Adapting Neural Single-Document Summarization Model for Abstractive Multi-Document Summarization: A Pilot Study

Jianmin Zhang and **Jiwei Tan** and **Xiaojun Wan**
Institute of Computer Science and Technology, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
{zhangjianmin2015,tanjiwei,wanxiaojun}@pku.edu.cn

Abstract

Till now, neural abstractive summarization methods have achieved great success for single document summarization (SDS). However, due to the lack of large scale multi-document summaries, such methods can be hardly applied to multi-document summarization (MDS). In this paper, we investigate neural abstractive methods for MDS by adapting a state-of-the-art neural abstractive summarization model for SDS. We propose an approach to extend the neural abstractive model trained on large scale SDS data to the MDS task. Our approach only makes use of a small number of multi-document summaries for fine tuning. Experimental results on two benchmark DUC datasets demonstrate that our approach can outperform a variety of baseline neural models.

1 Introduction

Document summarization is a task of automatically producing a summary for given documents. Different from Single Document Summarization (SDS) which generates a summary for each given document, Multi-Document Summarization (MDS) aims to generate a summary for a set of topic-related documents. Previous approaches to document summarization can be generally categorized to extractive methods and abstractive methods. Extractive methods produce a summary by extracting and merging sentences from the original document(s), while abstractive methods generate a summary using arbitrary words and expressions based on understanding the document(s). Due to the difficulty of natural language understanding and generation, previous research on document summarization is more focused on extractive methods (Yao et al., 2017). However, extractive methods suffer from the inherent drawbacks of discourse incoherence and long, redundant sentences, which hampers its application in reality (Tan et al., 2017). Recently, with the success of sequence-to-sequence (seq2seq) models in natural language generation tasks including machine translation (Bahdanau et al., 2014) and dialog systems (Mou et al., 2016), abstractive summarization methods has received increasing attention. With the resource of large-scale corpus of human summaries, it is able to train an abstractive summarization model in an end-to-end framework. Neural abstractive summarization models (See et al., 2017; Tan et al., 2017) have surpass the performance of extractive methods on single document summarization task with abundant training data.

Unfortunately, the extension of seq2seq models to MDS is not straightforward. Neural abstractive summarization models are usually trained on about hundreds of thousands of gold summaries, but there are usually very few human summaries available for the MDS task. More specifically, in the news domain, there is only a few hundred multi-document summaries provided by DUC and TAC conferences in total, which are largely insufficient for training neural abstractive models. Apart from insufficient training data, neural models for abstractive MDS also face the challenge of much more input content, and the study is still in the primary stage.

In this study, we investigate applying seq2seq models to the MDS task. We attempt various ways of extending neural abstractive summarization models pre-trained on the SDS data to the MDS task, and reveal that neural abstractive summarization models do not transfer well on a different dataset. Then we study the factors which affect the transfer performance, and propose methods to

Proceedings of The 11th International Natural Language Generation Conference, pages 381–390,
Tilburg, The Netherlands, November 5-8, 2018.

adapt the pre-trained model to the MDS task. We also study leveraging the few MDS training data to further improve the pre-trained model. We conduct experiment on the benchmark DUC datasets, and experiment results demonstrate our approach is able to achieve considerable improvement over a variety of neural baselines.

The contributions of this study are summarized as follows:

- To the best of our knowledge, our work is one of the very few pioneering works to investigate adapting neural abstractive summarization models of single document summarization to the task of multi-document summarization.
- We propose a novel approach to adapt the neural model trained on the SDS data to the MDS task, and leverage the few MDS training data to further improve the pre-trained model.
- Evaluation results demonstrate the efficacy of our proposed approach, which outperforms a variety of neural baselines.

We organize the paper as follows. In Section 2 we introduce related work. In Section 3 we describe the previous neural abstractive summarization model. Then we introduce our proposed approach in Section 4. Experiment results and discussion are presented in Section 5. Finally, we conclude this paper in Section 6.

