Abstractive Document Summarization with a Graph-Based Attentional Neural Model

Jiwei Tan, Xiaojun Wan and **Jianguo Xiao**
Institute of Computer Science and Technology, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
{tanjiwei,wanxiaojun,xiaojianguo}@pku.edu.cn

Abstract

Abstractive summarization is the ultimate goal of document summarization research, but previously it is less investigated due to the immaturity of text generation techniques. Recently impressive progress has been made to abstractive sentence summarization using neural models. Unfortunately, attempts on abstractive document summarization are still in a primitive stage, and the evaluation results are worse than extractive methods on benchmark datasets. In this paper, we review the difficulties of neural abstractive document summarization, and propose a novel graph-based attention mechanism in the sequence-to-sequence framework. The intuition is to address the saliency factor of summarization, which has been overlooked by prior works. Experimental results demonstrate our model is able to achieve considerable improvement over previous neural abstractive models. The data-driven neural abstractive method is also competitive with state-of-the-art extractive methods.

1 Introduction

Document summarization is a task to generate a fluent, condensed summary for a document, and keep important information. As a useful technique to alleviate the information overload people are facing today, document summarization has been extensively investigated. Efforts on document summarization can be categorized to extractive and abstractive methods. Extractive methods produce the summary of a document by extracting sentences from the original document. They have the advantage of producing fluent sentences and preserving the meaning of original documents, but also inevitably face the drawbacks of information redundancy and incoherence between sentences. Moreover, extraction is far from the way humans write summaries.

On the contrary, abstractive methods are able to generate better summaries with the use of arbitrary words and expressions, but generating abstractive summaries is much more difficult in practice. Abstractive summarization involves sophisticated techniques including meaning representation, content organization, and surface realization. Each of these techniques has large space to be improved (Yao et al., 2017). Due to the immaturity of natural language generation techniques, fully abstractive approaches are still at the beginning and cannot always ensure grammatical abstracts.

Recent neural networks enable an end-to-end framework for natural language generation. Success has been witnessed on tasks like machine translation and image captioning, together with the abstractive sentence summarization (Rush et al., 2015). Unfortunately, the extension of sentence abstractive methods to the document summarization task is not straightforward. Encoding and decoding for a long sequence of multiple sentences, currently still lack satisfactory solutions (Yao et al., 2017). Recent abstractive document summarization models are yet not able to achieve convincing performance, with a considerable gap from extractive methods.

In this paper, we review the key factors of document summarization, i.e., the saliency, fluency, coherence, and novelty requirements of the generated summary. Fluency is what neural generation models are naturally good at, but the other factors are less considered in previous neural abstractive models. A recent study (Chen et al., 2016) starts to consider the factor of novelty, using a distraction mechanism to avoid redundancy. As far as we

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 1171–1181
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17-1108

know, however, saliency has not been addressed by existing neural abstractive models, despite its importance for summary generation.

In this work, we study how neural summarization models can discover the salient information of a document. Inspired by the graph-based extractive summarization methods, we introduce a novel graph-based attention mechanism in the encoder-decoder framework. Moreover, we investigate the challenges of accepting and generating long sequences for sequence-to-sequence (seq2seq) models, and propose a new hierarchical decoding algorithm with a reference mechanism to generate the abstractive summaries. The proposed method is able to tackle the constraints of saliency, non-redundancy, information correctness, and fluency under a unified framework.

We conduct experiments on two large-scale corpora with human generated summaries. Experimental results demonstrate that our approach consistently outperforms previous neural abstractive summarization models, and is also competitive with state-of-the-art extractive methods.

We organize the paper as follows. Section 2 introduces related work. Section 3 describes our method. In Section 4 we present the experiments and have discussion. Finally in Section 5 we conclude this paper.

2 Related Work

2.1 Extractive Summarization Methods

Document summarization can be categorized to extractive methods and abstractive methods. Extractive methods extract sentences from the original document to form the summary. Notable early works include (Edmundson, 1969; Carbonell and Goldstein, 1998; McDonald, 2007). In recent years much progress has also been made under traditional extractive frameworks (Li et al., 2013; Dasgupta et al., 2013; Nishikawa et al., 2014).

Neural networks have also been widely investigated on the extractive summarization task. Earlier works explore to use deep learning techniques in the traditional framework (Kobayashi et al., 2015; Yin and Pei, 2015; Cao et al., 2015a,b). More recent works predict the extraction of sentences in a more data-driven way. Cheng and Lapata (2016) propose an encoder-decoder approach where the encoder learns the representation of sentences and documents while the decoder classifies each sentence using an attention mechanism. Nallapati et al. (2017) propose a recurrent neural network (RNN)-based sequence model for extractive summarization of documents. Neural sentence extractive models are able to leverage large-scale training data and achieve performance better than traditional extractive summarization methods.

2.2 Abstractive Summarization Methods

Abstractive summarization aims at generating the summary based on understanding the input text. It involves multiple subproblems like simplification, paraphrasing, and fusion. Previous research is mostly restricted in one or a few of the subproblems or specific domains (Woodsend and Lapata, 2012; Thadani and McKeown, 2013; Cheung and Penn, 2014; Pighin et al., 2014; Sun et al., 2015).

