Augmenting Neural Sentence Summarization through Extractive Summarization

Junnan Zhu[†], Long Zhou[†], Haoran Li[†], Jiajun Zhang[†],
Yu Zhou[†],Chengqing Zong[†‡]

[†]University of Chinese Academy of Sciences
National Laboratory of Pattern Recognition, CASIA
[‡]CAS Center for Excellence in Brain Science and Intelligence Technology
{junnan.zhu,jjzhang,yzhou,cqzong}@nlpr.ia.ac.cn

Abstract. Neural sequence-to-sequence model has achieved great success in abstractive summarization task. However, due to the limit of input length, most of previous works can only utilize lead sentences as the input to generate the abstractive summarization, which ignores crucial information of the document. To alleviate this problem, we propose a novel approach to improve neural sentence summarization by using extractive summarization, which aims at taking full advantage of the document information as much as possible. Furthermore, we present both of streamline strategy and system combination strategy to achieve the fusion of the contents in different views, which can be easily adapted to other domains. Experimental results on CNN/Daily Mail dataset demonstrate both our proposed strategies can significantly improve the performance of neural sentence summarization.

1 Introduction

Text summarization is a task to condense a piece of text to a shorter version that preserves its meaning. There are two broad approaches for summarization: extractive summarization [6, 10] and abstractive summarization [3, 7]. Extractive methods extract parts of a document (usually whole sentences) to form a summary in two steps: sentence ranking and sentence selection. While abstractive methods generate and paraphrase sentences not featured in the source text – as a human-written summary usually does.

Due to the difficulty of abstractive summarization [3, 7], great majority of previous works focus on extractive summarization [6, 10]. Recently, sequence-to-sequence (seq2seq) models [19] provide an effective new way for abstractive summarization [11, 15]. These works all use lead (first) sentence-headline pairs to train seq2seq sentence summarization model. However, previous works take no account of the fact that lead sentences can not offer sufficient information for summarization. Alfonseca et al. [1] also indicate that the most important information is usually distributed in multiple sentences in a document.

To address this problem, we propose in this paper a new approach to boost neural sentence summarization by using extractive summarization that can extract most important sentences of document. We further present two strategies to fuse the extracted contents in different views, which aims at leveraging the document information as much as possible. On one hand, streamline strategy is designed to concatenate and compress each summary of several extractive methods to get an intermediate summary. Then we can obtain the the final summary by a seq2seq model. On the other hand, we propose a neural system combination strategy for sentence summarization, which is adapted from neural system combination for machine translation [23]. It employs a hierarchical attention mechanism to utilize document information.

Specifically, we make the following contributions in this paper:

- We present a simple but effective streamline strategy to leverage the content information provided by extractive methods for neural sentence summarization.
- We propose a neural system combination strategy for sentence summarization, which takes the summaries of several extractive systems as input and produces the final summary.
- Experiments on CNN/DailyMail corpus show that two proposed strategies achieve substantial improvements over strong baselines.

2 Neural Sentence Summarization

Rush et al. [15] were the first to apply the seq2seq framework to abstractive sentence summarization, which provides an effective new way for text generation. Rush et al. [15] assume that the first sentence contains the most important information, therefore they train a seq2seq neural network on first sentences and headlines. In this section we briefly introduce the seq2seq model and its encoder-decoder framework.

2.1 Sequence-to-sequence Model

Seq2seq model consists of two Recurrent neural networks (RNN) [4]: an encoder that processes a input sequence $X = (x_1, x_2, ..., x_m)$ and maps it to a sequence of vector representation $h = (h_1, h_2, ..., h_m)$, and a decoder that generates the output sequence of symbols $Y = (y_1, y_2, ..., y_n)$ from the vector representation. Specifically, the encoder maps a variable-length source sequence to a fixed-length vector, and the decoder maps the vector representation back to a variable-length target sequence of symbol. The two networks are trained jointly to maximize the conditional probability of target sequence given a source sequence:

$$P(Y|X;\theta) = \prod_{j=1}^{N} P(y_j|\{y_1, y_2, ..., y_{j-1}\}, h; \theta) \tag{1}$$

2.2 Encoder

The role of the encoder is to read the input sequence and map it to hidden representation. The output sequence is depended not only on the previous predicted word but also on the previous hidden representation. In this paper, we use a bidirectional Gated Recurrent Units (BiGRU)[4].