2 Related Work

2.1 Extractive Summarization Methods

The study of MDS is pioneered by (McKeown and Radev, 1995), and early notable works also include (McKeown et al., 1999; Radev et al., 2000). Extractive summarization systems that compose a summary from a number of important sentences from the source documents are by far the most popular solution for MDS (Avinesh and Meyer, 2017). Redundancy is one of the biggest problems for extractive methods (Gambhir and Gupta, 2017), and the Maximal Marginal Relevance (MRR) (Carbonell and Goldstein, 1998) is a well-known algorithm for reducing redundancy. In the past years various models under extractive framework have been proposed (Tao et al., 2008; Wan and Yang, 2008; Wang et al., 2011; Tan et al., 2015). One important architecture is to model MDS as a budgeted maximum coverage problem, including the prior approach (McDonald, 2007) and improved models (Woodsend and Lapata, 2012; Li et al., 2013; Boudin et al., 2015). There are still recent studies under traditional extractive framework (Peyrard and Eckle-Kohler, 2017; Avinesh and Meyer, 2017).

2.2 Abstractive Summarization Methods

Abstractive summarization methods aim at generating the summary based on understanding the original documents. Sequence-to-sequence models with attention mechanism have been applied to the abstractive summarization task. Success attempts are on sentence summarization (Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016) or single document summarization (Tan et al., 2017; See et al., 2017; Paulus et al., 2017), which have abundant gold summaries to train an end-to-end system.

Until very recently, there occurs attempt for abstractive multi-document summarization under the seq2seq framework. The lack of enough train examples is the major obstacle to this end. To address this, Liu et al. (2018) study the task of generating English Wikipedia under a viewpoint of multi-document summarization. They construct a large corpus with reference summaries, so that end-to-end training of a seq2seq is capable. Their study reveals that seq2seq model works when there are abundant training data for MDS. Very recently Baumel et al. (2018) try to apply pre-trained abstractive summarization model of SDS to the query-focused summarization task. They sort the input documents and then iteratively apply the SDS model to summarize each single document until the length limit is reached. Their major concern is incorporating query information into the abstractive model or using the query to filter the original documents, which is different from our work focusing on generic multi-document summarization. Moreover, the intuitive idea of using the SDS model for summarizing each single document in the multi-document set is adopted in the baseline models for comparison as well.

3 Preliminaries

In this work we investigate abstractive MDS approach based on the state-of-the-art neural abstractive model in Tan et al. (2017). Compared with another neural abstractive model in See et al. (2017),

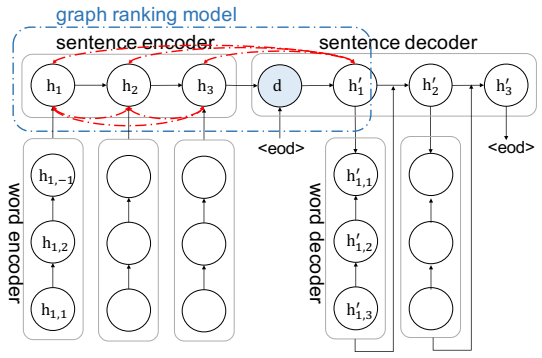

Figure 1: SinABS model. The figure is borrowed from Tan et al. (2017).

Tan et al. (2017) adopt a hierarchical encoder-decoder framework which we found is more scalable to more and longer input documents. The model is named **SinABS** in this paper. SinABS uses a hierarchical encoder-decoder framework like Li et al. (2015), where a PageRank (Page et al., 1999) based attention mechanism is proposed to identify salient sentences in the original documents. The SinABS model is illustrated in Figure 1.We introduce the SinABS model following Tan et al. (2017).