As for neural network models, success is achieved on sentence abstractive summarization. Rush et al. (2015) train a neural attention model on a large corpus of news documents and their headlines, and later Chopra et al. (2016) extend their work with an attentive recurrent neural network framework. Nallapati et al. (2016) introduce various effective techniques in the RNN seq2seq framework. These neural sentence abstraction models are able to achieve state-of-the-art results on the DUC competition of generating headline-level summaries for news documents.

Some recent works investigate neural abstractive models on the document summarization task. Cheng and Lapata (2016) also adopt a word extraction model, which is restricted to use the words of the source document to generate a summary, although the performance is much worse than the sentence extractive model. Nallapati et al. (2016) extend the sentence summarization model by trying a hierarchical attention architecture and a limited vocabulary during the decoding phase. However these models still investigate few properties of the document summarization task. Chen et al. (2016) first attempt to explore the novelty factor of summarization, and propose a distraction-based attentional model. Unfortunately these state-of-the-art neural abstractive summarization models are still not competitive to extractive methods, and there are several problems remain to be solved.

3 Our Method

3.1 Overview

In this section we introduce our method. We adopt an encoder-decoder framework, which is

widely used in machine translation (Bahdanau et al., 2014) and dialog systems (Mou et al., 2016), etc. In particular, we use a hierarchical encoder-decoder framework similar to (Li et al., 2015), as shown in Figure 1. The main distinction of this work is that we introduce a graph-based attention mechanism which is illustrated in Figure 1b, and we propose a hierarchical decoding algorithm with a reference mechanism to tackle the difficulty of abstractive summary generation. In the following parts, we will first introduce the encoder-decoder framework, and then describe the graph-based attention and the hierarchical decoding algorithm.

3.2 Encoder

The goal of the encoder is to map the input document to a vector representation. A document d is a sequence of sentences $d = \{s_i\}$, and a sentence s_i is a sequence of words $s_i = \{w_{i,k}\}$. Each word $w_{i,k}$ is represented by its distributed representation $\mathbf{e}_{i,k}$, which is mapped by a word embedding matrix E_v. We adopt a hierarchical encoder framework, where we use a word encoder enc_{word} to encode the words of a sentence s_i into the sentence representation, and use a sentence encoder enc_{sent} to encode the sentences of a document d into the document representation. The input to the word encoder is the word sequence of a sentence, appended with an "<eos>" token indicating the end of a sentence. The word encoder sequentially updates its hidden state after receiving each word, as $\mathbf{h}_{i,k} = enc_{\text{word}}(\mathbf{h}_{i,k-1}, \mathbf{e}_{i,k})$. The last hidden state (after the word encoder receives "<eos>") is denoted as $\mathbf{h}_{i,-1}$, and used as the embedding representation of the sentence s_i, denoted as $\mathbf{x}_i$. A sentence encoder is used to sequentially receive the embeddings of the sentences, given by $\mathbf{h}_i = enc_{\text{sent}}(\mathbf{h}_{i-1}, \mathbf{x}_i)$. A pseudo sentence of an "<eod>" token is appended at the end of the document to indicate the end of the whole document. The hidden state after the sentence encoder receives "<eod>" is treated as the representation of the input document $\mathbf{c} = \mathbf{h}_{-1}$.

We use the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) as both the word encoder enc_{word} and sentence encoder enc_{sent}. In particular, we adopt the variant of LSTM structure in (Graves, 2013).

3.3 Decoder with Attention

The decoder is used to generate output sentences $\{s'_j\}$ according to the representation of the input sentences. We also use an LSTM-based hierarchical decoder framework to generate the summary, because the summary typically comprises several sentences. The sentence decoder dec_{sent} receives the document representation $\mathbf{c}$ as the initial state $\mathbf{h}'_0 = \mathbf{c}$, and predicts the sentence representations sequentially, by $\mathbf{h}'_j = dec_{\text{sent}}(\mathbf{h}'_{j-1}, \mathbf{x}'_{j-1})$, where $\mathbf{x}'_{j-1}$ is the encoded representation of the previously generated sentence s'_{j-1}. The word decoder dec_{word} receives a sentence representation $\mathbf{h}'_j$ as the initial state $\mathbf{h}'_{j,0} = \mathbf{h}'_j$, and predicts the word representations sequentially, by $\mathbf{h}'_{j,k} = dec_{\text{word}}(\mathbf{h}'_{j,k-1}, \mathbf{e}_{j,k-1})$, where $\mathbf{e}_{j,k-1}$ is the embedding of the previously generated word. The predicted word representations are mapped to vectors of the vocabulary size dimension, and then normalized by a softmax layer as the probability distribution of generating the words in the vocabulary. A word decoder stops when it generates the "<eos>" token and similarly the sentence decoder stops when it generates the "<eod>" token.

