Document-based Question Answering Improves Query-focused Multi-Document Summarization

Weikang Li[1], Xingxing Zhang[2], Yunfang Wu[1], Furu Wei[2], and Ming Zhou[2]

[1] Key Laboratory of Computational Linguistics, Peking University, MOE, China
[2] Microsoft Research Asia, Beijing, China
{wavejkd,wuyf}@pku.edu.cn
{xizhang,fuwei,mingzhou}@microsoft.com

Abstract. Due to the lack of large scale datasets, it remains difficult to train neural Query-focused Multi-Document Summarization (QMDS) models. Several large size datasets on the Document-based Question Answering (DQA) have been released and numerous neural network models achieve good performance. These two tasks above are similar in that they all select sentences from a document to answer a given query/question. We therefore propose a novel adaptation method to improve QMDS by using the relatively large datasets from DQA. Specifically, we first design a neural network model to model both tasks. The model, which consists of a sentence encoder, a query filter and a document encoder, can model the sentence salience and query relevance well. Then we train this model on both the QMDS and DQA datasets with several different strategies. Experimental results on three benchmark DUC datasets demonstrate that our approach outperforms a variety of baselines by a wide margin and achieves comparable results with state-of-the-art methods.

Keywords: Document-based Question Answering· Query-focused Multi-Document Summarization · Task Adaptation.

1 Introduction

Automatic document summarization aims to rewrite a document (or documents) into a short piece of text while still retaining the important content from the original. Query-focused Multi-Document Summarization (QMDS) moves one step further, which produces a summary that not only reflects the original documents but also is relevant to the given query. Methods used for QMDS can be grouped into two categories: extractive QMDS and abstractive QMDS. The extractive QMDS copies parts of original documents (usually sentences) as their summaries while its abstractive counterpart can generate new words or phrases, which do not belong to the input documents. Abstractive methods, which is usually based on the sequence to sequence learning, still cannot guarantee the generated summaries are grammatical and conveys the same meaning as the original documents do. Therefore, we focus on the extractive QMDS method.

In the past years, numerous extractive QMDS methods have been developed. Early attempts mainly focus on feature engineering, where features such as sentence length, sentence position, TF-IDF are utilized [13, 11]. Recently, neural network models for extractive summarization attract much attention [3, 2, 17], which are data-driven and are usually needed to be trained on hundreds or thousands of training examples. Unfortunately, the datasets available for training QMDS models are quite small. For example, the numbers of topic clusters are only 50, 50 and 45 in the DUC 2005, 2006 and 2007 datasets, respectively (see details in Section 3.1). The lack of enough training data has become the major obstacle for further improving the performance.

On the other hand, Document-based Question Answering (DQA) datasets have exactly the same format as the QMDS datasets. Given a document, the DQA task (also known as sentence selection) is first to score and then select the high score sentence as the predicted answer of a given question. Especially, there are several large-scale, high-quality datasets for DQA (i.e., SelQA [7]). Moreover, we can easily transform reading comprehension datasets (i.e., SQuAD [16]) to the format of DQA via distant-supervised methods.

With further analysis on the DQA and QMDS datasets, we find that this two kinds of data have similar question length (about 10 words) and document length (about 30 sentences). Considering the similarities of two tasks, we aim to improve the QMDS task with the DQA task.

Specifically, we design a neural network architecture, suitable for both tasks, which mainly consists of a sentence encoder, a query filter and a document encoder. It should be noted that although both tasks share the same network but with different training objectives. Therefore, we propose a novel adaptation method to apply pre-trained DQA models to the QMDS task. We conduct extensive experiments on the benchmark DUC 2005, 2006 and 2007 datasets, and the experimental results demonstrate our approach obtains considerable improvement over a variety of baselines and yields comparable performance with the state-of-the-art results.

Our contributions in this paper can be summarized as follows:

- To the best of our knowledge, we are the first to investigate adapting DQA to the QMDS task.
- We propose a neural network model for both DQA and QMDS tasks and explore a novel adaptation method to improve QMDS with DQA.
- Experimental results validate the efficiency of our proposed approach, which outperforms a variety of baselines.