3.1 Encoder

The target of the encoder is to encode the input documents into vector representations. SinABS adopts a hierarchical encoder framework, where a word encoder enc_{word} is used for encoding a sentence into the sentence representation from its words, as $\mathbf{h}_{i,k} = enc_{\text{word}}(\mathbf{h}_{i,k-1}, \mathbf{e}_{i,k})$, where $\mathbf{h}_{i,k}$ represents the hidden state when LSTM receives word $\mathbf{e}_{i,k}$. Then a sentence encoder enc_{sent} is used for encoding an input document into the document representation from its sentences, as $\mathbf{h}_i = enc_{\text{sent}}(\mathbf{h}_{i-1}, \mathbf{x}_i)$, where $\mathbf{x}_i = \mathbf{h}_{i,-1}$ is the last hidden state when word encoder receives the whole sentence i. 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 last hidden state after the word encoder receives "<eos>" is used as the embedding representation of the sentence. A sentence encoder is used to sequentially receive the embeddings of the sentences. 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, denoted as $\mathbf{c}$. Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) is used as the word encoder enc_{word} and also the sentence encoder enc_{sent}.

3.2 Decoder

Similar to the hierarchical encoder, The sentence decoder dec_{sent} receives the document representation $\mathbf{d}$ as the initial state $\mathbf{h}'_0 = \mathbf{d}$, 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 first concatenated with the context vector $\mathbf{c}_j$, and then mapped to vectors of the vocabulary size dimension by a projection layer, and finally 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.

3.3 Attention Mechanism

The attention mechanism used in SinABS sets a different context vector $\mathbf{c}_j$ when generating the words of sentence j, by $\mathbf{c}_j = \sum_i \alpha_i^j \mathbf{h}_i$. The graph-based attention mechanism in Tan et al. (2017) adopts the topic-sensitive PageRank algorithm to compute the attention weights, by

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

where $\mathbf{f} = [f_1, \ldots, f_n] \in \mathcal{R}^n$ denotes the rank scores of the n original sentences. D is a diagonal matrix with its (i,i)-element equal to the sum of the i-th column of W. $W(i,j) = \mathbf{h}_i^T P \mathbf{h}_j$ where P is a parameter matrix to be learned. λ is a damping factor and set to 0.9. $\mathbf{y} \in \mathcal{R}^n$ is a one hot vector and only $y_0 = 1$. The ranked scores are then integrated with a distraction mechanism, and finally computed as:

$$\alpha_i^j = \frac{\max(f_i^j - f_i^{j-1}, 0)}{\sum_l \left(\max(f_l^j - f_l^{j-1}, 0)\right)} \qquad (2)$$

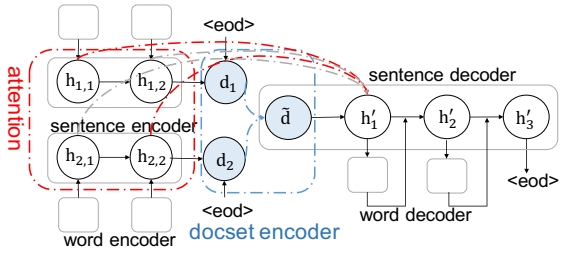

Figure 2: Framework of our model. The difference from Figure 1 is the docset encoder and the concentrated attention mechanism.

4 Our Approach

4.1 Overview

In this section we introduce our approach. Our abstractive MDS model is the extension of the single document summarization model SinABS. It is an encoder-decoder framework, which takes all the documents of a document set as input, then encodes the documents into a document set representation, and further generates the summary with a decoder. To adapt SinABS to the MDS task, our model is different from SinABS in the encoder model and the attention mechanism, and it will also be tuned on the MDS dataset to adapt to the MDS task. The framework of our model is illustrated in Figure 2.

4.2 Multi-Document Encoder

The major difference of MDS is that we need to generate a summary for multiple input documents. So our system needs to deal with the multiple input documents although SinABS is trained to generate a summary for one document. Considering that the decoder generates the summary from the representation vector encoded by the encoder, we can generate a summary for a document set if the document set is encoded to a representation vector containing its key information. In our approach, we achieve this by adding a document set encoder, to encode a set of document representation vectors into a document set representation. Thus the hierarchical encoder structure becomes three levels.

