

# An Unsupervised Multi-Document Summarization Framework Based on Neural Document Model

Shulei Ma

Zhi-Hong Deng\*

Yunlun Yang

Key Laboratory of Machine Perception (Ministry of Education)

School of Electronics Engineering and Computer Science

Peking University, Beijing 100871, China

mashulei@pku.edu.cn, zhdeng@cis.pku.edu.cn

incomparable-lun@pku.edu.cn

## Abstract

In the age of information exploding, multi-document summarization is attracting particular attention for the ability to help people get the main ideas in a short time. Traditional extractive methods simply treat the document set as a group of sentences while ignoring the global semantics of the documents. Meanwhile, neural document model is effective on representing the semantic content of documents in low-dimensional vectors. In this paper, we propose a document-level reconstruction framework named *DocRebuild*, which reconstructs the documents with summary sentences through a neural document model and selects summary sentences to minimize the reconstruction error. We also apply two strategies, sentence filtering and beamsearch, to improve the performance of our method. Experimental results on the benchmark datasets DUC 2006 and DUC 2007 show that *DocRebuild* is effective and outperforms the state-of-the-art unsupervised algorithms.

## 1 Introduction

Multi-document summarization aims at capturing the important information of a set of documents related to the same topic and presenting it in a brief, representative, and pertinent summary. Most existing researches focus on extraction-based methods, in which sentences are selected from the original document set.

Typically, two kinds of unsupervised models are used for sentence selection. One is based on sentence ranking, which uses methods such as clustering (Lin and Hovy, 2002; Radev et al., 2004), PageRank (Erkan and Radev, 2004; Mihalcea and Tarau, 2005) and topic modeling (Harabagiu and Lacatusu, 2005; Wang et al., 2008), to rank the sentences. Considering that top-ranked sentences tend to convey much redundant information, additional strategies are usually applied to reduce redundancy when selecting sentences. This kind of methods usually need to weigh between relevance and redundancy, which may be hard to balance.

The other is based on sparse reconstruction (He et al., 2012; Liu et al., 2015; Yao et al., 2015), which selects a sparse subset of the sentences that can linearly reconstruct all the sentences in the original document set. This kind of methods has a good motivation but also weaknesses in their hypotheses. First, reconstructing single sentences may lose the global information of documents; Second, there are more reasonable ways than linear combination in reconstruction.

Commonly, in the above methods the document set is treated as a set of sentences and all the operations are carried out on the sentence set, losing the global information of documents. Meanwhile, neural document model (Le and Mikolov, 2014; Li et al., 2015; Lin et al., 2015) is an emerging technique which has made significant progress in capturing semantic information of documents by projecting the text into the low-dimensional continuous distributed representation. It has been applied to natural language processing tasks such as sentiment classification (Tang et al., 2015).

---

\*Zhi-Hong Deng is the corresponding author.

This work is licenced under a Creative Commons Attribution 4.0 International License. License details: <http://creativecommons.org/licenses/by/4.0/>

In this paper, we address the aforementioned problems of existing methods and propose a document-level reconstruction framework based on neural document model, named *DocRebuild*, for multi-document summarization.

Intuitively, a good summary is supposed to reconstruct the main content of the multi-document set. In our model, we first introduce neural document model to represent the content of each document and use averaging to obtain the main content of the document set. The main content is reconstructed by concatenating the summary sentences into a sequence and feeding the summary sequence into the document model. Hence the multi-document summarization is converted into an optimization problem via taking the reconstruction error as the objective function. Summary sentences are selected to minimize this error. Furthermore, two strategies are addressed in selecting sentence to yield a better performance. First, irrelevant sentences are filtered and only a subset of related sentences is reserved as candidate set. Second, a beamsearch algorithm is applied to get a better solution in sentence selection stage.

Our contributions can be concluded as follows:

- We introduce neural document model into multi-document summarization task. As far as we know, no such works have been presented before.
- We propose a document-level reconstruction framework *DocRebuild*, and further adopt two effective strategies to improve the performance of our method.
- The experimental results on two benchmark DUC data sets show that our method outperforms the state-of-the-art unsupervised approaches.

## 2 Related Work

Multi-document summarization has received widespread attention in recent years. Most existing multi-document systems use extraction-based methods, in which sentences are directly selected from the original document set.