The BiGRU consists of a forward GRU and a backward GRU. The forward GRU reads the input from left to right, while the backward reads the input reversely. Then forward hidden representation will be concatenated to the backward hidden representation to get the basic sentence representation.

2.3 Decoder

The decoder reads the previous predicted word and the previous context vector to predict next word. We use GRU with attention as the decoder to produce the output sequence. Attention mechanism [2] can make the decoder focus on the different positions of the input. In this paper, we compute the context vector c_j for current time step j by the concatenate attention mechanism [9], which matches the current decoder state s_j with each encoder hidden state h_i to get an importance score. The score is then normalized to get the current context vector by the weighted sum:

$$e_{j,i} = v_a^T tanh(W_a \tilde{s}_{j-1} + U_a h_i) \tag{2}$$

$$\alpha_{t,i} = \frac{\exp(e_{j,i})}{\sum_{k=1}^{m} \exp(e_{j,k})} \tag{3}$$

$$c_j = \sum_{i=1}^{m} \alpha_{j,i} h_i \tag{4}$$

The decoder then combines current context vector, previous predicted word embedding, and decoder state to predict current word.

3 Our Models

A document is too long to be the input to the seq2seq model. Recent works tend to limit the length of a document to a fixed length such as 100 or 200 words. However, it will destroy the consistency of the document. Since not all sentences in the source document are important. Therefore we apply extractive summarization method to filter out some less important sentences to improve the performance of sentence summarization.

We propose two strategies to alleviate this problem. The first method is similar to the idea of pipeline, we use three different extractive methods to obtain the most important sentences to form three different summaries. Then we merge the summaries to an intermediate summary under the limit of a fixed length. Lastly, the summary is fed to the seq2seq model to generate the final result. The second one uses different summaries from the perspective of neural system combination. Each summary is encoded independently followed by a unified decoding with a hierarchical attention mechanism.

Our baseline model is similar to Nallapati et al. [11]. Our model consists of a sentence encoder using a BiGRU [4] and an decoder using GRU. First, the bidirectional encoder reads an input sequence $x = (x_1, x_2, ..., x_m)$ and encodes it into a sequence of hidden representations $h = (h_1, h_2, ..., h_m)$. Then the GRU decoder predicts a target sequence $y = (y_1, y_2, ..., y_n)$ with attention to the tailored representation. Each word y_j is predicted based on a recurrent hidden state s_j, a previous predicted word y_{j-1} and a context vector c_j. c_j is calculated by the weighted sum of the annotations h_i.

3.1 Extractive Summarization

We conduct extractive methods to select the most salient sentences from the input document. The advantage of extractive methods is the guarantee of correctness of grammar. Extractive methods have also been studied for a long time. We employ three typical extractive methods to produce the summaries of input documents. The length limit of summaries is set to 50 words. The extractive summarization methods are:

Submodular: Submodular method performs summarization by maximizing submodular functions under a budget constraint. The submodularity of the coverage, diversity and non-redundancy can be reflected by a series of submodular functions. We use the same two submodular functions as [22] for extractive summarization.

LexRank: LexRank [6] is inspired by PageRank [13] algorithm. A graph is constructed by creating vertices representing sentences in the document. Edges between sentences are obtained based on cosine similarity of TF-IDF vectors. It computes sentence score based on the concept of eigenvector centrality in the graph representation.

LSA: LSA [18] uses Singular Value Decomposition (SVD) to acquire the semantic meaning of sentences. It can generate concept dimensions which are orthogonal to each other, and then picks the most salient sentences from each dimensions.

