}_t$, $\tilde{\sigma}_t$:

$$G_{t,i} = -\frac{(i - \tilde{\mu}_t)}{\tilde{\sigma}_t{}^2/2} \tag{18}$$

The local bias $G \in (-\infty, 0]$ is added to the original attention distribution is approximate to multiplying by the weight $(0, 1]$ as the exponential operation.

3.5. Sentence Selection Based on Pointer Network

We make sentence selection based on the above encoding representation, and use another LSTM to train pointer network for extracting sentences recurrently. Given the sentence vector $(s_1, s_2, \ldots, s_n)$, and the target sequential indices$(r_1, r_2, \ldots r_m)$, $\forall r_{j,j<n}$. $(e_1, e_2, \ldots e_n)$ and $(d_1, d_2, \ldots d_m)$ denote the hidden state of encoder and decoder respectively. The attention distribution of pointer network at each decoding step t can be calculated as following:

$$u_i^t = v_p^T tanh(W_e e_i + W_d z_t^{'}) \tag{19}$$

$$p(r_t | r_1, r_2, \ldots, r_{(t-1)}) = softmax(u_t) = \frac{exp(u_i^t)}{\sum_{k=1}^{n} exp(u_k^t)} \tag{20}$$

where $z_t^{'}$ represents the result of glimpse operation [32]:

$$d_t = LSTM(d_{t-1}, [W_l l(r_{t-1}); e_t]) \tag{21}$$

$$z_i^t = v_g^T tanh(W_e e_i + W_d d_t) \tag{22}$$

$$att_t = Softmax(z_t) \tag{23}$$

$$z_t^{'} = \sum_i att_i^t W_g e_i \tag{24}$$

where v_g,W_e, W_d are automatically learned scalar weights. $Softmax$ function normalized vector z_j into the attention mask over input. At each decoding step, the pointer network selects one vector with the highest probability from the n input vectors. d_t is the output of the added LSTM-based decoder. Pointer network performs two-hop attention at each time step: firstly, it pays attention to the encoder state e_i to obtain context vector $z_t^{'}$, secondly, it attends to e_i for extraction probabilities. Thus, the pointer network effectively takes the previous decisions as relative importance gain during each decoding step.

4. Experiments

4.1. Dataset

A large corpus is crucial for training deep learning models. The experiments were conducted on the CNN/Daily Mail dataset [10,33] without anonymizing entities or lower case tokens. We used the standard split of CNN/Daily Mail for training, validation, and testing. The dataset contained 287,227 documents for training, 13,362 documents for validation, and 11,490 documents for testing. The average number of sentences in the original document and the human-generated summary was 28 and 3.5, respectively.

4.2. Evaluation Metric

We adapted the commonly used recall-oriented understanding for gisting evaluation (ROUGE) for automatic evaluation, which measured the quality of summary by comparing generated summaries with gold summaries. Three variants of this (the script can be found at: https://github.com/falcondai/

pyrouge): ROUGE-1, ROUGE-2, and ROUGE-L were calculated by matching unigrams, bigrams, and the longest common subsequences (LCS) respectively. In order to compare with most baseline models, the full-length F1 ROUGE is reported.

4.3. Settings

During the training process, the word embedding dimension for context-independent representations was set to 100. The cross-entropy loss was employed. We neither limited the length of sentences nor the maximum number of sentences for per-document. The hidden state dimension of LSTM was set to 300. We used Adam optimizer and the parameters were set to: learning rate = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$. We utilized the gradient clipping to regularize our model, and early stopping based on validation loss. At the test time, we selected sentences based on the predicted probability until they reached the maximum length limit.

4.4. Comparison Baselines

We compared the RSME with strong baselines from previous state-of-art abstractive and extractive summarization systems.

Abstractive **:**

ConvS2S: Gehring et al. [19] innovatively applied the convolutional neural network the sequence to sequence (seq2seq) model and improved on several tasks, including the abstractive summarization.

