Dual Encoding for Abstractive Text Summarization

Kaichun Yao, Libo Zhang, Dawei Du, Tiejian Luo, Lili Tao, and Yanjun Wu

***Abstract*—Recurrent neural network-based sequence-to-sequence attentional models have proven effective in abstractive text summarization. In this paper, we model abstractive text summarization using a dual encoding model. Different from the previous works only using a single encoder, the proposed method employs a dual encoder including the primary and the secondary encoders. Specifically, the primary encoder conducts coarse encoding in a regular way, while the secondary encoder models the importance of words and generates more fine encoding based on the input raw text and the previously generated output text summarization. The two level encodings are combined and fed into the decoder to generate more diverse summary that can decrease repetition phenomenon for long sequence generation. The experimental results on two challenging datasets (i.e., CNN/DailyMail and DUC 2004) demonstrate that our dual encoding model performs against existing methods.**

***Index Terms*—Abstractive text summarization, dual encoding, primary encoder, recurrent neural network (RNN), secondary encoder.**

I. Introduction

TEXT summarization aims to generate short, accurate, and informative summary from larger text documents. It is widely applied in natural language understanding and information retrieval, etc. Summarization techniques are mainly grouped into *extractive* and *abstractive* approaches. Extractive methods construct a summary by extracting salient words, phrases, or sentences from the source text. Abstractive methods produce a summary similar to a human-written abstract by concisely paraphrasing the source content. That is, the former ensures the grammatical and semantic correctness of the generated summaries, while the latter creates more diverse and novel content. In this paper, we focus on abstractive text summarization.

Recently, neural networks are widely leveraged in many natural language processing tasks because of promising performance [1]. Specifically, the neural networks-based encoder–decoder models are used in the sequence-to-sequence tasks, such as neural machine translation [2], [3], speech recognition [4], [5], image captioning [6], conversational system [7], and text summarization [8]–[11]. Within text summarization tasks, the encoder reads the whole input sequence and generates a fixed dimensional feature vector, and then the decoder uses the feature representation to produce desired output sequence. For example, the summarization model employs a convolutional neural network (CNN) as an encoder and a feed-forward neural network language model as a decoder [8]. An extension to this framework is to add attention mechanism by considering context cues in hidden states of the encoder, which facilitates to decode the target sequence [12].

Although the above encoder–decoder models are promising, some problems still remain. In these works, a decoder uses a fixed target vocabulary to output the corresponding probability distribution at each timestep. This may lead to the incapable to handle rare or out-of-vocabulary (OOV) words. Increasing the size of the target vocabulary could alleviate this problem, but this increases the computational complexity in decoding as a softmax function needs to calculate over all possible words. This can be improved by applying a copy mechanism that dynamically copies the words from the input sequence when decoding without enlarging the size of the vocabulary [10], [13]–[16].

Another problem with encoder–decoder models is that they often generate unnatural summaries consisting of repeated phrases, especially evident for long text summarization generation. On a long sentence summarization dataset (e.g., CNN/DailyMail dataset containing multisentence summaries of up to 56 tokens on average), a coverage mechanism is used to avoid the repetition problem. It records past attentional weights in the decoder and dampens the decoder from attending to the same parts of the input text when decoding in future [13]. Different from this mechanism, an intra-attention mechanism takes the attention at the decoder into account, which is prominently effective for eliminating repetition [17]. However, they consider little about the relations between the input tokens in the encoder and the already generated words by decoder.

To solve these problems, we propose a novel dual encoding for abstractive text summarization (DEATS) that extends the existing sequence-to-sequence framework. Specifically, our dual encoding model consists of a primary encoder, a secondary encoder, and a decoder. It conducts the primary encoder and the decoder as the standard attentional encoder–decoder model. The secondary encoder is based on the input and the previously produced output, and generates a new context vector as an additional input of the decoder. The context vector makes the decoder obtain more meaningful information and generate better output. Besides, we conduct a multistep

Manuscript received January 23, 2018; revised June 24, 2018 and September 17, 2018; accepted October 2, 2018. This work was supported by the National Natural Science Foundation of China under Grant 61807033. This paper was recommended by Associate Editor M. Last. *(Corresponding author: Libo Zhang.)*

K. Yao, D. Du, and T. Luo are with the School of Computer Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China.

L. Zhang and Y. Wu are with the Institute of Software Chinese Academy of Sciences, State Key Laboratory of Computer Science, Beijing 100190, China (e-mail: libo@iscas.ac.cn).

L. Tao is with the Department Engineering, Design and Mathematics, University of the West of England, Bristol BS16 1QY, U.K.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2018.2876317

decoding operation in the decoder, and model the decoded content at each stage as a semantic feature vector. This makes the decoder "remember" the content produced in the earlier time-steps in order to avoid the repetition. The main contributions of this paper are summarized as follows.

1) We propose a dual encoding mechanism (DEM) to extend the traditional sequence-to-sequence model to make full use of the document text information by adding an additional encoder.
2) We consider the importance of words in the input to make the secondary encoder reweigh "remembered" and "forgotten" parts in the input sequence.
3) We introduce an enhanced repetition avoidance mechanism (RAM). It combines an existing coverage mechanism and the previously decoded content produced in the earlier time-steps to improve the repetition problem in sequence generation tasks.
4) We conduct experiments on two challenging datasets (i.e., the CNN/DailyMail dataset and the DUC 2004 dataset), which shows that our dual encoding model outperforms existing models.

The remainder of this paper is organized as follows. Related work on text summarization is reviewed in Section II. The proposed approach is presented in Section III. Experimental evaluations and discussions are given in Sections IV and V, respectively. Concluding remarks are given in Section VI.

II. Related Work

A. Text Summarization

A large majority of work in the past few years has been focused on extractive summarization [18]–[26], where a summary consists of key words or sentences from the source text. Different from extractive methods copying units from the source article directly, abstractive summarization uses the readable language for human to summarize the key information of the original text. Therefore, abstractive approaches can produce much more diverse and richer summaries. Abstractive summarization task has been standardized by the DUC2003 and DUC2004 competitions [27]. Hence, there emerge a series of notable methods without neural networks on this task, e.g., the best performer TOPIARY system [28].

Recently, the emergence of the generative neural models [12] for text has inspired new work in abstractive summarization. A neural network model uses a convolutional encoder to encode the source and attentional feed-forward network to produce a summary [8]. It achieves the state-of-the-art results on the DUC-2004[1] and Gigaword datasets. An extension of this paper uses a similar encoder but replaces the decoder with a recurrent neural network (RNN) [9], and achieves better performance on the both above datasets. Apart from English text summarization, a large dataset for Chinese short text summarization is introduced. The context is used as input of the decoder which is computed as a sum of all hidden states from the encoder [29].

B. Rare or OOV Words Problem

Rare and OOV words prevent models from learning representations for new words during training. This may result in a poor readability for the generated summaries. Although RNN-based encoder–decoder models with attention have shown good performance on many datasets, it is challenging to model rare or OOV words effectively. To handle this problem, a pointer mechanism (PM) is proposed to use a new decoder network to point back to OOV words and phrases in the input text and copy them into the output [30]. Furthermore, an approach combining the PM and the original word generation layer in the decoder considers either of them at each decoding step [13], [27]. Different from the work in [13] and [27], the model with encoder–decoder structure integrates the copying mechanism with word generation in the decoder [15]. Another copying mechanism derives the representations of OOV words from their corresponding context in the input text [16].