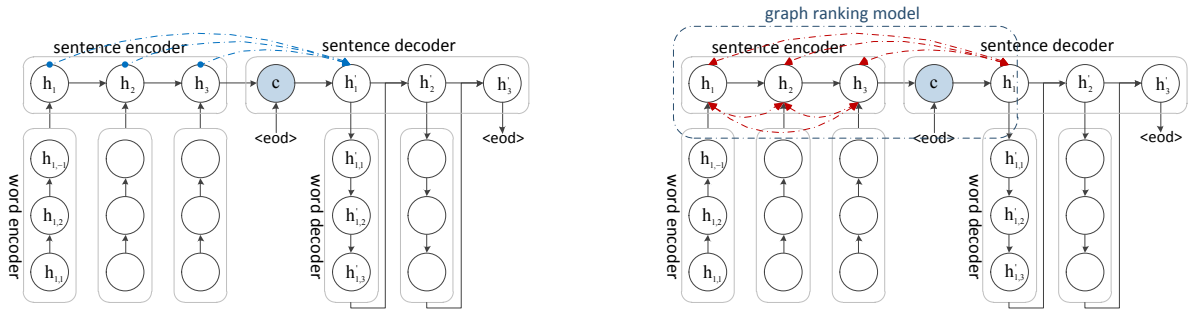
In primitive decoder models, $\mathbf{c}$ is the same for generating all the output words, which requires $\mathbf{c}$ to be a sufficient representation for the whole input sequence. The attention mechanism (Bahdanau et al., 2014) is usually introduced to alleviate the burden of remembering the whole input sequence, and to allow the decoder to pay different attention to different parts of input at different generation states. The attention mechanism sets a different $\mathbf{c}_j$ when generating sentence j, by $\mathbf{c}_j = \sum_i \alpha_i^j \mathbf{h}_i$. α_i^j indicates how much the i-th original sentence s_i contributes to generating the j-th sentence. α_i^j is usually computed as:

$$\alpha_i^j = \frac{e^{\eta\left(\mathbf{h}_i, \mathbf{h}'_j\right)}}{\sum_l e^{\eta\left(\mathbf{h}_l, \mathbf{h}'_j\right)}} \quad (1)$$

where η is the function modeling the relation between $\mathbf{h}_i$ and $\mathbf{h}'_j$. η can be defined using various functions including $\eta(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T\mathbf{b}$, $\eta(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T M \mathbf{b}$, and even a non-linear function achieved by a multi-layer neural network. In this paper we use $\eta(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T M \mathbf{b}$ where M is a parameter matrix.

3.4 Graph-based Attention Mechanism

Traditional attention computes the importance score of a sentence s_i, when generating sentence s'_j, according to the relation between the hidden state $\mathbf{h}_i$ and current decoding state $\mathbf{h}'_j$, as shown



(a) Traditional attention.

(b) Graph-based attention.

Figure 1: Hierarchical encoder-decoder framework and comparison of the attention mechanisms.

in Figure 1a. This attention mechanism is useful in scenarios like machine translation and image captioning, because the model is able to learn a relevance mapping between the input and output. However, for document summarization, it is not easy for the model to learn how to summarize the salient information of a document, i.e., which sentences are more important to a document.

To tackle this challenge, we learn from graph-based extractive summarization models TextRank (Mihalcea and Tarau, 2004) and LexRank (Erkan and Radev, 2004), which are based on the PageRank (Page et al., 1999) algorithm. These unsupervised graph-based models show good ability to identify important sentences in a document. The underlying idea is that a sentence is important in a document if it is heavily linked with many important sentences (Wan, 2010).

In graph-based extractive summarization, a graph G is constructed to rank the original sentences. The vertices V are the set of n sentences to be considered, and the edges E are the relations between the sentences, which are typically modeled by the similarity of sentences. Let $W \in \mathcal{R}^{n \times n}$ be the adjacent matrix. Then the saliency scores of the sentences are determined by making use of the global information on the graph recursively, as:

$$\mathbf{f}(t+1) = \lambda W D^{-1} \mathbf{f}(t) + (1-\lambda)\mathbf{y} \quad (2)$$

where $\mathbf{f} = [f_1, \ldots, f_n] \in \mathcal{R}^n$ denotes the rank scores of the n sentences. $\mathbf{f}(t)$ denotes the rank scores at the t-th iteration. D is a diagonal matrix with its (i,i)-element equal to the sum of the i-th column of W. Assume we use $\mathbf{h}_i$ as the representation of s_i, and $W(i,j) = \mathbf{h}_i^T M \mathbf{h}_j$, where M is a parameter matrix to be learned. λ is a damping factor. $\mathbf{y} \in \mathcal{R}^n$ with all elements equal to $1/n$. The solution of $\mathbf{f}$ can be calculated using the closed-form:

$$\mathbf{f} = (1-\lambda)(I - \lambda W D^{-1})^{-1}\mathbf{y} \quad (3)$$

In the graph model, the importance score of a sentence s_i is determined by the relation between $\mathbf{h}_i$ and the $\{\mathbf{h}_l\}$ of all other sentences. Relatively, in traditional attention mechanisms, the importance (attention) score α_i^j is determined by the relation between $\mathbf{h}_i$ and $\mathbf{h}_j'$, regardless of other original sentences. In our model we hope to combine the two effects, and compute the rank scores of the original sentences regarding $\mathbf{h}_j'$, so that the importance scores of original sentences are different when decoding different state $\mathbf{h}_j'$, denoted by $\mathbf{f}^j$. In our model we use the scores $\mathbf{f}^j$ to compute the attention. Therefore, $\mathbf{h}_j'$ should be considered in the graph model. Inspired by the query-focused graph-based extractive summarization model (Wan et al., 2007), we realize this by applying the idea of topic-sensitive PageRank (Haveliwala, 2002), which is to rank the sentences with the concern of their relevance to the topic. We treat the current decoding state $\mathbf{h}_j'$ as the topic and add it into the graph as the 0-th pseudo-sentence. Given a topic T, the topic-sensitive PageRank is similar to Eq. 3 except that $\mathbf{y}$ becomes:

$$\mathbf{y}_T = \begin{cases} \frac{1}{|T|} & i \in T \\ 0 & i \notin T \end{cases} \quad (4)$$

Therefore $\mathbf{y}_T$ is always a one hot vector and only $y_0 = 1$, indicating the 0-th sentence is s_j'. Denote W^j as the new adjacent matrix added with $\mathbf{h}_j'$, and D^j as the new diagonal matrix corresponding to W^j. Then the convergence score vector $\mathbf{f}^j$ contains the importance scores for all the

input sentences when generating sentence $s_j^{'}$, as:

$$\mathbf{f}^j = (1-\lambda)(I - \lambda W^j D^{j^{-1}})^{-1}\mathbf{y}_T \quad (5)$$

The new scores $\mathbf{f}^j$ can be used to compute the graph-based attention when decoding $\mathbf{h}_j^{'}$, to find the sentences which are both globally important and relevant to current decoding state $\mathbf{h}_j^{'}$. Inspired by (Chen et al., 2016) we adopt a distraction mechanism to compute the final attention value α_i^j, which subtracts the rank scores of the previous step, to penalize the model from attending to previously attended sentences, and also help to normalize the ranked scores $\mathbf{f}^j$. The graph-based attention is finally computed as:

$$\alpha_i^j = \frac{\max(f_i^j - f_i^{j-1}, 0)}{\sum_l \left(\max(f_l^j - f_l^{j-1}, 0)\right)} \quad (6)$$

where $\mathbf{f}^0$ is initialized with all elements equal to $1/n$. The graph-based attention will only focus on those sentences ranked higher over the previous decoding step, so that it concentrates more on the sentences which are both salient and novel. Both Eq. 5 and Eq. 6 are differentiable; thus we can use the graph-based attention function Eq. 6 to replace the traditional attention function Eq. 1, and the neural model using the graph-based attention can also be trained using traditional gradient-based methods.

3.5 Model Training

The loss function $\mathcal{L}$ of the model is the negative log likelihood of generating summaries over the training set $\mathcal{D}$:

$$\mathcal{L} = \sum_{(Y,X)\in\mathcal{D}} -\log p(Y|X;\theta) \quad (7)$$

where $X = \{x_1, \ldots, x_{|X|}\}$ and $Y = \{y_1, \ldots, y_{|Y|}\}$ denote the word sequences of a document and its summary respectively, including the "<eos>" and "<eod>" tokens for structure information. Then

$$\log p(Y|X;\theta) = \sum_{\tau=1}^{|Y|} \log p\left(y_\tau | \{y_1, \ldots, y_{\tau-1}\}, \mathbf{c}; \theta\right) \quad (8)$$

and $\log p\left(y_\tau | \{y_1, \ldots, y_{\tau-1}\}, \mathbf{c}; \theta\right)$ is modeled by the LSTM encoder and decoder. We use the Adamax (Kingma and Ba, 2014) gradient-based optimization method to optimize the model parameters θ.

3.6 Decoding Algorithm

We find there are several problems during the generation of summary, including out-of-vocabulary (OOV) words, information incorrectness, error accumulation and repetition. These problems make the generated abstractive summaries far from satisfactory. In this work, we propose a hierarchical decoding algorithm with a reference mechanism to tackle these difficulties, which effectively improves the quality of generated summaries.

As OOV words frequently occur in name entities, we can first identify the entities of a document using NLP toolkit like Stanford CoreNLP[1]. Then we prefix every entity with an "@entity" token and a number indicating how many words the entity has. We hope the entity prefixes can help better deal with entities which have more than one word, and help improve the accuracy of recovering OOV words in entities. After decoding we recover the OOV words by matching entities in the original document according to the contexts.

For the hierarchical decoder, a major challenge is that same sentences or phrases are often repeated in the output. A beam search strategy may help to alleviate the repetition in a sentence, but the repetition in the whole generated summary is remained a problem. The word-level beam search is not easy to be extended to the sentence level. The reason is that the K-best sentences generated by a word decoder will mostly be similar to each other, which is also noticed by Li et al. (2016).