The document set encoder enc_{docset} takes document vectors $\{\mathbf{d}_m\}$, $m \in [1, M]$ where M is the number of documents in a document set as input, produces a new document set vector $\tilde{\mathbf{d}}$, and then $\tilde{\mathbf{d}}$ is provided to the decoder to generate the summary for the document set. The decoder will be a two-level hierarchical framework similar to that in Tan et al. (2017). Since there is no order and dependency relationship between different documents in a document set, it is not reasonable to use LSTM as the document set encoder. Instead, we define the document set encoder as:

$$\tilde{\mathbf{d}} = enc_{\text{docset}}(\{\mathbf{d}_m\}) = \sum_m w_m \mathbf{d}_m \quad (3)$$

where $\mathbf{w} = [w_1, \ldots, w_m] \in \mathcal{R}^m$ is a weight vector to merge the document vectors into a document set representation. The weight vector $\mathbf{w}$ can be a fixed one as $\mathbf{w} = [1/m, \ldots, 1/m]$, but in our system we hope to assign different w_m to different $\mathbf{d}_m$, since different documents may contribute differently to the overall summary. However, it is unreasonable to treat $\mathbf{w}$ as a parameter vector and learn it directly, because the weight w_m for $\mathbf{d}_m$ should be based on $\mathbf{d}_m$. The position of a document should not affect its weight since there is no order in a document set.

In our system the weight for a document is decided based on the document itself, and its contribution to the representation of the overall document set. Therefore, we define:

$$w_m = \frac{\mathbf{q}^T[\mathbf{d}_m; \mathbf{d}_\Sigma]}{\sum_{m'} \mathbf{q}^T[\mathbf{d}_{m'}; \mathbf{d}_\Sigma]} \quad (4)$$

where $\mathbf{d}_\Sigma = \sum_m \mathbf{d}_m$ and $[\mathbf{d}_m; \mathbf{d}_\Sigma]$ is the concatenation of $\mathbf{d}_m$ and $\mathbf{d}_\Sigma$. The intuitive explanation of Eq. 4 is that the weight of $\mathbf{d}_m$ is decided by its relationship (modeled by parameterized dot product) with the representation of the whole document set $\mathbf{d}_\Sigma$. $\mathbf{q}$ is the parameter to be learned, whose dimension is twice the dimension of $\mathbf{d}_m$ or $\mathbf{d}_\Sigma$.

4.3 Attention

The decoder receives the document set vector $\tilde{\mathbf{d}}$ as initial state and generates the output summary from the document set representation. The difference of the decoder to SinABS is that when computing the attention distribution now it should be computed on all the sentences in a document set. Not only the amount of original sentences becomes larger, but also the original sentences come from different documents. Nevertheless, we believe the topic-sensitive PageRank attention mechanism is still able to identify salient sentences, since similar idea in LexRank and TextRank methods achieves good performance on MDS. Therefore, the attention distribution is now computed on

all the input sentences, by conducting the topic-sensitive PageRank algorithm in Eq. 1 and Eq. 2 on all the original sentences.

However, a problem does occur because the amount of original sentences is much larger than that of single document summarization task. Even though the graph-based attention mechanism is still able to rank the relevance and salience of original sentences, the attention distribution will be too disperse and even. This results in that too many sentences are considered to produce the context vector, making the context vector contain too much information. We believe a more concentrated attention distribution will be better. Therefore, when computing the attention weights, only the top K ranked sentences can have attention weights. This can be easily realized by switching the rank scores of sentences not in largest K sentences to minimum value and re-normalizing the attention weights. K is a hyper-parameter.

4.4 Model Tuning

SinABS is trained on the single document summarization corpus - CNN/DailyMail. Although both the CNN/DailyMail corpus and DUC datasets are news data, the reference summaries of the datasets differ much. In order to better adapt the SinABS model on the MDS task, we attempt to fine tune the pre-trained SinABS model, although we have only a few reference summaries for the MDS task. In our approach we tune the decoders of the model. The parameters are the LSTM parameters of the word and sentence decoders, and the weight vector $\mathbf{q}$ in the document set encoder. The loss function and the optimization algorithm are the same with those of the original SinABS model, and we use the cross-entropy loss and the Adam (Kingma and Ba, 2014) algorithm to train the model. To prevent overfitting the training is stopped when performance begins to decrease.