C. Repetition Problem

A common issue of neural networks-based encoder–decoder models is that they tend to generate repetitive and incoherent phrases in longer summaries. To avoid this, a coverage mechanism eliminates the repetition by discouraging it from attending to the same part in the input sequence when decoding [14]. The method is adapted from statistical machine translation tasks [31], [32]. A distraction mechanism can be incorporated into the neural networks-based summarization model [33]. All these approaches are devoted to the different forms of information encoding and acquisition at the encoder. On the contrary, the decoded information at the decoder can also be used to avoid the repetition [17]. Our dual encoding model is similar to [16] at the encoder, but conducts a different secondary encoding every several decoding steps. At the same time, at the decoder, the decoded content is modeled as a feature representation and then the secondary encoder uses it to fulfil a secondary encoding.

The repetition problem happens more often in long sequence generation tasks. However, researchers pay little attention to large-scale datasets for summarization of longer text. In [10], the RNN-based encoder–decoder model with hierarchical attention is proposed for abstractive summarization task and evaluated on the CNN/DailyMail dataset [34]. Later, another hierarchical RNN model is developed and achieves significantly better abstractive result with respect to the ROUGE metric [35]. Our dual encoding model is mainly designed for long sequence generation tasks, therefore, we also use the CNN/DailyMail dataset to evaluate the proposed method.

III. Dual Encoding Model

Abstractive text summarization can be formulated as a generation task that a "output sequence" is generated from a "input sequence." The input is a source text sequence $X = [x_1, x_2, \ldots, x_j, \ldots, x_m]$, where j and m are the index and the number of the words in source text, respectively. The output is a shorter summary sequence $Y = [y_1, y_2, \ldots, y_i, \ldots, y_n]$ of that text, where i and n are the index and the number of the words in summary text, respectively.

[1] http://duc.nist.gov/duc2004/tasks.html

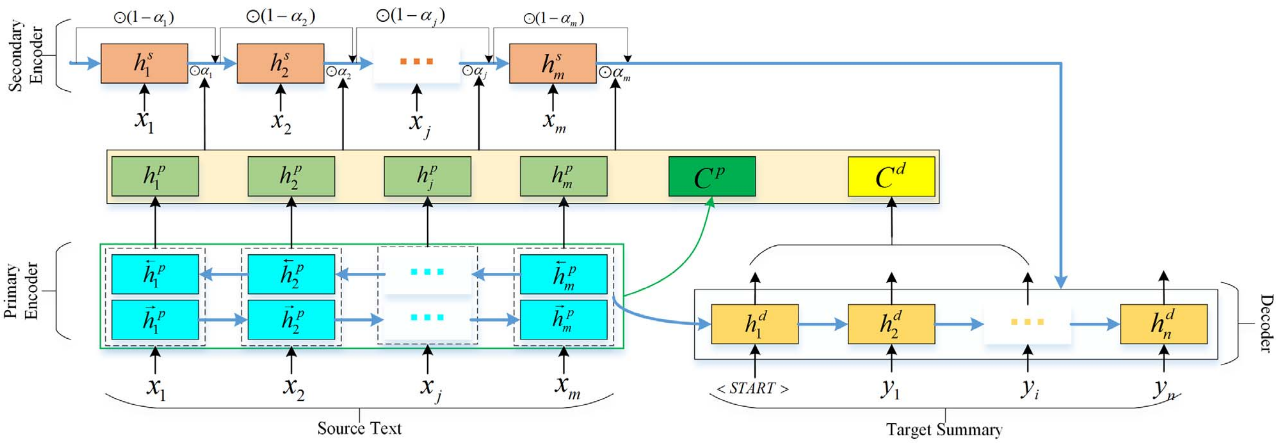

Fig. 1. Overview of our dual encoding model.

In this section, we describe the dual encoding model in detail. As depicted in Fig. 1, our dual encoding model consists of a primary encoder, a secondary encoder, and a decoder equipped with an attention mechanism.

1) The primary encoder calculates the semantic vectors for each word in input sequence.
2) The secondary encoder first calculates the importance weight for each word in input sequence and then recalculates the corresponding semantic vectors.
3) The decoder with attention mechanism decodes by stages and generates a partial fixed-length output sequence at each stage.

Notably, all the above three modules employ the gated recurrent unit (GRU) [36].

For each iteration, the primary encoder first reads each word in the input sequence, and produces the corresponding hidden representation (i.e., h_j^p in Fig. 1) and the content representation (i.e., C^p in Fig. 1) for the whole source text.

Second, the decoder generates the partial fixed-length sequence for every K decoding steps which is modeled as the current decoded content representation (i.e., C^d in Fig. 1), where K is defined in (9). Based on the above representations (h_j^p, C^p, and C^d), the importance weight (i.e., α_j in Fig. 1) is calculated with (4).

Third, the secondary encoder fulfils more fine encoding on the input sequence for every K decoding steps. Specially, we put the α_j on the secondary encoder in the form of skip-connections. Then, the secondary encoding is conducted on the source text followed in (5). It is worth mentioning that the new encoding generates a new semantic context vector (i.e., h_m^s in Fig. 1) to facilitate the decoder to output more accurate target sequence. When the decoder finishes decoding the next fixed-length (K decoding steps) subsequence, the secondary encoder conducts the secondary encoding once again.

In our dual encoding model, the secondary encoder conducts an encoding operation based on the input and already output at each stage. Therefore, it is of great significance for the quality of the previously texts generated by the decoder. The better previous output would significantly facilitate the latter text generation. We use the PM and the coverage mechanism to guarantee the decoder to obtain a better partial output. A combination of the DEM, the PM and the coverage mechanism could make them benefit from each other.

The details of the primary and the secondary encoder are described in Sections III-A and III-B, respectively. Different from the general decoder, our decoder conducts the decoding operation by stages, which is described in Section III-C. We use the PM to handle rare or OOV words and an enhanced RAM to discourage the repetition problem, which are given in Sections III-D and III-E, respectively. Besides, the training process for our dual encoding model is given in Algorithm 1.

A. Primary Encoder

RNN has achieved promising performance in sequence processing tasks, especially in handling a variable-length sequence [37]. Moreover, RNN with the gating units is more easily trained than vanilla RNN with better performance in many tasks [38]. Therefore, we employ the primary encoder to generate coarse encoding using a GRU-based RNN [38]. A GRU can adaptively capture dependencies of different time scales, which is defined as the following equations:

$$\begin{cases} u_t = \sigma\big(W_u[x_t, h_{t-1}]\big) \\ r_t = \sigma\big(W_r[x_t, h_{t-1}]\big) \\ h_t' = \tanh\big(W_h[x_t, r_t \odot h_{t-1})]\big) \\ h_t = (1-u_t) \odot h_{t-1} + u_t \odot h_t' \end{cases} \tag{1}$$

where W_u, W_r, and W_h are parameter matrices. x_t and h_t indicate the corresponding input embedding vector and the hidden state vector at the time step t, and $\odot$ is an element-wise multiplication operator.