In this paper we propose a hierarchical beam search algorithm with a reference mechanism. The hierarchical algorithm comprises K-best word-level beam search and N-best sentence-level beam search. At the word level, the only difference to vanilla beam search is that we add an additional term to the score $\tilde{p}(y_\tau)$ of generating word y_τ, and now $score(y_\tau) = \tilde{p}(y_\tau) + \gamma\left(ref(Y_{\tau-1} + y_\tau, s_*) - ref(Y_{\tau-1}, s_*)\right)$, where $Y_{\tau-1} = \{y_1, \ldots, y_{\tau-1}\}$ and $\tilde{p}(y_\tau) = \log p\left(y_\tau | Y_{\tau-1}, \mathbf{c}; \theta\right)$. s_* is an original sentence to refer to. ref is a function which calculates the ratio of bigram overlap between two texts. The added term aims to favor the generated word y_τ with improving the bigram overlap between current generated summary $Y_{\tau-1}$ and the target orig-

[1] http://stanfordnlp.github.io/CoreNLP/

Dataset	Train	Valid	Test	D.L.	S. L.
CNN	83568	1220	1093	29.8	3.54
DailyMail	196557	12147	10396	26.0	3.84

Table 1: The statistics of the two datasets. D.L. and S.L. indicate the average number of sentences in the document and summary, respectively.

inal sentence s_*. At the word decoder level, the reference mechanism helps to both improve the information correctness and avoid redundancy. Because the reference score is based on the bigram overlap improvement to the whole generated summary $Y_{\tau-1}$, the awareness of previously generated sentences also helps alleviate sentence-level redundancy. A factor γ is introduced to control the influence of the reference mechanism. Note that because of the non-optimal search, the generated sentence will still be different to the original sentence even with an extremely large γ.

At the sentence level, N-best sentence beam is to keep the N generated sentences by referring to N different original sentences, which have the highest attention scores and have not been used as a reference. With referring to N different sentences, the N candidate sentences are guaranteed diverse. Sentence-level beam search is realized by maximizing the accumulated score of all the sentences generated.

4 Experiments

4.1 Dataset

We conduct experiments on two large-scale corpora of CNN and DailyMail, which have been widely used in neural document summarization tasks. The corpora are originally constructed in (Hermann et al., 2015) by collecting human generated abstractive highlights from the news stories in the CNN and DailyMail website. The statistics and split of the two datasets are listed in Table 1.

4.2 Implementation

We use the corpora which are already provided with labeled entities (Nallapati et al., 2016). The documents and summaries are first lowercased and tokenized, and all digit characters are replaced with the "#" symbol, similar to (Nallapati et al., 2016, 2017). We keep the 40,000 most frequently occurring words and other words are replaced with the "<OOV>" token.

We use Theano[2] for implementation. For the word encoder and decoder we use three layers of LSTM, and for the sentence encoder and decoder we use one layer of LSTM. The dimension of hidden vectors are all 512. We use pre-trained GloVe (Pennington et al., 2014) vectors[3] for the initialization of word vectors, which will be further trained in the model. The dimension of word vectors is 100. λ is set to 0.9. The parameters of Adamax are set to those provided in (Kingma and Ba, 2014). The batch size is set to 8 documents, and an epoch is set containing 10,000 randomly sampled documents. Convergence is reached within 200 epochs on the DailyMail dataset and 120 epochs on the CNN dataset. It takes about one day for every 30 epochs on a GTX-1080 GPU card. γ is tuned on the validation set and the best choice is 300. The beam sizes for word decoder and sentence decoder are 15 and 2, respectively.

4.3 Evaluation

We adopt the widely used ROUGE (Lin, 2004) toolkit for evaluation. We first compare with the reported results in (Chen et al., 2016) including various traditional extractive methods and a state-of-the-art abstractive model (Distraction-M3) on the CNN dataset, as shown in Table 2. Uni-GRU is a non-hierarchical seq2seq baseline model. In Table 3 we compare our method with the results of state-of-the-art neural summarization methods reported in recent papers. Extractive models include NN-SE (Cheng and Lapata, 2016) and SummaRuNNer (Nallapati et al., 2017), while SummaRuNNer-abs is also an extractive model similar to SummaRuNNer but is trained directly on the abstractive summaries. Moreover, we include several baselines for comparison, including the baselines reported in (Cheng and Lapata, 2016) although they are tested on 500 samples of the test set. LREG is a feature based method using linear regression. NN-ABS is a neural abstractive baseline which is a simple hierarchical extension of (Rush et al., 2015). NN-WE is the abstractive model which restricts the generation of words from the original document. Lead-3 is a strong extractive baseline that uses the lead three sentences as the summary.

In Table 4 we compare our model with the abstractive attentional encoder-decoder models in

[2]https://github.com/Theano/Theano
[3]http://nlp.stanford.edu/projects/glove

Method	Rouge-1	Rouge-2	Rouge-L
Lead-3	26.1	9.6	17.8
Luhn	23.2	7.2	15.5
Edmundson	24.5	8.2	16.7
LSA	21.2	6.2	14.0
LexRank	26.1	9.6	17.7
TextRank	23.3	7.7	15.8
Sum-basic	22.9	5.5	14.8
KL-sum	20.7	5.9	13.7
Uni-GRU	18.4	4.8	14.3
Distraction-M3	27.1	8.2	18.7
Our Method	**30.3**	**9.8**	**20.0**

Table 2: Comparison results on the **CNN test set** using the **full-length F1** variants of Rouge.