2 Method

In this section, we first formally define the task, then we introduce the details of our summarization model, and finally, we present the adaptation method to leverage DQA models.

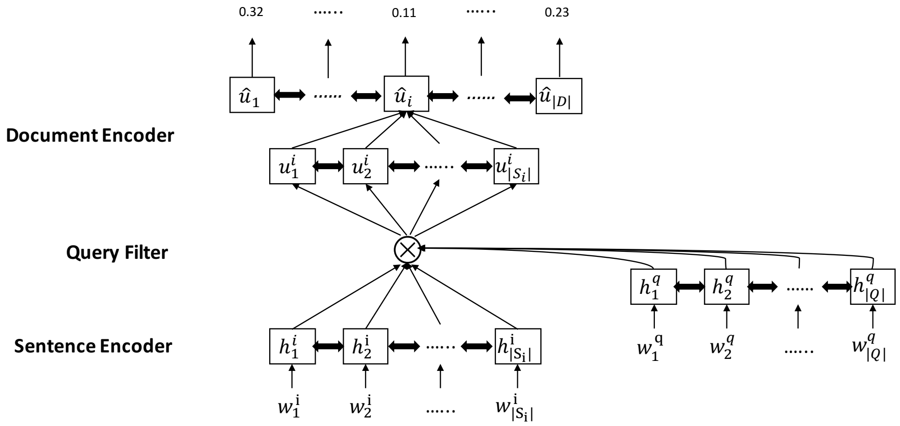

Fig. 1. Framework of our proposed neural network model for QMDS (also suitable for DQA).

2.1 Task Formulation

As mentioned earlier, our model is an extractive model. Given a document cluster (usually contains multiple documents) and a query, our model selects a subset of sentences from the cluster as its summary.

Let $D = (S_1, S_2, \ldots, S_{|D|})$ denote a document in the given document cluster and $S_i = (w_1^i, w_2^i, \ldots, w_{|S_i|}^i)$ a sentence in D. Let $Q = (w_1^q, \ldots, w_{|Q|}^q)$ denote a query for the cluster. Our summarization model is expected to assign a score $\delta(S_i, Q, D)$ for each sentence S_i. Finally a subset of sentences in the document cluster is selected according to $\delta(S_i, Q, D)$ and other constraints as the summary (details of our sentence selection strategies are in Section 2.4).

2.2 Model

Since the extractive model aims to select sentences from the document, it is crucial to model the document well firstly. It is well known that a document is hierarchically structured. Especially, a document is composed of sentences, and each sentence is composed of words. To leverage the hierarchical structure, as shown in Figure 1, we design a hierarchical neural model, where the word level encoders (*Query Filter* and *Sentence Encoder*) aim to learn representations of the query and each sentence in a document and the document level encoder (*Document Encoder*) aims to learn the representation of the document. In the following, we will describe each component of our model in detail.

Sentence Encoder As mentioned earlier, the sentence-level encoder learns to encode a sentence in a document. We opt for a bidirectional Gated Recurrent Unit (BiGRU) [5] network because it is capable of creating context dependent representations for each word. Given the embeddings of a sequence words

$(w_1, w_2, \ldots, w_S)$ (denoted as $(e_1, e_2, \ldots, e_S,))$, a Bi-GRU, which contains two GRUs, processes a sentence in both left-to-right and right-to-left directions and yields two hidden sequences $(h_1^f, h_2^f, \ldots, h_S^f)$ and $(h_1^b, h_2^b, \ldots, h_S^b)$. The final representation of w_j is the concatenation of h_j^f and h_j^b. Now we apply BiGRUs (with different parameters) to a sentence S_i and also to the query Q and obtain $(h_1^i, \ldots, h_{|S_i|}^i)$ and $(h_1^q, \ldots, h_{|Q|}^q)$.