The purpose of the primary encoder is to construct the feature representation of the input sentence. Here, we employ a bidirectional GRU (Bi-GRU) as the recurrent unit of the primary encoder, as shown in the bottom-left of Fig. 1. The Bi-GRU consists of a forward and a backward GRU. Given a sequence of the input word embeddings [i.e., (x_1, x_2, x_j, x_m) in Fig. 1], the forward GRU computes hidden state representations $(\vec{h}_1^p, \vec{h}_2^p, \ldots, \vec{h}_j^p, \ldots \vec{h}_m^p)$ at each word position sequentially according to the current word embedding and the previous hidden state. The backward GRU generates hidden

Algorithm 1 Training Process for Dual Encoding Model

```
Given training set $< X, Y >$
for $epsiode = 0, M$ do
    Sample $(x,y)$ from source text $X$ and gold summary $Y$
    Compute the hidden state of primary encoder $h_t^p$ for each word in $x$ Eq. (1) and Eq. (2)
    Compute the content representation $C^p$ for $x$ using Eq. (3)
    for decoding time-step $i = 0, len(Y)$ do
        Compute the hidden state of decoder $h_i^d$ using Eq. (7)
        if $i\%K == 0$ then
            if $i == 0$ then
                Set the content representation of partial generated sequence $C^d$ to zero
            else
                Compute $C^d$ using Eq. (8)
            end if
            Compute the importance weight $\alpha_t$ using Eq. (4)
            Compute the hidden state of secondary encoder $h_t^s$ using Eq. (5)
            Compute the hidden state of decoder $h_i^d$ based on $h_{i-1}^d$ and $h_m^s$ in Eq. (9)
        end if
        Compute the vocabulary distribution $P_w$ using Eq. (12)
        Update network parameters based on the overall loss $\mathcal{L}$ in Eq. (16)
    end for
end for
```

state representations $(\overleftarrow{h}_1^p, \overleftarrow{h}_2^p, \ldots, \overleftarrow{h}_j^p, \ldots, \overleftarrow{h}_m^p)$ for each word in *reversed* sequence (i.e., from the last word to the first). The two kinds of hidden states are defined as

$$\begin{cases} \overrightarrow{h}_t^p = \mathrm{GRU}^p\left(x_t, \overrightarrow{h}_{t-1}^p\right) \\ \overleftarrow{h}_t^p = \mathrm{GRU}^p\left(x_t, \overleftarrow{h}_{t-1}^p\right). \end{cases} \tag{2}$$

We set the initial states of the Bi-GRU to zero vectors, i.e., $\overrightarrow{h}_1^p = \mathbf{0}$ and $\overleftarrow{h}_m^p = \mathbf{0}$. After the input sequence is read by the primary encoder, each word in the sequence can be represented as a concatenated hidden state of forward GRU and backward GRU, denoted as $h_t^p = [\overrightarrow{h}_t^p, \overleftarrow{h}_t^p]$ (i.e., $h_1^p, h_2^p, h_j^p, h_m^p$ in Fig. 1). Then, we can model the representation of the whole input text sequence as a nonlinear transformation of the average pooling of the concatenated hidden states of Bi-GRU. The representation C^p is calculated as

$$C^p = \tanh\left(W_p \frac{1}{N} \sum_{t=1}^{N} h_t^p + b_p\right) \tag{3}$$

where W_p and b_p are parameters, and N represents the length of the input sequence.

B. Secondary Encoder

The secondary encoder is depicted in the top of Fig. 1. As discussed above, the primary encoder reads the input sequence only once to create the hidden state representations. It computes the context with attention mechanism at each decoding time step adaptively. Different from the primary encoder, the secondary encoder is built with unidirectional GRU RNN, and reads the input sequence every K decoding steps according to the decoded information at each stage. At the same time, the importance weight α_t is computed based on the feature representation of each word h_t^p in the input sequence, the content of entire input text sequence C^p and the content representation of output sequence C^d generated by decoder at the current stage. We have

$$\begin{aligned} \alpha_t = \sigma\Big(W_2\Big(\tanh\Big(W_1\Big[h_t^p, C^p, C^d\Big] + b_1\Big)\Big) + h_t^{pT} W_s C^p \\ + h_t^{pT} W_s C^d - C^{pT} W_r C^d + b_2 \end{aligned} \tag{4}$$

where W_1, W_2, W_s, W_r, b_1, and b_2 are the learning parameters. The importance weight α_t signifies how much attention should be paid to the current input word x_t. For the summarization task, the saliency between every word and the entire content of source text is modeled as $h_t^{pT} W_s C^p$ and $h_t^{pT} W_s C^d$ in (4). The redundancy between the content of source text and the decoded content in current stage is modeled as $C^{pT} W_r C^d$ in (4). Finally, α_t is computed for each word in the input sequence based on the information itself, its saliency and the redundancy.

As shown in Fig. 1, we put the importance weight α_t on the skip-connections to bias the two information flows. That is, if the current input word x_t has a very small weight α_t, then the hidden state h_t^s encoded by the secondary encoder will take the majority of information directly from the previous hidden state h_{t-1}^s, neglecting the effect of the current word. If α_t approximates to 1, it is similar to a standard GRU, which is only influenced from the current word. Thus, the secondary encoder has the following update rule:

$$h_t^s = (1 - \alpha_t) \odot h_{t-1}^s + \alpha_t \odot \mathrm{GRU}^s\left(x_t, h_{t-1}^s\right). \tag{5}$$

Notably, the final hidden state h_m^s is the complementary information to help decoder generate target summary. Thus, both the secondary encoder and the primary encoder together complete our dual encoding process.

C. Decoding by Stages

As shown in the right of Fig. 1, we also use GRU as the decoder to generate the output summary. The decoder and primary encoder constitute a basic sequence-to-sequence model. Moreover, some advanced techniques, such as attention mechanism [12]; copy mechanism [15], [16]; pointer-generator network [10], [13], [14]; and coverage mechanism [14], [31] can be applied in the basic sequence-to-sequence model to achieve better performance. In the sequence generation task, the secondary encoder is used as a complementary and independent encoder to improve the performance of our basic model. In this paper, we use a decoder with attention mechanism to compute the context vector according to the hidden states $(h_1^p, h_2^p, \ldots, h_j^p, \ldots, h_m^p)$ of the primary encoder. The context vector c_i is computed as a weighted sum of these hidden states as

$$c_i = \sum_{j=1}^{n} a_{ij} h_j^p \tag{6}$$

where the weight a_{ij} of each hidden state h_j^p is computed by

$$\begin{cases} a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \\ e_{ij} = v_a^T \tanh\left(W_a h_{i-1}^d + U_a h_j^p\right) \\ h_i^d = \mathrm{GRU}^d\left(y_i, h_{i-1}^d\right). \end{cases} \tag{7}$$

Score e_{ij} represents how well the inputs around position j match the output at position i, and h_i^d is the hidden state generated by the decoder that is based on its last hidden state h_{i-1}^d and the ith target y_i in the output sequence.

Our dual encoding model does not decode the whole output sequence at one time but decodes the partial fixed-length sequence by stages. We decode the partial sequence for the fixed length K at each stage and model the whole decoded sequence as

$$C^d = \tanh\left(W_d \frac{1}{L} \sum_{i=1}^{L} h_i^p + b_d\right) \tag{8}$$

where W_d and b_d are parameters, and L denotes the length of the current decoded sequence. C^d is the current content produced by the decoder, which is used to adjust the attention weight of the secondary encoder to each word in the input sequence, and we set C^d to a zero vector at the beginning of decoding. After every fixed-length decoding, the secondary encoder generate a new final state h_m^s, and our decoder is rewritten as follows:

$$h_i^d = \begin{cases} \mathrm{GRU}^d\left(y_i, \left[h_{i-1}^d, h_m^s\right]\right) & \text{if } L \,\%\, K == 0 \\ \mathrm{GRU}^d\left(y_i, h_{i-1}^d\right) & \text{otherwise.} \end{cases} \tag{9}$$

The initial state of the decoder is set to the final state of the primary encoder, namely $h_0^d = h_m^p$. We compute the decoded content and the secondary encoding at every K decoding steps. Then, we concatenate the current context vector c_i acquired from the primary encoder and the decoder hidden state h_i^d, and feed through one linear layer to produce the vocabulary distribution as

$$P_v = P(y_i | y_1, \ldots, y_{i-1}; x) = \mathrm{softmax}\left(W_v \left[h_i^d, c_i\right] + b_v\right) \tag{10}$$

where $P(y_i | y_1, \ldots, y_{i-1}; x)$ is the conditional probability distribution for the target word y_i over all words in the vocabulary at time-step i. W_v and b_v are the learning parameters.