Method	Rouge-1	Rouge-2	Rouge-L
LREG(500)	18.5	6.9	10.2
NN-ABS(500)	7.8	1.7	7.1
NN-WE(500)	15.7	6.4	9.8
Lead-3	21.9	7.2	11.6
NN-SE	22.7	8.5	12.5
SummaRuNNer-abs	23.8	9.6	13.3
SummaRuNNer	26.2	10.8	14.4
Our Method	**27.4**	**11.3**	**15.1**

Table 3: Comparison results on the **DailyMail test set** using Rouge **recall** at **75 bytes**.

(Nallapati et al., 2016), which leverage several effective techniques and achieve state-of-the-art performance on sentence abstractive summarization tasks. The words-lvt2k and words-lvt2k-ptr are flat models and words-lvt2k-hieratt is a hierarchical extension.

Results in Table 2 show our abstractive method is able to outperform traditional extractive methods and the distraction-based abstractive model. The results in Tables 3 and 4 show that our method has considerable improvement over neural abstractive baselines, and is able to outperform state-of-the-art neural extractive methods. An interesting observation is the results of the hierarchical model in Table 4 are lower than the flat models, which may demonstrate the difficulty for a traditional attention model to identify the important information in a document.

We also conducted human evaluation on 20 random samples from the DailyMail test set and compared the summaries generated by our method with the outputs of Lead-3, NN-SE (Cheng and Lapata, 2016) and Distraction (Chen et al., 2016). The output summaries of NN-SE are provided by the authors, and the output summaries of Distraction are achieved by running the code provided by the authors on the DailyMail dataset. Three participants were asked to compare the generated summaries with the human summaries, and assess each summary from four independent perspectives: (1) How informative the summary is? (2) How concise the summary is? (3) How coherent (between sentences) the summary is? (4) How fluent, grammatical the sentences of a summary are? Each property is assessed with a score from 1 (worst) to 5 (best). The average results are presented in Table 5.

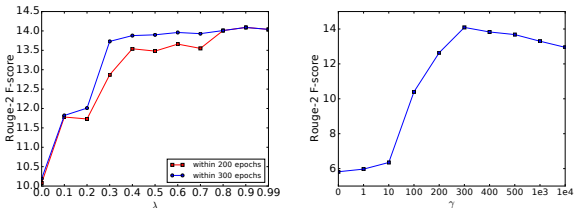
Method	Rouge-1	Rouge-2	Rouge-L
words-lvt2k	32.5	11.8	29.5
words-lvt2k-ptr	32.1	11.7	29.2
words-lvt2k-hieratt	31.8	11.6	28.7
Our Method	**38.1**	**13.9**	**34.0**

Table 4: Comparison results on the **merged CNN/DailyMail test set** using **full-length F1** metric.

Method	Informative	Concise	Coherent	Fluent
Lead-3	3.60	3.75	4.16	3.85
NN-SE	3.85	3.70	3.48	3.78
Distraction	3.03	3.25	2.93	3.65
Our Method	3.93	3.82	3.53	3.80

Table 5: Human evaluation results.

As shown in Table 5, our method consistently outperforms the previous state-of-the-art abstractive method Distraction. Compared with extractive methods, our method is able to generate more informative and concise summaries, which shows the advantage of abstractive methods. The Distraction method in fact usually produces the shortest summaries, but the conciseness score is low mainly because sometimes it generates repeated sentences. The repetition also causes Distraction to achieve a low coherence score. Concerning coherence and fluency, our abstractive method achieves slightly better scores than NN-SE, while not surprisingly Lead-3 gets the best scores. The fluency scores show the good ability of the abstractive model to generate fluent and grammatical sentences.



(a) Rouge-2 F1 score vs. λ. (b) Rouge-2 F1 score vs. γ.

Figure 2: Results of different setting of hyper-parameters tested on 500 samples from the DailyMail test set.

Framework	Rouge-1	Rouge-2	Rouge-L
Our Method	**30.3**	**9.8**	**20.0**
w/o GraphAtt	29.2	9.0	19.0
w/o SentenceBeam	29.6	9.3	19.1
w/o BeamSearch	25.1	6.7	17.9

Table 6: Results of removing different components of our method on the **CNN test set** using the **full-length F1** variants of Rouge. Two-tailed t-tests demonstrate the difference between **Our Method** and other frameworks are all statistically significant ($p < 0.01$).

4.4 Model Validation

We conduct experiments to see how the model's performance is affected by the choice of the hyper-parameters. For efficiency we test on 500 random samples from the DailyMail test set. Figure 2a shows the maximum average Rouge-2 F1-score achieved when the model is trained using different λ values within 200 and 300 epochs. When using a larger λ, the performance is better and the convergence is faster. When $\lambda = 1.0$ the model fails to train because of running into a singular matrix.

Figure 2b shows the results achieved when using different γ values in the hierarchical decoding algorithm. $\gamma = 0$ is the baseline of the traditional decoding algorithm which does not refer to the original document. The poor results indicate that even the model is able to learn to identify the salient information in the original document, the performance is limited by the model's ability of generating a long output sequence. That may be a reason why simple extensions of seq2seq models fail on the abstractive document summarization task. The performance is significantly improved using a reasonable γ, and the optimal γ value is consistent with the one chosen on the validation set. When using an extremely large γ, the permanence begins to decrease, because the model will copy too much from the original document, and at this time the generated text also becomes less fluent. Results show that introducing the reference mechanism in the hierarchical beam search is very effective. The γ factor significantly affects the results, but the optimal value is easy to be decided on a validation set.