3.2 Fusion of Extractive summaries

In order to construct text of limited length as input to seq2seq model but also to ensure the consistency of the text, we first use extractive summarization methods to extract the most important sentences. Each summarization method represents a different view. Submodular method treats summarization as an optimization problem. Lexrank uses graph models to tackle this problem. LSA tries to improve the summarization from the perspective of matrix decomposition. There are two benefits for using three different methods. First, this can increase the diversity of the input. Second, different methods bring us different views of information so that it can be attractive to fuse these information. We then propose two strategies to achieve fusion of three extractive summaries.

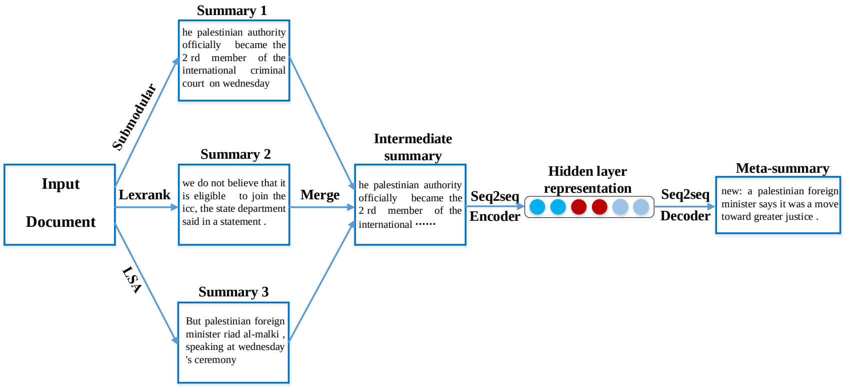

Summary 1
he palestinian authority officially became the 2 rd member of the international criminal court on wednesday
Submodular
Summary 2
we do not believe that it is eligible to join the icc, the state department said in a statement .
Input
Document
Lexrank
LSA
Summary 3
But palestinian foreign minister riad al-malki , speaking at wednesday 's ceremony
Merge
Intermediate summary
he palestinian authority officially became the 2 rd member of the international ……
Seq2seq
Encoder
Hidden layer representation
Seq2seq
Decoder
Meta-summary
new: a palestinian foreign minister says it was a move toward greater justice .

Fig. 1. The framework of our Streamline strategy

Streamline Strategy As depicted in Figure 1, the input text is summarized to three different summaries by three extractive summarization methods. Then we propose two ways to merge these summaries to intermediate summary. One is simply concatenating these summaries; the other employs Round-robin (RR) scheme to compress the summaries. RR scheme picks the first sentence from the first summary and then the second and so no until the summary length is reached. We apply this scheme due to its efficiency and low cost of time. Lastly, the intermediate summary is fed into seq2seq model to generate the meta-summary.

Neural System Combination Strategy In addition to fusion of various summaries in the source, we can also fuse these extractive summaries in the multi-source seq2seq model. Inspired by neural system combination for machine translation [23], which aims at combining the advantages of different machine translation systems through a multi-source model, we propose a neural system combination strategy for sentence summarization.

As depicted in Figure 2, each encoder encodes summary independently to hidden vector representation. However, decoder must be adapted to three inputs with a hierarchical attention mechanism.

We illustrate encoder-decoder for neural system combination in Figure 3. The network can take as input the results of extractive summarization or abstractive summarization. Extractive summaries have good readability, while abstractive methods such as neural network can generate fluent sentences. It is very attractive to combine both of these advantages. Therefore we attempt to use neural system combination to fuse different summaries to achieve complementary effects. Here, we use summarization results to detail the model.

At time j, the state s_{j-1} meets the previous prediction y_{j-1} to transfer to an intermediate state $\tilde{s}_{j-1}$, which can be calculated as follows:

$$s_j = GRU(\tilde{s}_{j-1}, c_j) \tag{5}$$

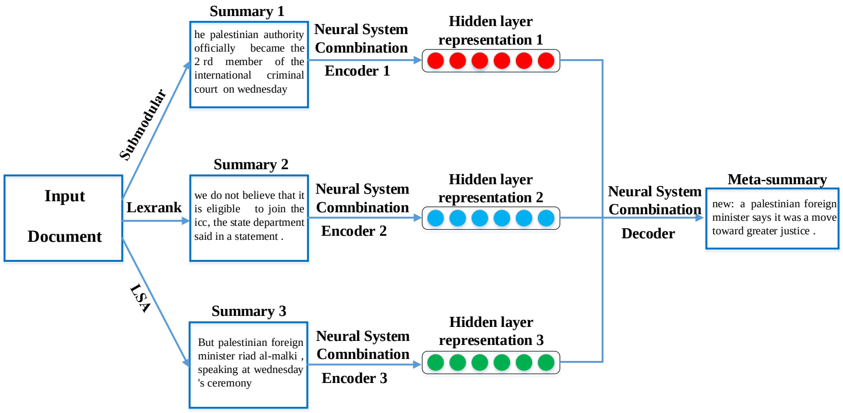

Input
Document
Submodular
Lexrank
LSA
Summary 1
he palestinian authority officially became the 2 rd member of the international criminal court on wednesday
Summary 2
we do not believe that it is eligible to join the icc, the state department said in a statement .
Summary 3
But palestinian foreign minister riad al-malki , speaking at wednesday 's ceremony
Neural System Comnbination
Encoder 1
Neural System Comnbination
Encoder 2
Neural System Comnbination
Encoder 3
Hidden layer representation 1
Hidden layer representation 2
Hidden layer representation 3
Neural System Comnbination
Decoder
Meta-summary
new: a palestinian foreign minister says it was a move toward greater justice .

Fig. 2. encoder-decoder for neural system combination strategy

$$\tilde{s}_{j-1} = GRU(s_{j-1}, y_{j-1}) \tag{6}$$

where y_{j-1} represents the word embeddings of the previous word. c_{ja}, c_{jb}, and c_{jc} represent the context vectors of different encoders, attention weight α_{ji} is computed as described in 4. The attention model calculates c_j as weighted sum of three summarization context vectors, as described in the red box in 3:

$$c_j = \sum_{k=1}^{K} \beta_{jk} c_{jk} \tag{7}$$

where k is the number of summarization systems, and β_{jk} is calculated as follows:

$$\beta_{jk} = \frac{exp(s_j c_{jk})}{\sum_{\tilde{k}} exp(s_j c_{j\tilde{k}})} \tag{8}$$

In order to keep consistency in training and testing, we use the similar training data simulation strategy as [23] when train the single seq2seq system. We select most of the training data, such as two-thirds of data, to train the seq2seq model. And the trained model is used to transform the rest of training data to the summaries. Then re-divide the training data and repeat the above steps until all the training data is summarized. Since extractive methods we use are all unsupervised, there is no need to do this step for extractive summarization.

4 Experiment

4.1 Dataset

We use the CNN/Daily Mail dataset [1] [8, 11], which contains online news articles (781 tokens on average) paired with multi-sentence highlight as summaries