The majority of these methods use the idea of sentence-ranking, assigning salient scores to sentences of the original document set and choosing the top sentences to form the summary. Typical methods are the centroid-based methods (Lin and Hovy, 2002; Radev et al., 2004), which score sentences basing on features such as cluster centroids, sentence position and TF-IDF. Besides, graph based models (Erkan and Radev, 2004; Mihalcea and Tarau, 2005) first measure the sentence similarity then use ranking algorithm such as PageRank on the similarity graph to estimate the importance of different sentences. Topics in documents are also discovered to be an effective feature for sentence ranking (Hardy et al., 2002; Harabagiu and Lacatusu, 2005; Wang et al., 2008). Maximum Marginal Relevance (MMR) (Goldstein et al., 1999) is widely used for greedily selecting sentences while considering the tradeoff between relevance and redundancy.

However, it is usually hard to get a good balance between relevance and redundancy. Recently, a couple of works have employed the idea of data reconstruction in the summarization task. DSDR (He et al., 2012) reconstructs each sentence in the document set by a non-negative linear combination of summary sentences then minimizes the reconstruction error. MDS-Sparse (Liu et al., 2015) introduces the diversity constraint and proposes a two-level sparse representation model to reconstruct the sentences in the document set. SpOpt (Yao et al., 2015) follows the sparse representation framework while simultaneously doing sentence selection and compression by adjusting reconstruction coefficients and compression coefficients alternately in optimization.

In this work, neural document model is involved in performing summarization task on document level. With the development of deep learning, some attempts have been made to model documents with neural networks. Le and Mikolov (2014) extends the neural network of word embedding (Mikolov et al., 2013) to learn the document embedding. Li et al. (2015) uses a hierarchical long-short term memory auto-encoder to reconstruct the original document. Lin et al. (2015) proposes a hierarchical recurrent neural network language model to consider sentence history information in word prediction. Tang et al. (2015) presents a convolutional-gated recurrent neural network and applies it to sentiment classification task.

However, few such researches have been reported on document level in multi-document summarization task.

### 3 Proposed Framework

We propose our framework *DocRebuild* in this section. The neural document model is introduced in Section 3.1, the objective function is formulated in Section 3.2, and summary sentences are selected with two effective strategies as shown in Section 3.3. Figure 1 illustrates the framework of *DocRebuild*.

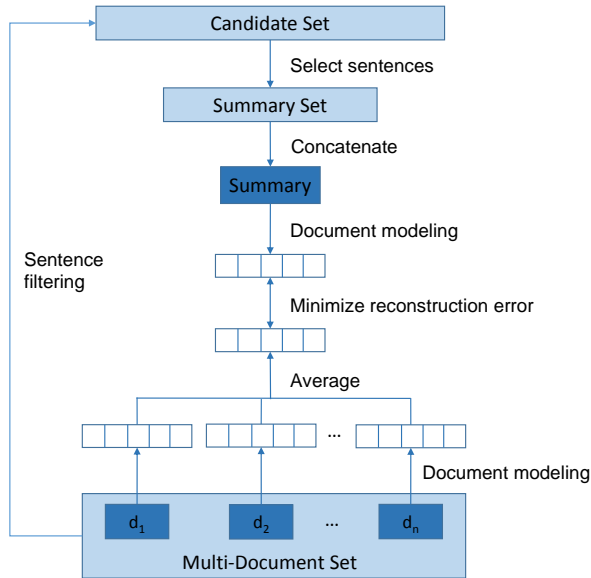


Figure 1: The framework of *DocRebuild*. Light blue boxes represent the sets of documents or sentences, deep blue boxes represent the document or summary sequences. Document modeling process projects document or summary sequences into real-valued vectors as represented by the bars.

#### 3.1 Neural Document Model

Neural document model aims to represent the semantic content of a document with low-dimensional vector representation. As the basis of our framework, it is the key to a good performance. Here we focus on the unsupervised methods and exploit two kinds of unsupervised document models, named Bag-of-Words(BoW) model and Paragraph Vector(PV) model respectively, in our task.

In the BoW model, we simply use the bag-of-words of the document without considering the original order or relationships between neighboring words. Word embedding(Mikolov et al., 2013) has been proven of great significance in most natural language processing tasks in recent years. So we represent each word by its corresponding word embedding and the document is represented as the weighted average of all the words in the document.