We also conduct ablation experiments on the CNN dataset to verify the effectiveness of the proposed model. Results on the CNN test set are shown in Table 6. "*w/o GraphAtt*" is to replace the graph-based attention by a traditional attention function. "*w/o SentenceBeam*" is to remove the sentence-level beam search. "*w/o BeamSearch*" is to remove both the sentence-level and word-level beam search, and use a greedy decoding algorithm with the reference mechanism. As seen from Table 6, the graph-based attention mechanism is significantly better than traditional attention mechanism for the document summarization task. Beam search helps significantly improve the generated summaries. Our proposed decoding algorithm enables a sentence-level beam search, which helps improve the generated summaries with multiple sentences.

4.5 Case Study

We show the case study of a sample[4] from the DailyMail test set in Figure 3. We show the "@entity" and number here although they are removed in the evaluation. We compare our result with the output by a model using traditional attention as *Baseline Attention*. We also show the output generated by a *Baseline Decoder*, which sets $\gamma = 0$ and does not use the sentence-level beam search, to study the difficulty for a traditional decoder to generate multiple sentences. Many observations can be found in Figure 3. The lead three sentences mainly focus on the money information and are not sufficient. As for the *Baseline Decoder*, first it usually ends the generation too early. The "<eod>" token indicates where the original output stops. When we force the decoder not to end here, the model shows the ability to continue producing the important information. However, two flaws are presented. First is the repetition of "## - year -

[4]The original story and highlights can be found at http://www.dailymail.co.uk/news/article-3041766/Benefits-cheat-pocketed-17-000-taxpayers-money.html

Gold Summary:
@entity 2 mary day , ## , claimed over £ ##,### in benefits despite not being eligible .
she had £ ##,### savings in the bank which meant she was not entitled .
day used taxpayers ' money to go on luxury holidays to @entity 1 indian resort of @entity 1 goa .
pleaded guilty to dishonestly claiming benefits and has paid back money .

Lead3:
a benefits cheat who pocketed almost £ ##,### of taxpayers ' money and spent it on a string of luxury holidays despite having £ ##,### in the bank has avoided jail .
@entity 2 mary day , ## , of @entity 1 swanage in @entity 1 dorset , used taxpayers ' money to go on luxury holidays to the @entity 1 indian resort of @entity 1 goa for up to a month each time .
day fraudulently claimed £ ##,### of income support and disability allowance despite having £ ##,### of her own savings in the bank .

Baseline Decoder:
- month - old @entity 2 mary day , ## , was given £ ##,### in money .
the ## - year - old claimed £ ##,### in disability allowance . <*eod*>
the ## - year - old was given a six - month prison sentence .
- year - old pleaded guilty to two counts of fraud .

Baseline Attention:
@entity 2 mary day , ## , used taxpayers ' money to go on luxury holidays .
claimed £ ##,### of income support and disability allowance despite having savings in the bank . <*eod*>
benefits of taxpayers £ ##,### in disability handouts .

Our Method:
@entity 2 mary day , ## , used taxpayers ' money to go on luxury holidays to the @entity 1 indian resort of @entity 1 goa .
despite having £ ##,### of her own savings in the bank , she claimed £ ##,### of income support and disability allowance .
she pleaded guilty and had given the sentence for three months in prison , but suspended the sentence for ## months .

Figure 3: Examples of generated summaries.

old". Because the word decoder is unaware of the history generated sentences, it repeats generating the sequence as the subject all the time. Second, more importantly, is the information incorrectness. The "## - month - old" is not appropriate to describe the heroine, and the "six - month prison sentence" is in fact "three months". Information incorrectness occurs because, for a decoder, it aims at generating a fluent sentence according to the input representation. However, no favor of consistent with the original input is concerned. The proposed hierarchical decoding algorithm helps to alleviate the two problems. The awareness of all the generated sentences helps prevent from always generating some important information. The favor of bigram overlapping with the original sentences helps generate more correct sentences. For example the model is able to correctly distinguish between the "three-month sentence" and the "##-month suspend". In conclusion, our method is able to identify the most important information in the original document, and the decoding algorithm we propose is able to generate a more discourse-fluent and information-correct abstractive summary.

The visualization of the graph-based attention when our method generates the presented example

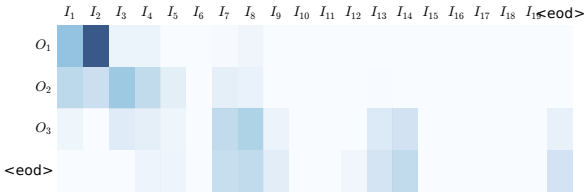


Figure 4: Attention heatmap when generating the example summary. I_i and O_i indicate the i-th sentence of the input and output, respectively.

is shown in Figure 4. It seems that the graph-based attention mechanism is able to find the important sentences in the input document, and the distraction mechanism makes the decoder focus on different sentences during decoding. Gradually the decoder attends to "<eod>" until it stops.