5 Experiments

5.1 Dataset

We conduct experiments on the DUC datasets which are widely used in document summarization. We use the MDS tasks of DUC 2002 and 2004 as test sets, which contain 50 document sets and 59 document sets, respectively. When evaluating on the DUC 2004 dataset, the DUC 2001-2003 and DUC 2005-2007 datasets are used for tuning the model, and DUC 2001, DUC 2003-2007 datasets are used when testing on the DUC 2002 dataset. The MDS tasks of DUC 2005-2007 are query focused summarization, but we ignore the query since these datasets are only used for training. There are on average 10 documents per set in DUC 2004 and 9.58 documents per set in DUC 2002. For the datasets of DUC 2005-2007 we use only the top 10 documents which are most similar to the topic of a document set.

5.2 Implementation

We implement our approach based on the source code and pre-trained model on the CNN/DailyMail corpus provided by Tan et al. (2017). We process the DUC datasets similar to Tan et al. (2017), including tokenizing and lowercasing the text, replacing all digit characters with the "#" symbol and label all name entities with CoreNLP toolkit[1]. The "#" symbols are mapped back to the original digits after decoding according to the context. We also implement our model in Theano[2] based on the SinABS model. K is set to 15 based on developing on the training set.

5.3 Evaluation Metric

ROUGE: We use ROUGE-1.5.5 (Lin and Hovy, 2003) toolkit and report the Rouge-1, Rouge-2 and Rouge-SU4 F1-scores, which has been widely adopted by DUC and TAC for automatic summary quality evaluation. It measured summary quality by counting overlapping units such as the n-gram, word sequences and word pairs between the candidate summary and the reference summary.

Edit distance: In order to test if our model is truly abstractive, instead of simply copying relevant fragments verbatim from the input documents, we compute the word edit-distance between each generated sentence s_i and the most similar original sentence of it, as ed_i, and report the average ED $= \frac{1}{n}\sum_{i=1}^{n} ed_i$.

Considering the significant difference of length between sentences, we also divide the word edit-distance for each generated sentence by its word number w_i as ED/$w = \frac{1}{n}\sum_{i=1}^{n} ed_i/w_i$.

5.4 Baselines

To verify the effectiveness of our approach, we investigate various strategies to adapt SinABS to MDS task for comparison. Since SinABS takes

[1] http://stanfordnlp.github.io/CoreNLP/
[2] https://github.com/Theano/Theano

one document as input but there are multiple input documents in the MDS task, we explore four possible approaches to address this ("ex." indicates extractive method and "ab." indicates abstractive method. SinABS is denoted as Δ).

Single-ab.: One representative document of every document set is selected as the input document to the SinABS model. This is the most straightforward way to adapt single document summarization model to the MDS task. The representative document is chosen by conducting the PageRank (Page et al., 1999) algorithm on every document set. This baseline is denoted as P.R.+Δ.

Single-ex.+Merge+Single-ab.: Different from selecting one representative document, we also investigate constructing a pseudo document as the input to SinABS. We achieve this by first using extractive single document summarization method to summarize every input document, and then concatenate these summaries to form a new document. The motivation of this strategy is to keep only the important content of original documents, so that the input is both the key information and suitable for SinABS to handle. The methods for extractive summarization are Lead, LexRank, TextRank and Centroid. These four baselines are denoted as Lead/Lex./Text./Cent.+Δ respectively.

Single-ab.+Merge+Single-ab.: Generate the abstractive summary for every original document with SinABS. Then the abstractive summaries are concatenated to form a pseudo document, as the input to SinABS again. The difference from Single-ex.+Merge+Single-ab. is that no extractive methods are required. This baseline his denoted as Δ+Δ.

Single-ab.+Multi-ex.: Generate the summary for every original document, then summarize these summaries using some extractive MDS method instead of SinABS to get the final summary. The extractive MDS methods used are Lead, LexRank, TextRank, Centroid and Coverage. Note that Coverage is specially designed for the MDS task, therefore it is not used in Single-ex.+Merge+Single-ab. baselines. These five baselines are denoted as Δ+Lex./Text./Cent./Cov./Lead.