Since BoW model is likely to lose the semantic information hidden in the order and composition of words, we introduce a more complex model which takes word order into consideration. PV (Le and Mikolov, 2014) is an unsupervised framework that learns distributed representations for sentences and documents. Compared with other hierarchical document models (Li et al., 2015; Lin et al., 2015) built upon sentences, PV handles texts with various length in a common way, making it possible to measure sentences, short summaries and long documents in the same semantic space.

Figure 2 illustrates the framework of PV model. In this model, every document is mapped to a unique vector, as a column in matrix  $D \in \mathbb{R}^{n \times l}$  and every word is mapped to a unique vector, as a column in matrix  $W \in \mathbb{R}^{m \times l}$ .  $n, m, l$  represent document set size, vocabulary size and dimensionality of vector, respectively.

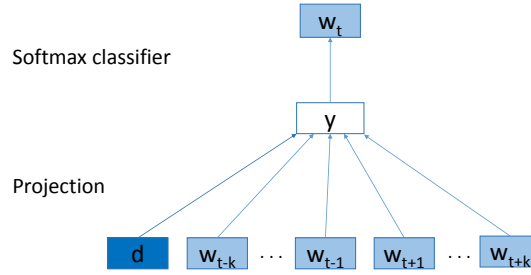


Figure 2: The framework of PV model

It employs a similar idea as the one in Mikolov et al. (2013), to predict the next word given many contexts sampled from the document. More precisely, given a document  $d$  consisting of a sequence of words  $w_1, w_2, \dots, w_T$  and the fixed window size  $k$  of context, the objective of PV model is to maximize the log-likelihood,

$$\mathcal{L} = \sum_{t=1}^T \log \mathbb{P}(w_t | w_{t-k} : w_{t+k}, d) \quad (1)$$

Note that  $w_{t-k} : w_{t+k}$  represents the word sequence from  $w_{t-k}$  to  $w_{t+k}$  except  $w_t$ . The probability  $\mathbb{P}(w_t | w_{t-k} : w_{t+k}, d)$  is defined using the softmax,

$$\mathbb{P}(w_t | w_{t-k} : w_{t+k}, d) = \frac{e^{y_{w_t}}}{\sum_{i=1}^m e^{y_{w_i}}} \quad (2)$$

$\mathbf{y} \in \mathbb{R}^m$  represents un-normalized log probability for all the possible words in vocabulary, computed as,

$$\mathbf{y} = U h(w_{t-k} : w_{t+k}, d) + b \quad (3)$$

where  $U \in \mathbb{R}^{m \times l}$ ,  $b \in \mathbb{R}^m$  are the parameters of softmax classifier and  $h$  stands for the averaging of document vector and word vectors extracted from  $D$  and  $W$ .

During training, parameters  $D, W, U, b$  are randomly initialized and then updated to maximize equation (1) using stochastic gradient descent via backpropagation. Once the model is trained, it can further be employed in predicting representations of the documents not included in the training set.

At inference stage, a new document is fed into the PV model to do the same prediction task as the training documents. But this time only the document vector  $d$  is randomly initialized while other parameters  $W, U, b$  are fixed. Then we update the document vector  $d$  to maximize equation (1) by gradient descent as well. After convergence,  $d$  is taken as the corresponding document vector.

### 3.2 Objective Function

We denote the **multi-document set** as  $D = \{d_1, d_2, \dots, d_n\}$ . All the documents in  $D$  are processed into a group of sentences, defined as the **candidate set** and denoted as  $C = \{s_1, s_2, \dots, s_m\}$ . The sentences selected from  $S$  form the **summary set**, denoted as  $S = \{s_1^*, s_2^*, \dots, s_l^*\}$ , where  $S \subset C$  and  $|S| \ll |C|$ . Note that all the elements in the above sets are sequences of words. Let  $\theta$  denote the required summary length, our task is to select the optimal subset of candidate set  $C$  that composes summary shorter than  $\theta$ .

We consider the multi-document summarization task as a data reconstruction problem. We assume that a good multi-document summary is supposed to reconstruct the main content of the document set. Therefore we focus on two issues: (1) how to represent the main content of the document set, and (2) how to use the summary set to reconstruct the main content. In this work, both issues are resolved by document modeling.