D. Pointer Mechanism

Some rare words or OOV words such as named-entities are central to the summary, but they prevent models from learning representations for new words when training. It is commonly dealt with the use of an universal "UNK" token for words representation, but resulting in a poor readability for the generated summaries. In summarization tasks, an intuitive way to handle such OOV words is to simply point to their location in the source document. Inspired by [14] and [30], we use a PM between the primary encoder and the decoder in our dual encoding model. We allow copying words via pointing, along with generating words from a fixed vocabulary. A soft switch P_p is used to choose between generating a word from the fixed vocabulary by sampling from P_v, and copying a word from the input sequence by sampling from the attention distribution a_i. P_p is a generation probability for time-step i, which is calculated as

$$P_p = \sigma\left(w_c^T c_i + w_h^T h_i^d + w_y^T y_i + w_d^T C^d + b_g\right) \tag{11}$$

where w_c, w_h, w_y, w_d, and b_g are the learning parameters. c_i is the context vector, h_i^d is the decoder hidden state, y_i is the decoder input, and C^d is the content representation of partial decoded sequence. σ is the sigmoid function, hence, $P_p \in [0, 1]$.

For each document, we use an extended vocabulary to denote the union of the fixed vocabulary and all words appearing in the source document. The probability distribution over the extended vocabulary is calculated as

$$P_w = P_p P_v(w) + \left(1 - P_p\right) \sum_{j: w_j = w} a_{ij}. \tag{12}$$

Note that if w is an OOV word, then $P_v(w)$ is zero. Similarly, if w does not appear in the source document, then $\sum_{j: w_j = w} a_{ij}$ is also zero. The PM is more robust in dealing with rare words as it uses the hidden-state representation of rare words from the encoder to decide which word from the source document to point to. The model is still able to accurately point to unseen words which do not appear in the target vocabulary, because the hidden state depends on the entire context of the word.

During training, the loss for time-step i is the negative log likelihood of the target word w_i, i.e., $\mathcal{L}_i = -\log P(w_i)$. Therefore, the overall loss for the whole sequence is

$$\mathcal{L} = \frac{1}{T} \sum_{i=0}^{T} \mathcal{L}_i \tag{13}$$

where T denotes the target sequence length.

E. Repetition Avoidance

For sequence-to-sequence models, repetition is a common problem in sequence generation tasks, especially notable in generating multisentence text. In our dual encoding model, an enhanced mechanism is used to solve this problem. On one hand, the secondary encoder generates an encoding feature vector every K steps, which makes the decoder remembers the content produced in the earlier time-steps to avoid the repetition. On the other hand, we use the coverage mechanism [14], in which the coverage vector c^v is defined to the sum of attention distributions over all previous decoder time-steps

$$c_i^v = \sum_{i'=0}^{i-1} a_{i'}. \tag{14}$$

Note that c_0^v is a zero vector, because none of the source document has been covered on the first decoding time-step. Next, the coverage vector is also used as extra input to the attention mechanism in (7). Hence, the formula for attention mechanism is updated to

$$e_{ij} = v_a^T \tanh\Big(W_a h_{i-1}^d + U_a h_j^p + W_{ch} c_i^v\Big). \tag{15}$$

At the same time, an additional coverage loss is defined to penalize repeatedly attending to the same locations. Combining (13), the primary loss function is rewritten as

$$\mathcal{L} = \frac{1}{T} \sum_{i=0}^{T} \left(\mathcal{L}_i + \lambda \sum_j \min\Big(a_{ij}, c_{ij}^v\Big) \right) \tag{16}$$

where λ is a hyper parameter. i and j denote the decoding time-step and the position in input sequence, respectively. The coverage mechanism aims to deal with the repetition problems from the encoder by discouraging the decoder from attending to the same part on the input sequence according to the past attentional weights. Combined the coverage mechanism with the content produced in the earlier time-steps by the decoder, the enhanced mechanism in this paper can be regarded as avoiding the repetition from both the encoder and the decoder.

IV. Experimental Results

In this section, the datasets for evaluation and the evaluation metric are introduced first, and then the proposed method is compared against with the state-of-the-art methods in the challenging datasets. The experimental results of other state-of-the-art methods are provided from the authors or reproduced from the available source codes. The proposed method is implemented using Tensorflow.[2] It takes about two weeks to train our model until the model converges on a machine with a 3.4 GHz Intel *i*7 processor, 32 GB memory, and a NVIDIA GTX 1080 GPU card with 8 GB memory.

A. Datasets

For our experiments, we train and test our dual encoding model on the joint CNN/DailyMail dataset, namely a multiple sentences summarization dataset. The dataset is originally constructed for the question answering task [34], and remodified for abstractive summarization task [10]. On average, there are 28 sentences per document in the training set, and an average of 3 $\sim$ 4 sentences in the reference summaries. Overall, the dataset contains 286 817 in training set, 13 368 in validation set, and 11 487 examples in testing set. Besides, there are on average 781 and 56 tokens in the input articles and the output summaries, respectively.

We also use the DUC 2004 corpus as a testing dataset to evaluate our model. It contains 500 documents and their corresponding summaries, where each document has four different human-written reference summaries. In this paper, we test our dual encoding model on this dataset which is trained on the CNN/DailyMail dataset, and we limit the length of every summary to 30 words since the official evaluation on it is based on limited-length Rouge recall.

B. Experimental Settings

In our experiments, the batch size is set to 32 when training our model. We set the dimension of hidden states of both encoders and decoder to 256. We limit the size of vocabulary to 50 000 by selecting the most frequent tokens in the training set. OOV words are represented as token <UNK>. Similar to the settings in the work [14], we do not pretrain the word embeddings, but learn themselves from scratch during training. The dimension of word embeddings are set to 128. The network parameters are randomly initialized over a uniform distribution within $[-0.05, 0.05]$, and optimized using Adagrad [39] algorithm. The learning rate and an initial accumulator value is set to 0.15 and 0.1, respectively. We clip gradient with the maximum gradient norm of 5. In addition, we set the decoding length to 20 for the CNN/DailyMail dataset with long summary, and 10 for the DUC 2004 dataset with relatively short summary.

For speeding up training, we truncate the input sequences to 400 tokens and restrict the length of summaries to 100 tokens on the CNN/DailyMail dataset. At the testing time, we also use the same length settings, and decode the output summaries using beam search with beam size 4.

C. Compared Methods

In this section, we compare the proposed dual encoding model (DEATS) method with the following state-of-the-art methods on the CNN/DailyMail dataset and the DUC 2004 test dataset.

1) DUC 2004 Dataset: We compare the performance of our dual encoding model with the following models on the DUC 2004 dataset.

1) *TOPIARY* [28] using a combination of linguistically motivated compression methods and an unsupervised topic detection algorithm, which is the best performer on the dataset.
2) *ABS* [8] with a local attention-based mechanism to generate each word of the summary conditioned on the input sentence.
3) *ABS+* [8] combining conventional ABS combined and an additional log-linear extractive summarization model with hand-crafted features.
4) *RAS-Elman* [9] using an attentive encoder and RNN-based decoder.
5) *words-lvt5k-lsent* [10] is an attentional encoder–decoder model with the large vocabulary trick.
6) *SEASS* [40] is a selective encoding model with a selective gate network to construct a second level sentence

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

representation by controlling the information flow from encoder to decoder.

2) CNN/DailyMail Dataset: On this joint dataset, we compare the performance of our dual encoding model with the following approaches.

1) *seq2seq+atten* [12] is a standard sequence-to-sequence model with attention mechanism employed for abstractive text summarization.
2) *words-lvt2k-temp-att* [10] is an abstractive encoder–decoder-based model using the temporal attention mechanism from [41] that keeps track of past attentional weights of the decoder and restrains the repetitive parts in the later sequence.
3) *SummaRuNNer-abs* [25] is an RNN-based sequence model for abstractive summarization and is converted from an extractive model by using a novel training mechanism.
4) *pointer-generator* [14] is a standard sequence-to-sequence attentional model-based hybrid pointer-generator network to deal with rare or OOV words problem.
5) *pointer-generator+coverage* [14] is improved from "pointer-generator" model by adding a coverage mechanism to discourage the repetition, denoted as "pg+cg" in Table I.
6) *RL+ML* [17] is a neural network model with intra-attention and a new training approach for abstractive summarization.