We introduce the extractive MDS methods used in previous baselines as follows. These extractive methods themselves can also be the baselines for comparison.

Lead: This baseline method takes the first sentences one by one in single document or the first document in the document collection, where documents in the collection are assumed to be ordered by name.

Coverage: It takes the first sentence one by one from the first document to the last document in the document collection.

LexRank: LexRank (Erkan and Radev, 2004) computes sentence importance based on the concept of eigenvector centrality in a graph representation of sentences. In this model, a connectivity matrix based on intra-sentence cosine similarity is used as the adjacency matrix of the graph representation of sentences.

TextRank: TextRank (Mihalcea and Tarau, 2004) builds a graph and adds each sentence as vertices, the overlap of two sentences is treated as the relation that connects sentences. Then graph-based ranking algorithm is applied until convergence. Sentences are sorted based on their final score and a greedy algorithm is employed to impose diversity penalty on each sentence and select summary sentences.

Centroid: In centroid-based summarization (Radev et al., 2000) method, a pseudo-sentence of the document called centroid is calculated. The centroid consists of words with TF-IDF scores above a predefined threshold. The score of each sentence is defined by summing the scores based on different features including cosine similarity of sentences with the centroid, position weight and cosine similarity with the first sentence.

Method	R-1	R-2	R-SU4	ED	ED/w
P.R. +Δ	28.3	4.83	8.8	24	0.88
Lead+Δ	31.9	5.85	10.1	30	0.87
Lex. +Δ	31.0	5.52	9.8	25	0.87
Text.+Δ	32.3	5.68	10.4	34	0.89
Cent.+Δ	32.4	6.42	10.4	31	0.90
Δ+Lead	31.5	5.34	9.9	27	0.87
Δ+Cov.	32.4	5.65	10.3	29	0.88
Δ+Lex.	32.7	5.80	10.5	20	0.96
Δ+Text.	32.6	5.96	10.4	32	0.79
Δ+Cent.	31.7	5.44	10.0	43	0.80
Δ+Δ	31.5	5.30	10.0	48	0.88
Our Model	**34.0**	**6.96**	**11.4**	22	1.01

Table 1: Comparison results with abstractive baselines on the DUC 2002 test set.

5.5 Results

Method	R-1	R-2	R-SU4	ED	ED/w
P.R. +Δ	31.7	5.56	10.1	27	0.85
Lead+Δ	31.8	5.74	10.0	28	0.83
Lex. +Δ	32.9	6.28	10.8	33	0.89
Text.+Δ	33.3	6.10	10.7	41	0.90
Cent.+Δ	34.4	6.68	11.1	44	0.93
Δ+Lead	33.2	6.12	10.6	27	0.83
Δ+Cov.	34.4	6.84	11.2	27	0.84
Δ+Lex.	34.0	6.30	11.0	20	0.91
Δ+Text.	34.3	6.71	11.1	35	0.78
Δ+Cent.	32.8	5.77	10.3	44	0.80
Δ+Δ	31.3	4.70	9.6	52	0.88
Our Model	**36.7**	**7.83**	**12.4**	22	1.10

Table 2: Comparison results with abstractive baselines on the DUC 2004 test set.

The comparison results with abstractive baselines are presented in Table 1 and Table 2, respectively. As seen from Table 1 and Table 2, selecting one document as the representation of a document set (Single-ab.) performs poorly. This indicates considering the information of all documents is necessary for MDS task. Generally generating the abstractive summary for every document first and then merging these summaries with extractive MDS methods (i.e. Single-ab.+Multi-ex.) performs slightly better than constructing pseudo single document by extractive summarization methods (i.e. Single-ex.+Merge+Single-ab.). It may be explained that Single-ab.+Multi-ex. keeps the integrity of a document, thus the SinABS model will perform better. Similarly Single-ab.+Merge+Single-ab. does not perform well because the constructed document is much different from a real one. Our system achieves the best performance on both datasets, since our model at the same time keeps the integrity of all original documents and takes into consideration only the salient sentences by ranking all original sentences in the attention mechanism.

The edit distance results verify that our method produces sentences that are quite different from original sentences, indicating the property of abstractive summarization.