As an example, we randomly choose four document sets and their corresponding human-written summaries in DUC2006 dataset, compute their vector representation with PV model and project the vectors into the two-dimensional space. As shown in figure 3, each color corresponds to a document set and four

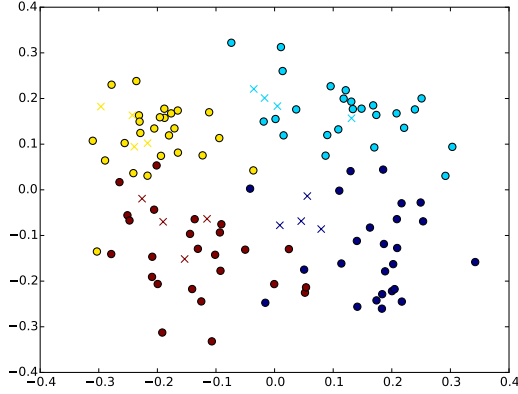


Figure 3: Visualization of document vectors and summary vectors. Documents are represented by circles and summaries are represented by crosses.

summaries, and we find that the centre of summary vectors are close to the centre of document vectors in the same color, implying the effectiveness of averaging.

Hence in our model, each document in the document set is mapped into a vector through document model, then the document vectors are averaged to represent the main content. As for the summary set, all the sentences in  $S$  are sequentially concatenated into a sequence  $S^*$  as the corresponding summary. Then the summary sequence  $S^*$  can be seen as a short document and fed into the document model to reconstruct the main content. Naturally, reconstruction error is applied as objective function and measured by distance between the summary vector and the main content vector. Summary set  $S$  is adjusted to minimize the reconstruction error.

Provided that the document modeling process is represented by  $DM$ , it takes a document or summary  $x$  as input and obtains the semantical vector representation of  $x$ , denoted as  $DM(x)$ . Our reconstruction model is formalized as follows:

$$\begin{aligned} \min_{S \subset C} \quad & \|DM(S^*) - \frac{1}{n} \sum_{i=1}^n DM(d_i)\|_2^2 \\ \text{s.t.} \quad & \text{len}(S^*) \leq \theta \end{aligned} \quad (4)$$

Where  $S^*$  denotes the corresponding summary sequence of summary set  $S$  and  $\text{len}(S^*)$  denotes the length of the summary sequence.

Our formulation is similar to the intuition behind He et al. (2012), but differs from it mainly in two aspects: first, we directly reconstruct the original document set on document level; second, we introduce neural document model to represent and reconstruct documents with the summary sentences.

In addition, multi documents are usually considered to bring the redundancy problem in previous works. Contrarily, multiple documents benefit our method by helping represent documents reliably and capture the main content unbiasedly.

### 3.3 Sentence Selection

Our task is essentially to find the optimal subset of sentences that minimize equation (4) with length constraints, which can be seen as a generalization of knapsack problem and is NP-hard as explained in Lin and Bilmes (2011). The simple approximate approach is to select sentences sequentially from the candidate set with a greedy algorithm. Here we introduce two strategies in sentence selection stage to guarantee both efficiency and effectiveness.

**Sentence filtering** This strategy aims to narrow the search space by filtering the irrelevant noisy sentences and reserving the promising sentences as candidate. It also benefits the document modeling process by removing noisy sentences with rare words or in bad format. Unsurprisingly, other existing

summarization systems are suitable for this task. In the experiments, we utilize the baseline methods to rank the sentences first and reserve a subset of top-ranked sentences as candidate. Our method then selects summary sentences from the filtered candidate set.

---

**Algorithm 1** BeamSearch

---

**Require:** Candidate set  $C$ , multi-document set  $D$ , document model  $DM$ , beam size  $k$ , summary length threshold  $\theta$

**Ensure:** A list  $L_k$  including top-k summary sets

```

1:  $L_k, L_{old}, L_{new} \leftarrow \emptyset$ 
2:  $S \leftarrow \emptyset$  and append  $S$  to  $L_{old}$ 
3: while  $L_{old}$  is not empty do
4:   for each sentence  $s$  in  $C$  do
5:     for each summary set  $S$  in  $L_{old}$  do
6:       if  $s \notin S$  then
7:          $S_{new} \leftarrow S \cup s$ 
8:         if  $len(S_{new}^*) < \theta$  then
9:            $\delta \leftarrow \|DM(S_{new}^*) - \frac{1}{n} \sum_{i=1}^n DM(d_i)\|_2^2$ 
10:          if  $S_{new}$  can't further extend then
11:            Update  $L_k$  to reserve the top-k final summary sets with loss  $\delta$ 
12:          else
13:            Update  $L_{new}$  to reserve the top-k promising summary sets with loss  $\delta$ 
14:          end if
15:        end if
16:      end if
17:    end for
18:  end for
19:   $L_{old} \leftarrow L_{new}$ 
20:   $L_{new} \leftarrow \emptyset$ 
21: end while
22: return  $L_k$ 