Notably, "RL+ML" approach [17] combines maximum-likelihood training and reinforcement training. Likewise, the PM, similar to pointer-generator, is also used in their model. Different from the hybrid training method in their work, all above approaches including ours are the standard supervised sequence prediction model using maximum-likelihood training.

D. Evaluation

We evaluate our model using ROUGE metric [42]. ROUGE measures the quality of summary by computing the number of overlapping lexical units. In this paper, we use the scores from Rouge-1, Rouge-2, and Rouge-L, which, respectively, measure the matches of unigrams, bigrams, and longest common subsequences between the generated summaries and the reference summaries. In our experiments, we randomly select 100 test examples to evaluate the summary quality. The ROUGE scores are obtained using the pyrouge package.[3]

E. Results on CNN/DailyMail Corpus

We report the experimental results of various models on the CNN/DailyMail testing set in Table I. From the results shown in Table I, our dual encoding model achieves the state-of-the-art performance. This is attributed to the following factors. First, compared with the pg+cg approach only using a coverage mechanism to discourage the repetition, "DEATS" uses an enhanced repetition avoid mechanism which combines the coverage mechanism and the previously generated output by decoder to improve the quality of the generated summary. Second, the compared methods generate the complete target summary at once and just conduct one encoding process on the input sequence; while our DEATS method adopts a DEM and multisteps decoding operation. Specifically, the secondary encoding in DEM is more likely to fulfil a fine and selective encoding based on the input and the previous output that tends to help decoder produce better summary.

TABLE I
PERFORMANCE COMPARISON OF VARIOUS MODELS ON THE CNN/DAILYMAIL TESTING SET USING ROUGE F1 SCORE

Method	Rouge-1	Rouge-2	Rouge-L
seq2seq+atten	31.34	11.79	28.10
words-lvt2k-temp-att	35.46	13.30	32.65
SummaRuNNer-abs	37.50	14.50	33.40
pointer-generator	36.44	15.66	33.42
RL+ML	39.87	15.82	36.90
pg+cg	39.53	17.28	36.38
DEATS	**40.85**	**18.08**	**37.13**

TABLE II
PERFORMANCE COMPARISON OF VARIOUS MODELS ON THE DUC 2004 TESTING SET USING ROUGE RECALL SCORE

Method	Rouge-1	Rouge-2	Rouge-L
TOPIARY	25.12	6.46	20.12
ABS	26.55	7.06	22.05
ABS+	28.18	8.49	23.81
RAS-Elman	28.97	8.26	24.06
words-lvt5k-lsent	28.61	9.42	25.24
SEASS	29.21	9.56	25.51
DEATS	**29.91**	**9.61**	**25.95**

TABLE III
PERFORMANCE COMPARISON OF OUR DUAL ENCODING MODELS FOR DIFFERENT DECODING LENGTHS ON THE CNN/DAILYMAIL TESTING SET USING ROUGE F1 SCORE

DEATS variants	Rouge-1	Rouge-2	Rouge-L
DEATS-10	38.26	16.44	34.90
DEATS-15	40.35	17.57	36.72
DEATS-20	**40.85**	18.08	37.13
DEATS-25	40.62	17.37	**37.37**
DEATS-30	40.71	**18.18**	37.23
DEATS-40	40.76	17.99	37.20
DEATS-50	40.75	17.52	37.16
DEATS-100	39.85	17.45	36.58

F. Results on DUC 2004 Corpus

We also test DEATS on the out-of-domain DUC 2004 dataset which is trained on the CNN/DailyMail dataset. Evaluation of our DEATS method uses the limited-length Rouge Recall at 75 bytes. According to the results in Table II, our method achieves the best performance. It is worth mentioning that DEATS just achieves slightly better performance than "SEASS" (e.g., 29.91 versus 29.21 for Rouge-1). On the other hand, in terms of the CNN/DailyMail dataset with relatively

[3] http://pypi.python.org/pypi/pyrouge/0.1.3

TABLE IV

EXAMPLES GENERATED FROM THE DEATS METHOD ON THE CNN/DAILYMAIL TESTING SET UNDER THE DIFFERENT DECODING LENGTHS AND THE ROUGE SCORES CORRESPOND TO THE SPECIFIC EXAMPLE AND *K* DENOTES THE DECODING LENGTH

Source Document 1:
a duke student has admitted to hanging a noose made of rope from a tree near a student union , university officials said thursday . the prestigious private school did n't identify the student , citing federal privacy laws . in a news release , it said the student was no longer on campus and will face student conduct review . the student was identified during an investigation by campus police and the office of student affairs and admitted to placing the noose on the tree early wednesday , the university said . officials are still trying to determine if other people were involved . criminal investigations into the incident are ongoing as well . students and faculty members marched wednesday afternoon chanting " we are not afraid . we stand together , " after pictures of the noose were passed around on social media .... in the news release , the vice president for student affairs called the noose incident a " cowardly act . " " to whomever committed this hateful and stupid act , i just want to say that if your intent was to create fear , it will have the opposite effect , " larry moneta said wednesday . duke university is a private college with about 15,000 students in durham , north carolina.
Ground Truth Summary:
student is no longer on duke university campus and will face disciplinary review. school officials identified student during investigation and the person admitted to hanging the noose , duke says. the noose , made of rope , was discovered on campus about 2 a.m.
DEATS, $K = 10$ (**Rouge-1:**50.00, **Rouge-2:**16.22, **Rouge-L:**47.37)
duke student has admitted to hanging a noose made of rope from a tree near student union. the student was identified during an investigation by campus police and the office of student affairs.
DEATS, $K = 20$ (**Rouge-1:**52.87, **Rouge-2:**16.47, **Rouge-L:**48.28)
duke student has admitted to hanging a noose made of rope from a tree near a student union. the student was identified during an investigation by campus police and the office of student affairs and admitted to placing the noose on the tree early wednesday.
DEATS, $K = 30$ (**Rouge-1:**47.19, **Rouge-2:**13.79, **Rouge-L:**40.45)
duke student has admitted to hanging a noose made of rope from a tree near a student union. the student was identified during an investigation by campus police and the office of student affairs. the incident is one of several recent racist events to affect college students.
Source Document 2:
this time , it 's official : russia expects north korean leader kim jong un to visit moscow next month for world war ii anniversary celebrations . north korean representatives have confirmed that kim will be in the russian capital for may 9 victory day celebrations , russian presidential aide yuri ushakov said wednesday , according to russian state-run news agency tass . this would mark kim 's first official foreign trip since inheriting the leadership of north korea in late 2011 . kim will meet with russian president vladimir putin as part of the may visit , tass reported . kim 's trip has been anticipated since late december , when russian state media reported that moscow had extended an invitation to pyongyang. ...... this year 's victory day marks the 70th anniversary of the soviet union 's victory over nazi germany in world war ii . russia has said it has invited more than 60 world leaders to the celebrations . kim expected to visit moscow as north korea , russia foster warmer relations . cnn 's madison park and alla eshchenko contributed to this report.
Ground Truth Summary:
a russian presidential aide says kim will be in moscow for may 9 victory day celebrations , news agency reports. this victory day marks the 70 years since the soviet victory over germany in world war ii.
DEATS, $K = 10$ (**Rouge-1:**43.84, **Rouge-2:**28.17, **Rouge-L:**38.36)
russia expects north korean leader kim jong un to visit moscow next month for world war ii anniversary. north korean representatives have confirmed that kim will be in the russian capital for may 9 victory day celebrations.
DEATS, $K = 20$ (**Rouge-1:**43.24, **Rouge-2:**27.78, **Rouge-L:**37.84)
russia expects north korean leader kim jong un to visit moscow next month for world war ii anniversary celebrations. north korean representatives have confirmed that kim will be in the russian capital for may 9 victory day celebrations.
DEATS, $K = 30$ (**Rouge-1:**38.64, **Rouge-2:**23.26, **Rouge-L:**34.09)
russia expects north korean leader kim jong un to visit moscow next month for world war ii anniversary celebrations. north korean representatives have confirmed that kim will be in the russian capital for may 9 victory day celebrations. kim will meet with russian president vladimir putin as part of the may visit.
Source Document 3:
a 32-year-old massachusetts man is facing murder charges , authorities said wednesday , four days after another man 's remains were found in a duffel bag . the middlesex district attorney 's office said that carlos colina , 32 , will be arraigned the morning of april 14 for murder in connection with the remains discovered saturday in cambridge . earlier this week , colina was arraigned on charges of assault and battery causing serious bodily injury and improper disposal of a body . a middlesex county judge then revoked bail for colina in another case he 's involved in , for alleged assault and battery . the victim in that case is different from the one whose remains were found in recent days . police were notified saturday morning about a suspicious item along a walkway in cambridge. ...... that location is near the cambridge police department headquarters . the remains at both locations belonged to the same victim , identified monday as jonathan camilien , 26 . camilien and colina knew each other , according to authorities . " this was a gruesome discovery , " district attorney marian ryan said . cnn 's kevin conlon contributed to this report.
Ground Truth Summary:
prosecutor : carlos colina , 32 , will be arraigned on the murder charge next week. he 's already been arraigned for alleged assault and battery, improper disposal of a body. body parts were found in a duffel bag and a common area of an apartment building.
DEATS, $K = 10$ (**Rouge-1:**60.27, **Rouge-2:**42.25, **Rouge-L:**54.80)
carlos colina , 32 , will be arraigned for murder charge on the morning of april 14. colina was arraigned for alleged assault and battery and improper disposal of a body .
DEATS, $K = 20$ (**Rouge-1:**59.77, **Rouge-2:**42.35, **Rouge-L:**55.17)
carlos colina , 32 , will be arraigned for murder charge on the morning of april 14 . colina was arraigned for alleged assault and battery and improper disposal of a body. the victim is different from the one whose remains were found in recent days .
DEATS, $K = 30$ (**Rouge-1:**58.54, **Rouge-2:**40.00, **Rouge-L:**53.66)
carlos colina , 32 , will be arraigned for murder charge on the morning of april 14. colina was arraigned for alleged assault and battery and improper disposal of a body. a middlesex county judge then revoked bail for colina .