PGN + Cov: See et al. [15] integrated pointer and coverage mechanism into seq2seq-based abstractive system for solving out of vocabulary (OOV) and repetition problems during generation.

Fast-abs: Chen et al. [23] proposed a novel sentence-level policy gradient method to bridge the sentence selection network and sentence rewriting network in a hierarchical way.

Extractive **:**

Lead-3: The most common baseline model, which selects the lead three sentences in the document as a summary.

HSSAS: Sabahi et al. [18] used the hierarchical structure self-attention mechanism to create sentence and document representation.

Refresh: Narayan et al. [20] directly optimized the evaluation metric ROUGE through a reinforcement learning objective function.

BanditSum: Dong et al. [12] proposed to regard the extractive method as a context bandit problem, and using a policy gradient reinforcement learning algorithm to select the sentences with maximize ROUGE.

SWAP-NET: Jadhav et al. [5] proposed a two-level pointer network architecture for modeling the interaction of keywords and highlighted sentences respectively.

5. Results

5.1. Experimental Results Analysis

The results of the automatic evaluation in Table 1 show that the proposed RSME outperformed those compared abstractive baseline models, although abstractive methods were more faithful to the real summarization task (human-written summary combined information from several crucial pieces of the original document), most abstractive-based models still lagged behind the LEAD-3 in ROUGE. Among the abstractive methods, the Fast-abs proposed by Chen et al. [23] achieved the best performance and was comparable to ours. Interestingly, their system was mostly extractive as they followed the two-step principle that extracting and then rewriting. Consequently, the quality of summary heavily relied on the information of the extracted sentences at the first step.

Table 1. The full-length ROUGE F1 score on the non-anonymized CNN/Daily Mail dataset. Those marked with '#' represent training and evaluation on the anonymous version. All of our ROUGE scores are reported by the official ROUGE script, with a 95% confidence interval of at most ±0.24. The promotion is statistically significant with respect to those strong baselines. Results with * are statistical significantly.

Categories	Model	ROUGE-1	ROUGE-2	ROUGE-L
Abstractive	ConvS2S	39.8	17.3	36.5
	PGN + Cov	39.5	17.3	36.4
	Fastabs	41.4	18.7	37.7
Extractive	LEAD-3	40.3	17.7	36.6
	HSSAS #	**42.3**	17.8	37.6
	Refresh	40.0	18.2	36.6
	BanditSum	41.5	18.7	37.6
	SWAP-NET #	41.6	18.3	**37.7**
Extractive (ours)	RSME	41.5	**18.8**	**37.7**
	BERT-RSME	**42.4** *	**19.8** *	**38.9***

Among the extractive comparisons, the RSME achieved 41.5, 18.8, and 37.7 points respectively in the three ROUGE variants respectively, and the results showed a large promotion by +1.2, +1.1, and +1.1 points compared with LEAD-3. Notably, by modeling abstract feature such as document structure and novelty in the prediction process, HSSAS effectively calculated the respective probability of sentence-summary membership, and achieved a great score of 42.3 on ROUGE-1, leading the RSME by +0.8 points. While the RSME performed better on other metrics, especially on ROUGE-L with a large margin up to +1.0. To some extent, it proves that the proposed RSME was superior in the strategy of capturing document hierarchical property and extracting salient information. Moreover, our model surpassed the complicated reinforcement learning based models Refresh and BanditSum in a simpler method. The SWAP-NET was comparable to ours, and credit should be given to RSME in capturing the effective representation. Our BERT-RSME model consistently outperforms all the strong baselines on three metrics with a large margin.