```

---

**BeamSearch Algorithm** Beamsearch algorithm can be seen as the extension to greedy algorithm, which traverses the entire candidate set  $C$  while limiting itself to  $k$  potential sentences at each selection step. This is similar to the algorithm used for sentence decoding in neural machine translation tasks (Bahdanau et al., 2014; Sutskever et al., 2014) except that the search space in neural machine translation is the vocabulary of words.

As shown in Algorithm 1, a list  $L_k$  is maintained to store top-k final summaries and two lists  $L_{old}, L_{new}$  are used to store the top-k promising summaries at each iteration. The algorithm iterates on the candidate set  $C$  over and over to select sentences until the required summary length is satisfied. At each iteration, all the sentences in the candidate set are added to the current promising summaries in  $L_{old}$  to calculate the reconstruction error. The new summary sets within the length restriction are reserved. The reserved summary sets that have no room for adding new sentences are used to update  $L_k$ , while the rest are used to update  $L_{new}$ . At last, the top summary set in  $L_k$  is chosen and the sentences in it are concatenated sequentially as result.

## 4 Experiments

In this section, we present the experimental results of our model compared with other baseline approaches and analyze the influence of sentence selection strategies.

## 4.1 Experimental Setup

**Data Set** The document understanding conference (DUC)<sup>1</sup> was the main forum providing benchmarks for researchers working on document summarization. We employ DUC 2006 and DUC 2007, the benchmark datasets in multi-document summarization task, for evaluation. DUC 2006 and DUC 2007 contain 50 and 45 document sets respectively. Each document set has 25 news articles for summarization and 4 human-written summaries as ground truth. The length of a result summary is limited to 250 words.

**Evaluation Metric** We use ROUGE toolkit (Lin, 2004) as our evaluation metric, which is adopted by DUC for automatic summarization evaluation. ROUGE measures summary quality by counting overlapping units such as the n-gram, word sequences and word pairs between the candidate summary (produced by algorithms) and the reference summary (produced by humans). Here we report the average F-measures of ROUGE-1, ROUGE-2 and ROUGE-SU4<sup>2</sup>, which are based on uni-gram match, bi-gram match, and unigram plus skip-bigram match with maximum skip distance of 4 between the candidate summary and the reference summary, respectively.

**Document Model Training** DUC datasets are of small scale, making it hard to train a neural network. For BoW model, we simply use the word vectors pre-trained on GoogleNews<sup>3</sup> to infer the document vectors. For PV model, we employ the Thomson Reuters Text Research Collection (TRC2) in Reuters Corpora (Lewis et al., 2004) to train the PV model first, then fine-tune it on the documents in DUC2006 and DUC2007 datasets to learn more precise representation. The entire training set contains 215 million tokens, 1.3 million word types and 1.8 million documents. The python package gensim<sup>4</sup> is used for training PV model and the parameters are set to default.

**Compared Methods** Since we focus on unsupervised extractive summarization task in this work, we compare our model *DocRebuild* with several unsupervised extraction-based algorithms. As the same reason as He et al. (2012), we don't compare with supervised methods (Toutanova et al., 2007; Haghighi and Vanderwende, 2009; Çelikyilmaz and Hakkani-Tür, 2010; Lin and Bilmes, 2011) on DUC2006 and DUC2007 here.

1. **Random** randomly selects sentences from the original document set.
2. **Lead** (Wasson, 1998) sorts the documents chronologically and selects the leading sentences one by one.
3. **DSDR** (He et al., 2012) reconstructs all the sentences in the document set by linearly combining summary sentences and selects sentences to minimize reconstruction error with sparse coding.
4. **SpOpt** (Yao et al., 2015) uses a sparse representation model simultaneously selecting sentences and doing sentence compression, subject to the diversity constraint.

Among these methods, Random and Lead are weaker baselines, DSDR is the original reconstruction method, and SpOpt is a state-of-the-art summarization method which considers sentence compression at the same time. Note that we re-implement the above methods<sup>5</sup> to filter sentences and generate candidate sets for our method. Then we construct our reconstruction model on each candidate set. We use term BoW(\*) to denote the versions with BoW model and term PV(\*) for those with PV model.