Method	Encoder	Attention	Tuning
Model-1	fixed	raw	no
Model-2	fixed	concentrated	no
Model-3	fixed	concentrated	yes
Our Model	learned	concentrated	yes

Table 3: Details of model validation.

Method	R-1	R-2	R-SU4	ED	ED/w
Model-1	31.7	5.89	10.0	42	0.89
Model-2	32.2	6.16	10.3	43	0.90
Model-3	32.8	6.42	10.8	24	1.06
Our Model	**34.0**	**6.96**	**11.4**	22	1.01

Table 4: Model validation results on DUC 2002.

5.6 Model Validation

We conduct ablation experiments to verify the effectiveness of our model. Since we make three extensions to the SinABS model, namely the learned weights in the document set encoder, the attention mechanism and the tuning of the model. We validate their effect with three baseline models, by each changes one of the three parts. The difference of the three baselines are listed in Table 3. Model-1 is the simplest model without tuning, which uses a fixed weight vector $\mathbf{w} = [1/m, \ldots, 1/m]$, and uses the raw attention mechanism in Tan et al. (2017). Model-2 verifies the effectiveness of making the attention distribution more concentrated on the 15 most salient sentences. Model-3 verifies tuning the decoder but not the document set encoder. Compared with Model-3, our model further learns different weights for different documents in the document encoder. Results are presented in Table 4 and Table 5. As seen from Table 4 and Table 5, all the three strategies considerably improve the performance, validating how to better adapt single abstractive summarization model to the MDS task.

Method	R-1	R-2	R-SU4	ED	ED/w
Model-1	33.9	6.64	11.0	45	0.90
Model-2	34.1	7.10	11.2	49	0.91
Model-3	34.9	7.52	11.8	21	1.06
Our Model	**36.7**	**7.83**	**12.4**	22	1.10

Table 5: Model validation results on DUC 2004.

Method	Coherence	N.R.	Readability
Lead+Δ	2.32	2.74	2.71
Cent.+Δ	2.63	2.84	3.29
Δ+Cov.	2.30	3.53	2.92
Δ+Text.	3.18	3.75	3.34
Δ+Δ	2.23	2.57	2.57
Our Model	**3.76**	**3.92**	**4.08**

Table 6: Human evaluation results on 20 samples from the DUC 2002 and DUC 2004 datasets.

5.7 Human Evaluation

We also conduct human evaluation to evaluate the linguistic quality of the generated abstractive summaries, and compare with some significant baselines. We randomly sample 10 document sets from the DUC 2002 dataset and another 10 document sets from the DUC 2004 dataset for human evaluation. Three volunteers who are fluent in English were asked to perform manual ratings on three dimensions: Coherence, Non-Redundancy (N.R. for short) and Readability. The ratings are in the format of 1-5 numerical scores (not necessarily integral), with higher scores denote better quality. The average results are shown in Table 6. It can be observed that our system also outperforms other abstractive summarization approaches in human evaluation, achieving good coherence and readability.

5.8 Case Study

We show the abstractive summaries generated for an example from the DUC 2004 test set in Figure 3. It can be seen that the abstractive summaries generally read well, and has the potential to better convey the key information of original documents.

6 Conclusion and Future Work

Abstractive Multi-Document Summarization (MDS) is still a challenging and open problem. Although sequence-to-sequence models have achieved great progress in single document summarization, the demands of large amount of training data makes it hard to apply it to the MDS task. In this paper, we address this problem from another direction, that we investigate leveraging pre-trained successful single document summarization model to the MDS task. We propose a framework to realize this goal by adding a document set encoder into the hierarchical framework,

Lead+Δ:
politics , opposition leader hun sen and the prime minister were ousted <eos> in the u.s. khmer rouge , the government 's prime minister 's ruling party has had a lengthy majority of its leader in cambodia 's human rights record . <eos> of the country 's opposition party leaders and opposition members , the government have become prime minister <eos> of parliament with its prime minister , the presidency of the khmer rouge has been ruled out by the government 's leading opposition <eos> two political parties previously clashed with the government 's top two parties <eod>