5.2. Ablation Test

In order to analyze the contribution of different components to the final performance, we performed ablation tests on the RSME. The experimental results are shown in Table 2. When the Gaussian bias component was removed, the performance declined by 0.4 on ROUGE-1, 0.2 on ROUGE-2. When we removed the self-matching component, the performance declineed by a large margin on all three indicators by 0.7, 0.4 and 0.8 respectively. The deviation strongly reveals that global document awareness is crucial to the long document summarization task. The Gaussian bias was used to measure the distance between the predicted center and alignment, which can effectively model the localness and improve the representation of the local context. The improved performance of the two components on ROUGE was 1.1, 0.6, and 0.7, which reveals that combining them to the coarse-to-fine strategy was effective. When the pointer network component was removed, the performance decreased slightly by 0.2, 0.2 and 0.5 on the three indicators respectively, since pointer network can guide the sentence selection with respect to the relative importance gain of the previous selection. Pointer network was efficient in many tasks, and also embodied important value in our work.

Table 2. Ablation test for RSME, "-" means removing the corresponding component based the previous system. In short, "-Pointer Network" represents a model contains only a document encoder and a simple classifier.

Models	ROUGE-1	ROUGE-2	ROUGE-L
RSME	41.5	18.8	37.7
-Gaussian bias	41.1	18.6	37.7
-Self Matching	40.4	18.2	37.0
-Pointer Network	40.2	18.0	36.5

5.3. Discussion

Throughout the ablation testing process, the self-matching mechanism contributed the most to the model, to further explore the influence of self-matching. In this part, we discuss different level document encoding strategies, that is, the impact of document information representation for summarization task. We have implemented the unidirectional LSTM with a simple linear projection based classifier, namely "UniLSTM + Classifier", and the bidirectional LSTM with classifier (as the basic model), namely "BiLSTM + Classifier". The bidirectional LSTM with self-matching mechanism, namely "BiLSTM + Self Matching", to further study how the richness of global information work and affect the document information encoding.

As shown in Table 3, performance on the bidirectional LSTM based model were better than the model with single LSTM, since bidirectional LSTM can encode documents from both forward and backward, that can include more document information. On the contrary, unidirectional LSTM may lose numerous effective features due to memory problems. After the addition of the self-matching mechanism, the performance improved consistently on three indicators, which gives a clear direction to our future work, that is, to improve the global information richness of the document encoding.

Table 3. Analysis of the effect of encoding information richness on experimental results.

Models	ROUGE-1	ROUGE-2	ROUGE-L
UniLSTM + Classifier	39.8	17.9	36.4
BiLSTM + Classifier	40.2	18.0	36.5
BiLSTM + Self Matching	40.9	18.3	37.6

From previous Table 1, it can be seen that BERT significantly improved the overall performance, even surpassed the contribution of any single component. To study whether the strength of pre-trained knowledge would cover the effect of RSME, referring to Figure 1, we select GloVe [34] or BERT in the embedding layer, and remove the refined self-matching layer, and kept the rest unchanged, forming GloVe-basic and BERT-basic. The experimental results in Table 4 show that the proposed RSME has a promising improvement on the baseline of BERT and GloVe.We found that architectures with context-independent GloVe made little contribution to the current models, while models equipped with BERT are improved with a large margin, which explains that RSME is irreplaceable in comprehensive document information extraction.

Table 4. Analysis of pre-trained knowledge on experimental results.

Models	ROUGE-1	ROUGE-2	ROUGE-L
RSME	41.5	18.8	37.7
GloVe-basic	40.4	18.1	36.7
GloVe-RSME	41.4	18.7	37.8
BERT-basic	42.0	19.3	38.5
BERT-RSME	42.4	19.8	38.9