## 4.2 Experimental Result

**Overall Performance** Table 1 shows the system comparison results on the two datasets. The parameters of *DocRebuild* are set as follows: the dimensionality of document model is 300 for both BoW model and PV model, the candidate sentences are the top 10% sentences<sup>6</sup>, and the beamsize is set to

<sup>1</sup><http://duc.nist.org>

<sup>2</sup>ROUGE version 1.5.5 with options: -a -n 2 -x -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0 -d -l 250

<sup>3</sup><https://code.google.com/p/word2vec/>

<sup>4</sup><http://radimrehurek.com/gensim/index.html>

<sup>5</sup>Here we used the source code of SpOpt but failed to completely reproduce its results, which may be caused by document preprocessing and parameter setting.

<sup>6</sup>As for Lead, all the leading sentences are reserved as candidate.

Algorithm	DUC2006			DUC 2007		
	Rouge-1	Rouge-2	Rouge-SU4	Rouge-1	Rouge-2	Rouge-SU4
<b>Baselines</b>						
Random	0.34873	0.05447	0.11007	0.36573	0.06346	0.12097
Lead	0.35538	0.06419	0.11672	0.37807	0.07014	0.13111
DSDR	0.36418	0.06100	0.11948	0.38633	0.08052	0.13614
SpOpt	<u>0.40418</u>	<u>0.08388</u>	<u>0.14232</u>	<u>0.41674</u>	<u>0.09905</u>	<u>0.15665</u>
<b><i>DocRebuild</i> (different versions)</b>						
BoW(over Random)	0.37772	0.06584	0.12421	0.39769	0.07974	0.13704
BoW(over Lead)	0.37489	0.07210	0.12665	0.40086	0.08824	0.14070
BoW(over DSDR)	0.38638	0.07239	0.13013	0.41153	0.09236	0.14836
BoW(over SpOpt)	<u>0.40098</u>	<u>0.07953</u>	<u>0.13868</u>	<u>0.42443</u>	<u>0.09768</u>	<u>0.15548</u>
PV(over Random)	0.38117	0.06812	0.12705	0.41113	0.08733	0.14550
PV(over Lead)	0.37912	0.07632	0.13009	0.41423	0.10377	0.15514
PV(over DSDR)	0.40862	0.08485	0.14453	0.42726	0.10308	0.15810
PV(over SpOpt)	<b>0.42193</b>	<b>0.09314</b>	<b>0.15177</b>	<b>0.43426</b>	<b>0.10500</b>	<b>0.16246</b>

Table 1: Average F-measure performance on DUC2006 and DUC2007.

10 (parameters are tuned in the DUC2005). Among all the systems, Random and Lead unsurprisingly show the poorest performance, for they don't consider any semantic information. DSDR performs better by introducing reconstruction framework. SpOpt improves the performance by employing diversity constraint and doing sentence compression. *DocRebuild* obtains a significant<sup>7</sup> improvement on most of the corresponding baselines. Both PV(over DSDR) and PV(over SpOpt) outperform all the baselines and PV(over SpOpt) achieves the best performance. The result demonstrates the rationality of document-level reconstruction and the effectiveness of neural document model.

Besides, among all the versions of *DocRebuild*, PV model performs much better than BoW model, and the gap is more obvious on DUC2006 than on DUC2007. One possible reason is that PV model takes word order into consideration and this advantage is more apparent in the case of long documents (maximal length of documents is 5407 words in DUC2006 while 2663 words in DUC2007). It also can be observed that Rouge-1 score improves more obviously than Rouge-2 and Rouge-SU4 scores. This implicates our document models are more adept at handling words than n-grams since both document models do the prediction task on word-level.

Algorithm	Rouge-1	Rouge-2	Rouge-SU4
None	0.39048	0.07421	0.13383
+Sentence filtering	0.41576	0.08896	0.14847
+Beamsearch	<b>0.42193</b>	<b>0.09314</b>	<b>0.15177</b>

Table 2: Performance with different strategies on DUC2006.