Δ+Text:
king hun sen on tuesday praised by cambodia 's top two political parties, a coalition government led by prime minister <eos> in a short letter sent to news agencies, the king said he had received copies of fiscal and his cambodian people 's party in the government. <eos> cambodia 's leading opposition party ruled out sharing the top position in the presidency of parliament with its opposition <eos> in talks between the two party opposition bloc and the cambodian people 's party to form a new government. <eod>

Our System:
opposition leader cambodian people 's party won the election. <eos> in the u.s. , they were arrested in bangkok and charged with a lengthy coup of human rights . <eos> leading opposition party , the top position in parliament with its political rights , was arrested in bangkok , insisting it would lead to the presidency of thailand 's leading government . <eos> prime minister , political parties won a three - month agreement and agreed to a coalition government . <eos> the government would not end in a new coup vote and his arrest was rejected by the parties of parliament . <eod>

Figure 3: Example of generated abstractive summary by our system.

and we propose three strategies to further improve the model performance. Experimental results demonstrate our approach is able to achieve promising results on standard MDS datasets.

Our study is still primary effort towards abstractive MDS. Future work we can do includes alleviating the requirement of a good pre-trained abstractive summarization model, designing better attention mechanism for MDS, and investigating our approach based on other model architectures.

Acknowledgment

This work was supported by National Natural Science Foundation of China (61772036, 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 their helpful comments. Xiaojun Wan is the corresponding author.

References

P. V. S. Avinesh and Christian M. Meyer. 2017. Joint optimization of user-desired content in multi-document summaries by learning from user feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1353–1363.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly

learning to align and translate. *arXiv preprint*, arXiv:1409.0473.

Tal Baumel, Matan Eyal, and Michael Elhadad. 2018. Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. *arXiv preprint*, arXiv:1801.07704.

Florian Boudin, Hugo Mougard, and Benoît Favre. 2015. Concept-based summarization using integer linear programming: From concept pruning to multiple optimal solutions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1914–1918.

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.

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*, pages 93–98.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artif. Intell. Rev.*, 47(1):1–66.

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

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

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, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1004–1013.

Jiwei Li, Minh-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 of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1106–1115.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 71–78.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint*, arXiv:1801.10198.

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.

Kathleen McKeown, Judith Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multidocument summarization by reformulation: Progress and prospects. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, July 18-22, 1999, Orlando, Florida, USA.*, pages 453–460.

Kathleen McKeown and Dragomir R. Radev. 1995. Generating summaries of multiple news articles. In *SIGIR'95, Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington, USA, July 9-13, 1995 (Special Issue of the SIGIR Forum)*, pages 74–82.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

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 *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 3349–3358. ACL.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, 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, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 280–290. ACL.

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

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint*, arXiv:1705.04304.

Maxime Peyrard and Judith Eckle-Kohler. 2017. Supervised learning of automatic pyramid for optimization-based multi-document summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1084–1094.

Dragomir R Radev, Hongyan Jing, and Malgorzata Budzikowska. 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization-Volume 4*, pages 21–30. Association for Computational Linguistics.

Alexander M. 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, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 379–389.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2015. Joint matrix factorization and manifold-ranking for topic-focused multi-document summarization. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 987–990. ACM.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1171–1181. Association for Computational Linguistics.

Yuhui Tao, Shuigeng Zhou, Wai Lam, and Jihong Guan. 2008. Towards more effective text summarization based on textual association networks. In *Fourth International Conference on Semantics, Knowledge and Grid, SKG '08, Beijing, China, December 3-5, 2008*, pages 235–240.

Xiaojun Wan and Jianwu Yang. 2008. Multi-document summarization using cluster-based link analysis. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, pages 299–306.

Dingding Wang, Shenghuo Zhu, Tao Li, Yun Chi, and Yihong Gong. 2011. Integrating document clustering and multidocument summarization. *TKDD*, 5(3):14:1–14:26.

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, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 233–243.

Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. 2017. Recent advances in document summarization. *Knowledge and Information Systems*, 53(2):297–336.