5.4. Case Study

In order to further vividly analyze the proposed RSME and the reasons for performance improvements. We compared the quality of summary generated by the abstractive PGN+cov system, basic extractive system, and RSME with reference summary. In the Table 5, we marked the key information of the reference with a yellow background, and the generated summary with high semantic similarity to the key information were marked in pink. The key information contained in the reference can be abstracted into the following two points: (1) There is a calf called Nandi with five mouths, two of which drink milk, it may be the calf with the most mouths ever seen. Among the RSME-generated summary, the first and second sentences almost contain the information of this part. (2) People are flocking to pray and touch his hooves. The third sentence in the output of RSME corresponds to that. The performance of the abstractive method ConvS2S is the most unsatisfactory. Although three generated sentences with shorter length are more concise than the extractive methods, there are some fatal repetition and factual false problems. Among the sentences selected by our basic extractive model, two have little relevance to the reference and lack of key information. Our RSME can well understand the content of the original text, and the selected three sentences have a high semantic similarity with the reference summary. The above case analysis can reveal that the proposed RSME improves the capability of capturing comprehensive signals of the document. However, there remain difficulties for further research, such as generated sentences are not concise enough.

Table 5. Case study among different systems.

Reference Summary:
bizarre-looking creature can drink through two of his five mouths .
local people in narnaul are flocking to see him and pray at his hooves .
the calf, called Nandi , is thought to have the most mouths of any bovine.
Basic model:
in India's Hindu culture, cows are revered as a symbol of life. Nandi is attracting a constant stream of visitors wanting to celebrate him.
many of them kneel before him to pray and touch his hooves .
baby teeth: one of Nandi's mouths is clearly bigger than the others, which hang around his face.
ConvS2S:
Sukhbir said Nandi was in good health despite health. However, Nandi could only see the side of its body, not the front.
Two years ago, an American farm revealed that it had a calf with two heads
On everyone's lips: this baby calf with five mouths and draws a crowd of tourists .
RSME:
talk about gobby ! this little fellow has been born with five mouths - believed to be the most ever seen on a calf .
the strange-looking baby opens all ten lips when he is sucking at his mother's udders. but he can only take in milk through two of his mouths .
many of them kneel before him to pray and touch his hooves .

6. Conclusions and Future Works

In this work, we propose a novel model for extractive summarization that apply a refined self-matching mechanism to enhance the document encoding representation. The self-matching mechanism helps the model dynamically collect information from the full document without regarding distance barrier, and establish long-term dependence for each sentence. The central position scalar and the coverage scalar are spontaneously determined according to the sentence state. They are utilized to construct the Gaussian bias, which is incorporated into the original matching energy to further contextualize the neighboring signals. Moreover, at each step of sentence selection, the relative importance gain of the previous decisions and the current extraction state are considered. The ROUGE

evaluation on the CNN/Daily Mail dataset shows that the proposed RSME performs better than the recent strong baseline models. Future research effort can be devoted to the combination of the extractive and abstractive method. Such as using the extractive system to select informative sentences, then using the abstractive method to rewrite sentences, hence improve the relevance and conciseness of summary.

Author Contributions: Conceptualization B.Z.; methodology, R.X. and B.Z. ; software, R.X.; validation, W.Z.; writing, R.X.; review, Z.G., W.Z. and H.Y. All authors have read and agreed to the published version of the manuscript