[1] http://cs.nyu.edu/ kcho/DMQA/

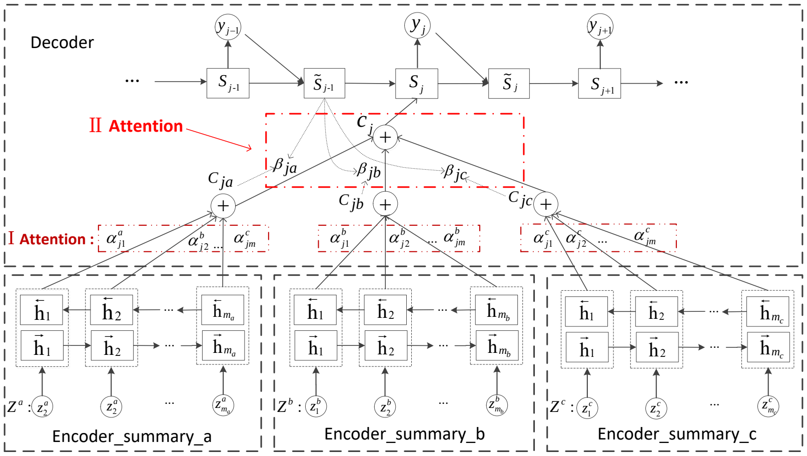

Decoder
y_{j-1}
y_j
y_{j+1}
…
s_{j-1}
$\tilde{s}_{j-1}$
s_j
$\tilde{s}_j$
s_{j+1}
…
Ⅱ Attention
c_j
β_{ja}
β_{jb}
β_{jc}
c_{ja}
c_{jb}
c_{jc}
Ⅰ Attention :
α^a_{j1} α^b_{j2} … α^c_{jm}
α^b_{j1} α^b_{j2} … α^b_{jm}
α^c_{j1} α^c_{j2} … α^c_{jm}
$\overleftarrow{h}_1$ $\overleftarrow{h}_2$ … $\overleftarrow{h}_{m_a}$
$\overrightarrow{h}_1$ $\overrightarrow{h}_2$ … $\overrightarrow{h}_{m_a}$
$\overleftarrow{h}_1$ $\overleftarrow{h}_2$ … $\overleftarrow{h}_{m_b}$
$\overrightarrow{h}_1$ $\overrightarrow{h}_2$ … $\overrightarrow{h}_{m_b}$
$\overleftarrow{h}_1$ $\overleftarrow{h}_2$ … $\overleftarrow{h}_{m_c}$
$\overrightarrow{h}_1$ $\overrightarrow{h}_2$ … $\overrightarrow{h}_{m_c}$
Z^a : z^a_2 z^a_2 … $z^a_{m_a}$
Z^b : z^b_1 z^b_2 … $z^b_{m_b}$
Z^c : z^c_1 z^c_2 … $z^c_{m_c}$
Encoder_summary_a
Encoder_summary_b
Encoder_summary_c

Fig. 3. The framework of our neural system combination strategy

(3.75 sentences or 56 tokens on average). We use open-source scripts supplied by Abigail et al. [16] to obtain the same non-anonymized version of the data. After filtering out data that article text is missing, we obtain 287,113 training pairs, 13,368 validation pairs and 11,490 test pairs. We use the first highlight as our gold label.

4.2 Implementation

The articles and the summaries in the dataset we obtained are all lowercased and tokenized by Stanford Corenlp toolkit [2]. We replace all the digit characters with # similar to Rush et al. [15]. We illustrate different methods as follows:

(1) **Baseline**

ABS Rush et al.[15] use an attentive Convolutional Neural Network (CNN) encoder and a Neural Network Language Model (NNLM) decoder to do this task. We trained this baseline with its released code[3].

Seq2seq+attn We implement a sequence-to-sequence model with attention based on the latest implementation of attention-based NMT[4].

abstractive model Nallapati et al. [11] used both RNN as encoder and decoder, and added some features such as POS, named-entities and TF-IDF, into encoder.

(2) **Extractive**

In order to compare with extractive methods, we also directly use three extractive methods to obtain the summaries. The reference summaries have 14.89 tokens on average in test set. We limit the length of output to 20 tokens in order to prevent zero output due to too short length limit.

[2] https://stanfordnlp.github.io/CoreNLP/
[3] https://github.com/facebook/NAMAS
[4] https://github.com/nyu-dl/dl4mt-tutorial

(3) **Streamline**

Sub+seq2seq We use the summary which is summarized by Submodular method as the input to the seq2seq model above. The length limit of all extractive summaries are all set to 50 words and so do Lex+seq2seq and LSA+seq2seq.

Sub+Lex+LSA We simply concatenate these three summaries. Since the length of each summary is less than 50 words, the length of the intermediate summary does not exceed 150 words.

Sub+Lex+LSA+RR We merge the summaries of the three extractive method by the Round-robin [20] scheme. We deduplicate the sentences in this step. To fully fuse all the important information, we set the length limit of the intermediate summary to 100 words. If the length limit is too short, the intermediate summaries usually come from the first two summaries so as not to achieve the fusion.

(4) **Neural System Combination**

Sub+Lex+LSA We employ the Neural System Combination (NSC) to map the three inputs to the output.