long summary, our method achieves more improvement in performance (e.g., 40.85 versus 39.87 for Rouge-1 in Table I). The potential reason may be that the DEM in our model is more suitable for long sequence generation tasks while the summary in the DUC 2004 dataset is relatively short.

V. DISCUSSION

We further perform experiments to study the effect of different aspects of our DEATS method on the performance. We use the CNN/DailyMail dataset to conduct the experiments.

A. Influence of Decoding Length

In our dual encoding model, we conduct a multistep secondary decoding process for one iteration. To evaluate the influence of different decoding lengths on the performance, we set the decoding length $K = \{10, 15, 20, 25, 30, 40, 50, 100\}$. When the decoding length is set to 100, it means that we decode the whole output sequence at one time.

As shown in Fig. 2, we can see that higher precision and lower recall are obtained when the decoding length is set to a smaller value, while setting a larger value results in the

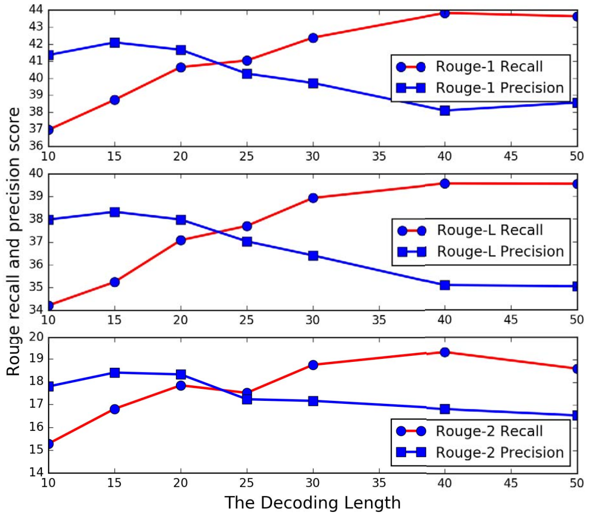

Fig. 2. Rouge recall and precision scores on the CNN/DailyMail testing set.

opposite result. We can obtain a good tradeoff between recall and precision when the decoding length is set to 20–25. In this paper, we fix the decoding length as 20 when decoding this dataset. Selecting a variable decoding length automatically may improve our model performance further, but it is a discrete action and needs to incorporate the reinforcement learning method in the model training.

We also report the ROUGE F1 score for our DEATS under different decoding lengths in Table III on this dataset. From Table III, we can see that the performance decreases notably for too small decoding length, and we obtain better results when the decoding length is set to 20–40. That is, because smaller decoding length gives rise to the increase of precision and the decrease of recall in ROUGE metric while bigger decoding length results in the opposite result. We need to balance recall and precision by setting a proper decoding length. In Table IV, we show the examples generated by our DEATS approach based on the different decoding lengths. To summarize, too large decoding length makes the secondary encoder out of function, while too small decoding length is not able to capture enough information and increases computational cost due to more secondary encoding operation.

Notably, the first two examples in Table IV show the generated summaries are directly extracted from the source documents. This is because the words in source text are more likely generated with higher probability. However, some words not existing in source text are still able to be generated, such as the third example in Table IV. The same phenomenon is also found in two other classic abstractive text summarization methods [14], [17].

B. *Effectiveness of Dual Encoding Mechanism*

We investigate the influence of different modules in our method in Table V. Specifically, we remove one or two of the three modules (i.e., PM, DEM, and RAM) from DEATS each time. Notably, our dual encoding model without the DEM and all three modules degenerates into pg+cg model and "seq2seq+atten" model in Table I, respectively. It degenerates into pointer-generator model when without the DEM and the RAM.

TABLE V
PERFORMANCE COMPARISON OF DEATS VARIANTS ON THE CNN/DAILYMAIL TESTING SET USING ROUGE F1 SCORE

DEATS variants	Rouge-1	Rouge-2	Rouge-L
DEATS	**40.85**	**18.08**	**37.13**
DEATS_w/o_RAM	38.05	16.16	34.32
DEATS_w/o_DEM	39.53	17.28	36.38
DEATS_w/o_PM	39.61	17.12	36.45
DEATS_w/o_PM+RAM	32.63	12.97	29.23
DEATS_w/o_DEM+RAM	36.44	15.66	33.42
DEATS_w/o_ALL	31.34	11.79	28.10

As presented in Table V, DEATS considering all the components shows a significant improvement of the performance over its variants. If the RAM is excluded, the Rouge scores decrease the most, indicating that the repetition phenomenon affects the performance of the generated summaries, and RAM in our model is able to restrain the repetition phenomenon well. Without the DEM, Rouge-1, Rouge-2, and Rouge-L are reduced by 1.32, 0.80, and 0.75, respectively, which is still a big decrease in summarization tasks. This indicates the secondary encoder in our model conducts a more fine encoding to help the model consider richer and more accurate information. If there is no the PM, the performance is also degraded. That is, because some OOV words are more easier to appear in the generated summaries. Using all the three components is critical to the performance of DEATS.