5 Conclusion and Future Work

In this paper we tackle the challenging task of abstractive document summarization, which is still less investigated to date. We study the difficulty of the abstractive document summarization task, and address the need of finding salient content from the original document, which is overlooked by previous studies. We propose a novel graph-based attention mechanism in a hierarchical encoder-decoder framework, and propose a hierarchical beam search algorithm to generate multi-sentence summary. Extensive experiments verify the effectiveness of the proposed method. Experimental results on two large-scale datasets demonstrate our method achieves state-of-the-art abstractive document summarization performance. It is also able to achieve competitive results with state-of-the-art neural extractive summarization models.

There is lots of future work we can do. An appealing direction is to investigate the neural abstractive method on the multi-document summarization task, which is more challenging and lacks training data. Further endeavor may be needed.

Acknowledgments

This work was supported by 863 Program of China (2015AA015403), NSFC (61331011), and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We thank the anonymous reviewers for helpful comments and Xinjie Zhou, Jianmin Zhang for doing human evaluation. Xiaojun Wan is the corresponding author.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015a. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*. pages 2153–2159.

Ziqiang Cao, Furu Wei, Sujian Li, Wenjie Li, Ming Zhou, and Houfeng Wang. 2015b. Learning summary prior representation for extractive summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, pages 829–833. https://doi.org/10.3115/v1/P15-2136.

Jaime G. Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*. pages 335–336.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for document summarization. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*. pages 2754–2760.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 484–494. https://doi.org/10.18653/v1/P16-1046.

Kit Jackie Chi Cheung and Gerald Penn. 2014. Unsupervised sentence enhancement for automatic summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pages 775–786. https://doi.org/10.3115/v1/D14-1085.

Sumit Chopra, Michael Auli, and M. Alexander Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 93–98. https://doi.org/10.18653/v1/N16-1012.

Anirban Dasgupta, Ravi Kumar, and Sujith Ravi. 2013. Summarization through submodularity and dispersion. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1014–1022. http://aclweb.org/anthology/P13-1100.

Harold P Edmundson. 1969. New methods in automatic extracting. *Journal of the ACM (JACM)* 16(2):264–285.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res. (JAIR)* 22:457–479.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* .

Taher H. Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of the Eleventh International World Wide Web Conference, WWW 2002, May 7-11, 2002, Honolulu, Hawaii*. pages 517–526.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. pages 1693–1701.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

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

Hayato Kobayashi, Masaki Noguchi, and Taichi Yatsuka. 2015. Summarization based on embedding distributions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1984–1989. https://doi.org/10.18653/v1/D15-1232.

Chen Li, Xian Qian, and Yang Liu. 2013. Using supervised bigram-based ilp for extractive summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1004–1013. http://aclweb.org/anthology/P13-1099.

Jiwei Li, Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1106–1115. https://doi.org/10.3115/v1/P15-1107.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562* .

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 workshop*. Barcelona, Spain, volume 8.

Ryan T. McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Advances in Information Retrieval, 29th European Conference on IR Research, ECIR 2007, Rome, Italy, April 2-5, 2007, Proceedings*. pages 557–564.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. pages 404–411.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, pages 3349–3358. http://aclweb.org/anthology/C16-1316.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*. pages 3075–3081.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pages 280–290. https://doi.org/10.18653/v1/K16-1028.

Hitoshi Nishikawa, Kazuho Arita, Katsumi Tanaka, Tsutomu Hirao, Toshiro Makino, and Yoshihiro Matsuo. 2014. Learning to generate coherent summary with discriminative hidden semi-markov model. In *Proceedings of COLING 2014*. Dublin City University and Association for Computational Linguistics, pages 1648–1659.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pages 1532–1543. https://doi.org/10.3115/v1/D14-1162.

Daniele Pighin, Marco Cornolti, Enrique Alfonseca, and Katja Filippova. 2014. Modelling events through memory-based, open-ie patterns for abstractive summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 892–901. https://doi.org/10.3115/v1/P14-1084.

M. Alexander Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 379–389. https://doi.org/10.18653/v1/D15-1044.

Rui Sun, Yue Zhang, Meishan Zhang, and Donghong Ji. 2015. Event-driven headline generation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 462–472. https://doi.org/10.3115/v1/P15-1045.

Kapil Thadani and Kathleen McKeown. 2013. Supervised sentence fusion with single-stage inference. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, pages 1410–1418.

Xiaojun Wan. 2010. Towards a unified approach to simultaneous single-document and multi-document summarizations. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Coling 2010 Organizing Committee, pages 1137–1145.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Manifold-ranking based topic-focused multi-document summarization. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*. pages 2903–2908.

Kristian Woodsend and Mirella Lapata. 2012. Multiple aspect summarization using integer linear programming. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, pages 233–243.

Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. 2017. Recent advances in document summarization. *Knowledge and Information Systems* .

Wenpeng Yin and Yulong Pei. 2015. Optimizing sentence modeling and selection for document summarization. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. pages 1383–1389.