Neural+Sub+Lex We use the seq2seq+attn model as a single system to the NSC. And the other two inputs use the summaries of Submodular and LexRank, so do Neural+Sub+LSA and Neural+Lex+LSA.

4.3 Training Details

The hyper-parameters used in our model are described as follows. For all experiments, we use a vocabulary of 50K words for source and 30K words for target. We set word embedding size to 128 and all GRU hidden state sizes to 256. We use dropout [17] with the probability of 0.5 . We do not pretrain the word embeddings, they are learned from scratch during training. We use Adadelta[21] with learning rate 0.0001 to update parameters in the network. We also apply gradient clipping [14] with range $[-1, 1]$ during training. We use mini-batch size 64 to both speed up the training and converge quickly. We employ beam search to generate multiple summary candidates to get better results. The beam size is set to 10. We use a single NVIDIA TITAN X to train our models. We trained all our models for about 30 epochs. All the models can be trained in 24 hours. And single seq2seq model using one input source can be trained in 12 hours.

4.4 Experimental Results

We evaluate our models with the standard ROUGE metric, reporting the F1 scores for ROUGE-1, ROUGE-2 and ROUGE-L (which measure unigram-overlap, bigram-overlap, and longest common sequence between the reference summary and summary to be evaluated respectively). We use the files2rouge package [5] to obtain our ROUGE scores. Our results are given in Table 1.

It is clear from Table 1 that seq2seq model with summary summarized by Submodular method as input achieves higher than the other single seq2seq

[5] https://github.com/pltrdy/files2rouge

Table 1. Summarization results (ROUGE F1 score) for different sentence summarization strategies or neural system combination methods. Sub, Lex, and LSA denote Submodular, Lexrank and LSA respectively. **Best** results per category are highlighted.

	Model	ROUGE-1	ROUGE-2	ROUGE-L
Baseline	ABS [15]	13.85	4.13	12.64
	Seq2seq+attn	15.66	4.38	14.30
	abstractive model [11]	**17.58**	**5.45**	**16.22**
Extractive	Sub	**18.86**	**6.77**	**16.56**
	Lex	14.98	3.93	12.92
	LSA	13.12	3.06	11.32
Streamline	Sub+seq2seq	17.18	5.25	15.82
	Lex+seq2seq	10.70	1.94	9.74
	LSA+seq2seq	9.25	1.37	8.47
	Sub+Lex+LSA	20.46	7.39	18.61
	Sub+Lex+LSA+RR	**23.06**	**9.29**	**21.08**
Neural System Combination	Sub+Lex+LSA	17.24	5.28	15.82
	Neural+Sub+Lex	**21.55**	**8.16**	**19.90**
	Neural+Sub+LSA	20.44	7.55	18.89
	Neural+Lex+LSA	19.81	7.19	18.38

model. Compare extractive methods with streamline strategy, we have noticed that it is difficult to improve the performance by simply making the extractive summaries as the input to the seq2seq model. Abigail et al. [16] find that their lead-3 baseline outperforms all other methods in their experiment. They attribute it to the reason that news articles tend to be structured with the most important information at the start. This is consistent with Rush's [15] assumption. We conduct an experiment on test data to analyze the position of the extractive summary sentences statistically.

Table 2. Statistics of the position of the three extractive summaries. Position denotes the average position of summary sentences and 1 denotes the first sentence of the article. Lead (%) denotes the percentage of lead sentences (first three sentences) of summary sentences.

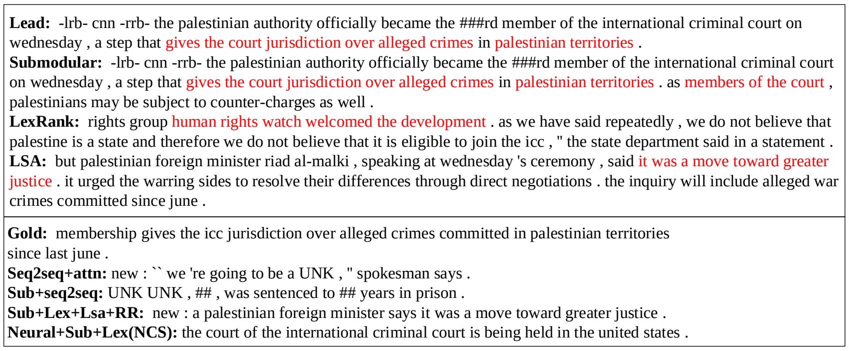
Metric	Submodular	Lexrank	LSA
Position	9.67	15.93	17.64
Lead (%)	49.04	13.99	10.96