C. *Effectiveness of Repetition Avoidance Mechanism*

Our dual encoding model uses an enhanced RAM which combines the existing coverage mechanism with the already produced output by the decoder. To verify the capability of eliminating the repetition phenomenon in the generated summaries without coverage mechanism, we set the decoding length to a smaller value to make the decoder better remember the decoded information in the earlier time-steps. We show some examples generated by our DEATS approach without using coverage mechanism in Table VI. The results indicate that our dual encoding model without coverage mechanism is still capable of dampening the repetition when the decoding length is set to a small value.

D. *Importance Weight Visualization*

The secondary encoder in our model reads each word in input sequence in the form of skip-connections as shown in Fig. 1. This makes the secondary encoder different from the ordinary encoder. Our secondary encoder will read the input sequence more than once in one training iteration. That is to say, it conducts a secondary encoding every K decoding steps based on the input text and the previous output. Moreover, every secondary encoding will pay attention to the different part of the input sequence. We model the importance of each word as α_t. Therefore, we can visualize the importance weight of each word to observe whether the secondary encoder

TABLE VI
EXAMPLES FROM THE CNN/DAILYMAIL TESTING DATASET GENERATED BY OUR DEATS APPROACH

Source Document 1:
the operator of the crippled fukushima daiichi nuclear plant has given up trying to recover a robotic probe after it stopped moving inside one of the reactors. tokyo electric power company -lrb- tepco -rrb- deployed the remote-controlled robot on friday inside one of the damaged reactors that had suffered a meltdown following a devastating earthquake and tsunami in 2011. it was the first time the probe had been used. the robot, set out to collect data on radiation levels and investigate the spread of debris, stalled after moving about 10 meters, according to a statement released by tepco. a newly released report and footage from the robot shows that a fallen object had blocked its path and left it stranded. tepco decided to cut off the cable connected to the device sunday as it had already covered two-thirds of the originally planned route. it managed to collect data on radiation levels in 14 of the 18 targeted locations. four years after the devastating nuclear crisis, the radiation levels inside the three damaged reactors are still extremely high and remain unsafe for people to enter. decommissioning work is estimated to cost $50 billion and will take years to complete. tepco called the robotic probe an "unprecedented" experiment. cnn's yoko wakatsuki reported from tokyo, japan and naomi ng wrote from hong kong.
Ground Truth Summary:
the operator of the fukushima nuclear plant said it has abandoned a robotic probe inside one of the damaged reactors. a report stated that a fallen object has left the robot stranded. the robot collected data on radiation levels and investigated the spread of debris.
DEATS, $K = 30$, without using coverage mechanism:
tokyo electric power company deployed the remote-controlled robot on friday inside one of the damaged reactors that had suffered a meltdown following a devastating earthquake and tsunami in 2011. it was the first time the probe had been used. it was the first time the probe had been used.
DEATS, $K = 15$, without using coverage mechanism:
tokyo electric power company deployed the remote-controlled robot on friday inside one of the damaged reactors that had suffered a meltdown following a devastating earthquake and tsunami in 2011. tepco decided to cut off the cable connected to the device sunday as it had already covered two-thirds of the originally planned route.
Source Document 2:
arsenal kept their slim hopes of winning this season's english premier league title alive by beating relegation threatened burnley 1-0 at turf moor. a first half goal from welsh international aaron ramsey was enough to separate the two sides and secure arsenal's hold on second place. more importantly it took the north london club to within four points of first placed chelsea, with the two clubs to play next week. but chelsea have two games in hand and play lowly queens park rangers on sunday, a team who are themselves struggling against relegation ...... another midlands club looking over their shoulder is west brom, who conceded an injury time goal to lose 3-2 against bottom club leicester city. but it was an awful day for sunderland's former dutch international coach dick advocaat, who saw his team lose 4-1 at home against form team crystal palace. democratic republic of congo international yannick bolasie scored crystal palace's first ever hat trick in the premier league to secure an easy victory.
Ground Truth Summary:
arsenal beat burnley 1-0 in the epl. a goal from aaron ramsey secured all three points. win cuts chelsea's epl lead to four points.
DEATS, $K = 30$, without using coverage mechanism:
arsenal kept their slim hopes of winning this season's english premier league title alive by beating relegation threatened burnley 1-0 at turf moor. chelsea have two games in hand and play lowly queens park rangers on sunday. chelsea have two games in hand and play lowly queens park rangers.
DEATS, $K = 15$, without using coverage mechanism:
arsenal kept their slim hopes of winning this season's english premier league title alive by beating relegation threatened burnley 1-0 at turf moor. chelsea have two games in hand and play lowly queens park rangers on sunday.

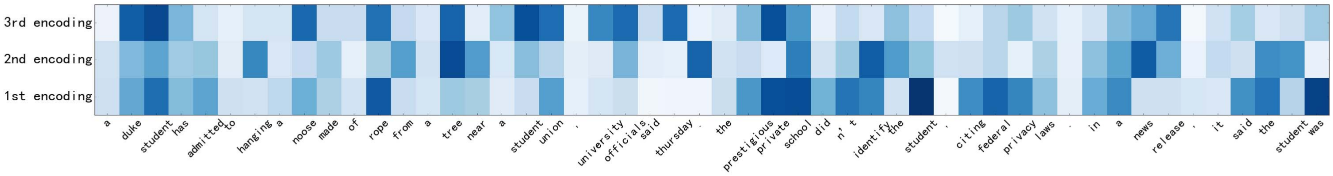

Fig. 3. Importance weight visualization on the partial input text.

assigns different weights to different words in every secondary encoding. As shown in Fig. 3, the secondary encoding is shown for three times and it pays attention to the different part every time. Specifically, the encoder assigns different weights to each word in the input sequence at one same encoding. Meanwhile, the encoder also assigns different weights to the same words at every different encoding. Notably, for clarity, we only show the importance weight visualization for each secondary encoding on the partial input sequence.

VI. CONCLUSION

In this paper, we present a dual encoding model which extends the sequence-to-sequence framework for abstractive text summarization. Our model is built on a basic encoder–decoder model with attention mechanism, the PM and the RAM. Different from the standard encoder–decoder model, the dual encoding model decodes the whole output sequence by stages and produces the partial fixed-length sequence at each stage. A combination of the DEM and the basic approaches could make them benefit from each other. The extensive experiments on the CNN/DailyMail and DUC 2004 datsets show that our dual encoding model achieves the state-of-the-art results compared to existing methods.

In our future work, we plan to focus on how to balance precision and recall to further boost F1 performance by selecting dynamic decoding length automatically based on reinforcement learning. Meanwhile, in order to achieve better performance, we also attempt to train our dual encoding model by using a hybrid training objective with reinforcement learning training and maximum-likelihood training.

Acknowledgment

The authors would like to thank Z. Peng and F. Zhang for helpful discussions about model training and network parameters tuning tricks. They would also like to thank the anonymous reviewers for their help in improving this paper.

References

[1] R. Collobert *et al.*, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Jan. 2011.

[2] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, Oct. 2014, pp. 1724–1734.

[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 3104–3112.

[4] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 4945–4949.

[5] G. Biagetti, P. Crippa, L. Falaschetti, S. Orcioni, and C. Turchetti, "An investigation on the accuracy of truncated DKLT representation for speaker identification with short sequences of speech frames," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4235–4249, Dec. 2017.

[6] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3156–3164.

[7] O. Vinyals and Q. V. Le, "A neural conversational model," *CoRR*, vol. abs/1506.05869, 2015. [Online]. Available: http://arxiv.org/abs/1506.05869

[8] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, Lisbon, Portugal, Sep. 2015, pp. 379–389.