Funding: The research was funded by National Natural Foundation of China, Multimodal Brian-Computer Interface and Its Application in Patients with Consciousness Disorder, Project approval number: 61876067.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. 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; Volume 1: Long Papers, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1095–1104.
2. Gehrmann, S.; Deng, Y.; Rush, A. Bottom-up abstractive summarization. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 4098-4109.
3. Zhang, Y.; Li, D.; Wang, Y.; Xiao, W. Abstract text summarization with a convolutional Seq2seq model. *J. Appl. Sci.* **2019**, *9*, 1665. [CrossRef]
4. Berend, G. Opinion expression mining by exploiting keyphrase extraction. In Proceedings of the 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, 8–13 November 2011; pp. 1162–1170.
5. Jadhav, A.; Rajan, V. Extractive summarization with swap-net: Sentences and words from alternating pointer networks. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics; Volume 1: Long Papers, Melbourne, Australia, 15–20 July 2018; pp. 142–151.
6. Eduard, H.; Lin, C.Y. Automated text summarization and the SUMMARIST system. In Proceedings of the 1998 Workshop on Held at Baltimore, Baltimore, MD, USA, 13–15 October 1998.
7. Erkan, G.; Radev, D.R. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.* **2004**, *22*, 457–479. [CrossRef]
8. Mihalcea, R.; Tarau, P. Textrank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–24 July 2004; pp. 404–411.
9. Zhou, Q.; Yang, N.; Wei, F.; Huang, S.; Zhou, M.; Zhao, T. Neural document summarization by jointly learning to score and select sentences. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics; Volume 1: Long Papers, Melbourne, VI, Australia, 15–20 July 2018; pp. 654–663.
10. Nallapati, R.; Zhai, F.; Zhou, B. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
11. Wu, Y.; Hu, B. Learning to extract coherent summary via deep reinforcement learning. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
12. Dong, Y.; Shen, Y.; Crawford, E.; van Hoof, H.; Cheung, J.C.K. Banditsum: Extractive summarization as a contextual bandit. *arXiv* **2018**, arXiv:1809.09672.
13. Cheng, J.; Lapata, M. Neural summarization by extracting sentences and words. *arXiv* **2016**, arXiv:1603.07252.
14. 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, Lisbon, Portugal, 17–21 September 2015; pp. 379–389.

15. See, A.; Liu, P.J.; Manning, C.D. 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, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1073–1083.
16. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the 2015 International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
17. Vinyals, O., Fortunato, M., Jaitly, N. Pointer networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 2692–2700.
18. Al-Sabahi, K.; Zuping, Z.; Nadher, M. A hierarchical structured self-attentive model for extractive document summarization (HSSAS). *arXiv* **2018** arXiv:1805.07799.
19. 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-Volume 70, Sydney, NSW, Australia, 6–11 August 2017; pp. 1243–1252.
20. Narayan, S.; Cohen, S.B.; Lapata, M. Ranking sentences for extractive summarization with reinforcement Learning. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; Volume 1 (Long Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 1747–1759.
21. Yin, W.; Pei, Y. Optimizing sentence modeling and selection for document summarization. In Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
22. Gu, J.; Lu, Z.; Li, H.; Li, V.O. 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, Berlin, Germany, 7–12 August 2016; pp. 1631–1640.
23. Chen, Y.C.; Bansal, M. Fast abstractive summarization with reinforce-selected sentence rewriting. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics; Volume 1: Long Papers, Melbourne, VI, Australia, 15–20 July 2018; pp. 675–686.
24. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
25. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **201**3, arXiv:1301.3781.
26. Wang, S.; Jiang, J. Learning natural language inference with LSTM. In Proceedings of the 2016 NAACL-HLT, San Diego, CA, USA, 12–17 June 2016; pp. 1442–1451.
27. Wang, W.; Yang, N.; Wei, F.; Chang, B.; Zhou, M. Gated self-matching networks for reading comprehension and question answering. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics; Volume 1: Long Papers, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 189–198.
28. Tan, J.; Wan, X.; Xiao, J. From neural sentence summarization to headline generation: A coarse-to-fine approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*; AAAI Press: Pao Alto, CA, USA, 2017; pp. 4109–4115.
29. Zeng, B.; Yang, H.; Xu, R.; Zhou, W.; Han, X. LCF: A local context focus mechanism for aspect-based sentiment classification. *Appl. Sci.* **2019**, *9*, 3389. [CrossRef]
30. Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-attention with relative position representations. *arXiv* **2018**, arXiv:1803.02155.
31. You, Y.; Jia, W.; Liu, T.; Yang, W. Improving abstractive document summarization with salient information modeling. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 2132–2141.
32. Vinyals, O.; Bengio, S.; Kudlur, M. Order matters: Sequence to sequence for sets. In Proceedings of the 2016 International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.

33. Hermann, K.M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; Blunsom, P. Teaching machines to read and comprehend. In Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1, Montreal, QC, Canada, 11–12 December 2015; pp. 1693–1701.
34. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.

Sample Availability: Samples of the compounds are not available from the authors.