As shown in Table 2, Submodular summaries cover more lead sentences than the others. The former average position locates, the higher performance of seq2seq model achieves. However, it is not to say that non-lead sentences are useless. The score of Sub+Lex+LSA (Streamline) shows these different sentences can still benefit this task. Although the length of input becomes longer, it does not mean that input contains more valuable information. The input of

Sub+Lex+LSA (Streamline) contains the input in Sub+Lex+LSA+RR (Streamline), but the latter scores higher. It illustrates that filtering out some redundant information can improve the performance of our model.

From the experimental results of neural system combination, we find that the fusion of the three extractive summaries achieves the lowest score. It may be due to the redundancy of multiple extractive summaries. Extractive summarization selects contents from source text in sentence level, it can meet the diversity requirements of system combination. Therefore the performance of the neural system combination is proportional to the amount of information in summaries. From the analysis above, Submodular summaries contain the most informative content, then Lexrank, and finally LSA. Therefore Neural+Sub+Lex (NSC) achieves the highest score among all the methods under NSC strategy.

The same is fusion of several system results, Sub+Lex+LSA+RR (Streamline) outperforms all the methods under NSC strategy. Since we have deduplicated the sentences in Round-robin step and all sentences come from a single document, there does not exist residual redundancy information. This further demonstrates the importance of removing redundant information. From the comparison of Sub+Lex+LSA (Streamline) and Sub+Lex+LSA(NSC), our conclusion is that Streamline strategy is more effective than NSC strategy. And the Streamline strategy requires much less time to train the network since there is only a single seq2seq model. Some of examples are given in Figure 4.



Lead: -lrb- cnn -rrb- the palestinian authority officially became the ###rd member of the international criminal court on wednesday , a step that gives the court jurisdiction over alleged crimes in palestinian territories .
Submodular: -lrb- cnn -rrb- the palestinian authority officially became the ###rd member of the international criminal court on wednesday , a step that gives the court jurisdiction over alleged crimes in palestinian territories . as members of the court , palestinians may be subject to counter-charges as well .
LexRank: rights group human rights watch welcomed the development . as we have said repeatedly , we do not believe that palestine is a state and therefore we do not believe that it is eligible to join the icc , " the state department said in a statement .
LSA: but palestinian foreign minister riad al-malki , speaking at wednesday 's ceremony , said it was a move toward greater justice . it urged the warring sides to resolve their differences through direct negotiations . the inquiry will include alleged war crimes committed since june .

Gold: membership gives the icc jurisdiction over alleged crimes committed in palestinian territories since last june .
Seq2seq+attn: new : `` we 're going to be a UNK , " spokesman says .
Sub+seq2seq: UNK UNK , ## , was sentenced to ## years in prison .
Sub+Lex+Lsa+RR: new : a palestinian foreign minister says it was a move toward greater justice .
Neural+Sub+Lex(NCS): the court of the international criminal court is being held in the united states .

Fig. 4. Summaries generated by extractive methods and our model. Lead denotes the lead sentence in the source document. Gold denotes our gold summary. The last four are the most representative methods in our experiments. Some of the important information in the source is shown in red.

As shown in Figure 4, the single seq2seq models result in poor output though the inputs contain some contents which match the output. We can also see that the summaries generated by our two strategies actually leverage the information from three extractive summaries, and are more fluent. There are less UNKs in the summaries in the last two summaries. This is consistent with observation in Zhou [23]. It also further illustrates the effectiveness of our model.