**Analysis on Sentence Selection Strategies** We further discuss the separate influence of two sentence selection strategies with PV(over SpOpt) version on DUC2006. As shown in Table 2, None stands for greedily selecting sentences from all the sentences in the document set, sentence filtering and beamsearch are added sequentially. We can see that sentence filtering has impressive effect on improving our method. This demonstrates that sentence filtering is necessary for making document model work well as document model may be weak in modeling noisy sentences with rare words or in bad format. In addition, beamsearch further improves the performance by considering more possible combination of sentences. The above results indicate these two strategies both work well.

<sup>7</sup>T-test with  $p$ -value  $\leq 0.05$



## 5 Conclusion and Future Work

In this paper, we introduced neural document model into multi-document summarization task and proposed a document-level reconstruction framework named *DocRebuild*. In this framework, we represent and reconstruct the main content of documents with summary sentences on neural document model and take the reconstruction error as objective. To obtain the summary, we use sentence filtering to generate a candidate set and select the summary sentences from the candidate set via beamsearch algorithm. The experiment results show that *DocRebuild* outperforms the state-of-the-art unsupervised algorithms and shows great potential in summarizing multiple documents. In future work, it would be of great interests to extend our model by two ways: (1) trying more complex neural networks to model the documents, and (2) designing new algorithm to improve sentence selection.

## Acknowledgements

This work is partially supported by the National High Technology Research and Development Program of China (Grant No. 2015AA015403) and the National Natural Science Foundation of China (Grant No. 61170091). We would also like to thank the anonymous reviewers for their helpful comments.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.
- Asli Çelikyılmaz and Dilek Hakkani-Tür. 2010. A hybrid hierarchical model for multi-document summarization. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 815–824.
- Günes Erkan and Dragomir R. Radev. 2004. Lexpagerank: Prestige in multi-document text summarization. In *Proceedings of EMNLP*, pages 365–371.
- Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. 1999. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of ACM SIGIR-1999*, pages 121–128.
- Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31 - June 5, 2009, Boulder, Colorado, USA*, pages 362–370.
- Sanda Harabagiu and Finley Lacatusu. 2005. Topic themes for multi-document summarization. In *Proceedings of SIGIR*, pages 202–209.
- Hilda Hardy, Nobuyuki Shimizu, Tomek Strzalkowski, Liu Ting, Xinyang Zhang, and G.Bowden Wise. 2002. Cross-document summarization by concept classification. In *Proceedings of SIGIR-02*, pages 121–128.
- Zhanying He, Chun Chen, Jiajun Bu, Can Wang, Lijun Zhang, Deng Cai, and Xiaofei He. 2012. Document summarization based on data reconstruction. In *Proceedings of AAAI*.
- Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of ICML*, pages 1188–1196.
- David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.
- Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of ACL*, pages 1106–1115.
- Hui Lin and Jeff A. Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of ACL-HLT*, pages 510–520.
- Chin-Yew Lin and Eduard H. Hovy. 2002. From single to multi-document summarization. In *Proceedings of ACL*, pages 457–464.
- Rui Lin, Shujie Liu, Muyun Yang, Mu Li, Ming Zhou, and Sheng Li. 2015. Hierarchical recurrent neural network for document modeling. In *Proceedings of EMNLP*, page 899907.

- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the ACL-04 Workshop*, pages 74–81.
- He Liu, Hongliang Yu, and Zhi-Hong Deng. 2015. Multi-document summarization based on two-level sparse representation model. In *Proceedings of AACL*, pages 196–202.
- Rada Mihalcea and Paul Tarau. 2005. A language independent algorithm for single and multiple document summarization. In *Proceedings of IJCNLP*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.
- Dragomir R. Radev, Hongyan Jing, Magorzata Sty, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Inf. Process. Manage.*, 40(6):919–938.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, page 31043112.
- Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of EMNLP*, page 14221432.
- Kristina Toutanova, Chris Brockett, Michael Gamon, Jagadeesh Jagarlamudi, Hisami Suzuki, and Lucy Vanderwende. 2007. The pythy summarization system: Microsoft research at duc 2007. *Proceedings of DUC-2007*.
- Dingding Wang, Tao Li, Shenghuo Zhu, and Chris Ding. 2008. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *Proceedings of SIGIR*, pages 307–314. ACM.
- Mark Wasson. 1998. Using leading text for news summaries: Evaluation results and implications for commercial summarization applications. In *Proceedings of COLING-ACL*, pages 1364–1368.
- Jinge Yao, Xiaojun Wan, and Jianguo Xiao. 2015. Compressive document summarization via sparse optimization. In *Proceedings of IJCAI*.