[9] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proc. Conf. North Amer. Assoc. Comput. Linguist. Human Lang. Technol.*, San Diego CA, USA, Jun. 2016, pp. 93–98.

[10] R. Nallapati, B. Zhou, C. N. dos Santos, Ç Gülçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in *Proc. 20th SIGNLL Conf. Comput. Nat. Lang. Learn. (CoNLL)*, Berlin, Germany, Aug. 2016, pp. 280–290.

[11] K. Filippova, E. Alfonseca, C. A. Colmenares, L. Kaiser, and O. Vinyals, "Sentence compression by deletion with LSTMs," in *Proc. Conf. Empir. Methods Nat. Lang. Process.*, 2015, pp. 360–368.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[13] Ç. Gülçehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, "Pointing the unknown words," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, vol. 1. Berlin, Germany, Aug. 2016, pp. 140–149.

[14] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, vol. 1. Vancouver, BC, Canada, Jul./Aug. 2017, pp. 1073–1083.

[15] J. Gu, Z. Lu, H. Li, and V. O. K. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, vol. 1. Berlin, Germany, Aug. 2016, pp. 1631–1640.

[16] W. Zeng, W. Luo, S. Fidler, and R. Urtasun, "Efficient summarization with read-again and copy mechanism," *CoRR*, vol. abs/1611.03382, 2016. [Online]. Available: http://arxiv.org/abs/1611.03382

[17] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," *CoRR*, vol. abs/1705.04304, 2017. [Online]. Available: http://arxiv.org/abs/1705.04304

[18] J. L. Neto, A. A. Freitas, and C. A. A. Kaestner, "Automatic text summarization using a machine learning approach," in *Proc. Adv. Artif. Intell. 16th Braz. Symp. Artif. Intell. (SBIA)*, Nov. 2002, pp. 205–215.

[19] G. Erkan and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization," *CoRR*, vol. abs/1109.2128, 2011. [Online]. Available: http://arxiv.org/abs/1109.2128

[20] K. Wong, M. Wu, and W. Li, "Extractive summarization using supervised and semi-supervised learning," in *Proc. Conf. 22nd Int. Conf. Comput. Linguist. (COLING)*, Manchester, U.K., Aug. 2008, pp. 985–992.

[21] K. Filippova and Y. Altun, "Overcoming the lack of parallel data in sentence compression," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, Oct. 2013, pp. 1481–1491.

[22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 27th Annu. Conf. Neural Inf. Process. Syst. Adv. Neural Inf. Process. Syst.*, Dec. 2013, pp. 3111–3119.

[23] C. A. Colmenares, M. Litvak, A. Mantrach, and F. Silvestri, "HEADS: Headline generation as sequence prediction using an abstract feature-rich space," in *Proc. Conf. North Amer. Assoc. Comput. Linguist. Human Lang. Technol. (NAACL HLT)*, Denver, CO, USA, May/Jun. 2015, pp. 133–142.

[24] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, vol. 1. Berlin, Germany, Aug. 2016, pp. 484–494.

[25] R. Nallapati, F. Zhai, and B. Zhou, "SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, Feb. 2017, pp. 3075–3081.

[26] Z. Yong, J. E. Meng, Z. Rui, and M. Pratama, "Multiview convolutional neural networks for multidocument extractive summarization," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3230–3242, Oct. 2016.

[27] P. D. Over, H. T. Dang, and D. K. Harman, "DUC in context," *Inf. Process. Manag.*, vol. 43, no. 6, pp. 1506–1520, 2007.

[28] D. Zajic, B. Dorr, and R. Schwartz, "Bbn/umd at DUC-2004: Topiary," in *Proc. Doc. Understanding Conf. NLT/NAACL*, 2004, pp. 112–119.

[29] B. Hu, Q. Chen, and F. Zhu, "LCSTS: A large scale Chinese short text summarization dataset," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, Lisbon, Portugal, Sep. 2015, pp. 1967–1972.

[30] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2015, pp. 2692–2700.

[31] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, vol. 1. Berlin, Germany, Aug. 2016, pp. 76–85.

[32] H. Mi, B. Sankaran, Z. Wang, and A. Ittycheriah, "Coverage embedding models for neural machine translation," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, Austin, TX, USA, Nov. 2016, pp. 955–960.

[33] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, and H. Jiang, "Distraction-based neural networks for document summarization," *CoRR*, vol. abs/1610.08462, 2016. [Online]. Available: http://arxiv.org/abs/1610.08462

[34] K. M. Hermann *et al.*, "Teaching machines to read and comprehend," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2015, pp. 1693–1701.

[35] R. Nallapati, F. Zhai, and B. Zhou, "SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, Feb. 2017, pp. 3075–3081. [Online]. Available: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14636

[36] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, Oct. 2014, pp. 1724–1734.

[37] A. Graves, "Supervised sequence labelling with recurrent neural networks," in *Studies in Computational Intelligence*, vol. 385. New York, NY, USA: Springer, 2012, pp. 1–131, [Online]. Available: https://doi.org/10.1007/978-3-642-24797-2, doi: 10.1007/978-3-642-24797-2.

[38] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[39] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.

[40] Q. Zhou, N. Yang, F. Wei, and M. Zhou, "Selective encoding for abstractive sentence summarization," in *Proc. Meeting Assoc. Comput. Linguist.*, 2017, pp. 1095–1104.

[41] B. Sankaran, H. Mi, Y. Al-Onaizan, and A. Ittycheriah, "Temporal attention model for neural machine translation," *CoRR*, vol. abs/1608.02927, 2016. [Online]. Available: http://arxiv.org/abs/1608.02927

[42] C. Flick, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Workshop Text Summarization Branches Out*, 2004, p. 10.

Kaichun Yao is currently pursuing the Ph.D. degree in computer software and theory with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China.

His current research interests include natural language processing, knowledge graph, deep learning, and reinforcement learning.

Libo Zhang received the bachelor's degree in microelectronics from Anhui University, Hefei, China, the master's degree in electrical engineering from the University of Electronic Science and Technology of China, Chengdu, China, and the Doctoral degree in computer software and theory from the University of Chinese Academy of Sciences, Beijing, China.

He is currently an Assistant Professor with the Institute of Software, Chinese Academy of Sciences, Beijing. His current research interests include image processing, pattern recognition, knowledge graph, and deep learning.

Dawei Du received the B.Eng. degree in automation and the M.S. degree in detection technology and automatic engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2010 and 2013, respectively. He is currently pursuing the Ph.D. degree with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China.

His current research interests include visual tracking, video segmentation, and deep learning.

Tiejian Luo received the B.Eng. degree in computer science from Guangxi University, Guangxi, China, in 1984, the M.S. degree in computer application technology from the Institute of Computer Systems Engineering, China, in 1991, and the Ph.D. degree in computer software and theory from the Graduate University of Chinese Academy of Sciences, Beijing, China, in 2001.

He is currently a Professor with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China, where he is also the Director of the Information Dynamic and Engineering Applications Laboratory.

His current research interests include Web mining, large scale Web performance optimization, and distributed storage systems.

Lili Tao received the Ph.D. degree in computer vision from the University of Central Lancashire, Preston, U.K., in 2014.

She is currently a Senior Lecturer with the Department of Engineering, Design and Mathematics, University of the West of England, Bristol, U.K. She is also an Honorary Researcher with the University of Bristol, Bristol. Her current research interests include computer vision and robotics with a particular interest in developing methods for estimating and analyzing deformable and articulated objects, such as human motion modeling and analysis, physical activity monitoring, and facial articulated assessment.

Yanjun Wu received the Ph.D. degree in computer science from the Institute of Software, Chinese Academy of Sciences (ISCAS), Beijing, China.

He is currently a Research Professor with ISCAS. His current research interests include operating systems and system security.