5 Related Work

Human-written summaries are highly abstracted and seldom consist of reproduction of original sentences from the document. Previous work [3, 7] which focused on abstractive summarization has employed sentence fusion to construct a sentence whose fragments come from different source sentences.

With the emergence of deep learning, researchers have considered the framework as a fully data-driven alternative to abstractive summarization. Rush [15] firstly apply neural network to abstractive sentence summarization. They propose leveraging news data in Gigaword [12] corpus to construct large scale parallel corpus for sentence summarizaiton task. Their model consists of an attentive Convolutional Neural Network encoder and an neural network language model decoder, producing state-of-the-art results on Gigaword and DUC datasets. In an extension to this work, Chopra [5] used a similar CNN encoder, but replaced the decoder with an Recurrent Neural Network decoder, producing further improvement on both dataset.

However, their models all take lead sentences as input with assumption that lead sentences carry the most information. This causes most of the information to be lost at the input. It is difficult for neural network to take whole document as input. We propose two strategies that can filter out some less important contents in the text so that it can be processed by neural network and we can leverage most information in the source text indirectly. And the greatest advantage of our approach is that we do not need to make any assumptions about the corpus. In other words, it is general to be adapted to other domains.

6 Conclusion

In this paper, we propose a novel approach to enhance neural sentence summarization by utilizing extractive summarization, which aims at taking full advantage of the document information, instead of lead sentences in conventional neural sentence summarization. Furthermore, we present streamline strategy and system combination strategy to achieve the fusion of the contents in different views. To show the effectiveness of our proposed approaches, we conduct experiments on CNN/DailyMail corpus. Experimental results demonstrate that our strategies achieve significant improvements over strong baselines.

Acknowledgments

The research work has been funded by the Natural Science Foundation of China under Grant No. 61673380, No. 61402478 and No. 61403379.

References

1. Alfonseca, E., Pighin, D., Garrido, G.: HEADY: news headline abstraction through event pattern clustering. In proceedings of ACL (2013)

2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In proceedings of ICLR (2015)
3. Barzilay, R., McKeown, K.R.: Sentence fusion for multidocument news summarization. Comput. Linguist
4. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In proceedings of EMNLP (2014)
5. Chopra, S., Auli, M., Rush., A.M.: Abstractive sentence summarization with attentive recurrent neural networks. North American Chapter of the Association for Computational Linguistics (2016)
6. Erkan, G., Radev, D.R.: LexRank: Graph-based Lexical Centrality as Salience in Text Summarization, vol. 22. Journal of Qiqihar Junior Teachers College (2011)
7. Filippova, K., Strube, M.: Sentence fusion via dependency graph compression. In proceedings of EMNLP (2008)
8. Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P.: Teaching machines to read and comprehend. In proceeding sof NIPS (2015)
9. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In proceedings of EMNLP (2015)
10. Nallapati, R., Zhai, F., Zhou, B.: SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents. In proceedings of AAAI (2017)
11. Nallapati, R., Zhou, B., glar Gulcehre, C.: Abstractive text summarization using sequence-to-sequence rnns and beyond. Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning (2016)
12. Napoles, C., Gormley, M., Durme, B.V.: Annotated gigaword. Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (2012)
13. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web (1999)
14. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In proceedings of ICML (2013)
15. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. In Proceedings of EMNLP (2015)
16. See, A., J.Liu, P., Manning, C.D.: Get To The Point: Summarization with Pointer-Generator Networks. In proceedings of ACL (2017)
17. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning (2014)
18. Steinberger, J., Jezek, K.: Text summarization and singular value decomposition. In proceedings of ADVIS (2004)
19. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In Proceedings of Neural Information Processing Systems (2014)
20. Wang, D., Li, T.: Weighted consensus multi-document summarization. Information Processing and Management (2012)
21. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. CoRR (2012)
22. Zhang, J., Wang, T., Wan, X.: PKUSUMSUM: A Java Platform for Multilingual Document Summarization. In proceedings of COLING (2016)
23. Zhou, L., Hu, W., Zhang, J., Zong, C.: Neural System Combination for Machine Translation. In proceedings of ACL (2017)