Query Filter In query-focused summarization, as its name implies, the document cluster must be summarized according to the query. Information selection is crucial in this task. We therefore design a *Query Filter* component to inject such information into document/sentence encoding. Specifically, we apply an attention model [12] upon the *Sentence Encoder*. Let $M \in R^{|Q| \times |S_i|}$ denote the attention score matrix between the query Q and a sentence S_i in a document and $M_{m,n}$ an element in M. The computation of $M_{m,n}$ is as follows:

$$M_{m,n} = \frac{\exp(h_m^q \; W \; {h_n^i}^{\mathrm{T}})}{\sum_{k=1}^{|Q|} \exp(h_k^q \; W \; {h_n^i}^{\mathrm{T}})} \tag{1}$$

where h_m^q is the representation of the mth word in query Q and h_n^i is the representation of the nth word in sentence S_i.

Once we have obtained the attention matrix M, we are ready to compute the *new* sentence encoding, which includes the query information. We inject the query information into the representation of each word in S_i using attentions from word representations of Q:

$$v_j^i = \sum_{k=1}^{|Q|} M_{k,j} \cdot h_k^q \tag{2}$$

The final representation of a word w_j^i in S_i is a concatenation of h_j^i and v_j^i as well a couple of binary operations between them:

$$f_j^i = \left[v_j^i; h_j^i; v_j^i \odot h_j^i; v_j^i + h_j^i; v_j^i - h_j^i\right] \tag{3}$$

where ; is the concatenation operation and $\odot$ is element-wise multiplication. Now we have finished the sentence level encoding and we will move to the document encoding in the next section.

Document Encoder The inject of query information has filtered irrelevant information of a sentence, however, it may break the context-dependent information for each word. Besides, a document usually begins with what to talk with and ends with what have talked, which reveals the importance of the sentence order in a document. Thus, we design a hierarchical document encoder, which is composed of a sentence-level encoder and a document-level encoder.

In the sentence level, we again apply one Bi-GRU to encode each word with the input f_j^i and then extract features among words' hidden vectors, $(u_1^i, \ldots, u_{|S_i|}^i)$, to obtain a sentence representation α^i, which is the concatenation of mean and max pooling $[\max_j u_j^i; \frac{1}{|S_i|}\sum_j u_j^i]$. In the document level, we also apply another Bi-GRU to encode each sentence with the input α^i and obtain each sentence's final representation $(\hat{u}_1, \ldots, \hat{u}_{|D|})$. In the end, we apply a feed-forward neural network to compute a salience score p_i for each sentence.

The model is trained to minimize the standard Mean Square Error (MSE) for the QMDS task:

$$L_{sum} = \frac{1}{|D|}\sum_{i \in |D|}(p_i - r(S_i|S_{ref}))^2 \tag{4}$$

where $r(S_i|S_{ref})$ is the ground truth score of S_i in terms of recall ROUGE-2 with respect to human written summaries S_{ref}.

2.3 Adaptation from DQA to QMDS

Since we aim to improve QMDS with DQA, we extend the proposed model to the DQA task by applying a different objective, which is to minimize the cross-entropy between the predicted sentence score p_i and the true sentence score a_i:

$$L_{qa} = -\frac{1}{|D|}\sum_{i \in |D|}\left[a_i \log p_i + (1-a_i)\log(1-p_i)\right] \tag{5}$$

where a_i is the gold label (either 0 or 1) and 1 means the sentence is able to answer the given question and vice versa.

Considering the model's similarities between these two tasks (the only exception is the objective), we therefore apply the pre-trained DQA model to obtain a good starting point for training the QMDS model. Moreover, the pre-trained DQA model, which is good at capturing the query semantic information, could probably improve the query-sentence matching capability when training the QMDS model. We propose a novel adaptation method which includes two aspects. One is to apply the pre-trained DQA model to initialize the QMDS model. Given the initial parameters θ^0, we firstly learn a DQA model θ^{qa} based on the large DQA datasets D^{qa}. The learning process is formulated as follows:

$$Learn(D^{qa};\theta^{qa}) = \arg\min_{\theta^0} L_{qa}(\theta^0) \tag{6}$$

Once we have finished the learning of DQA, we use the θ^{qa} as the initialization parameters of the QMDS model. The other is to utilize the pre-trained DQA model to obtain a query relevance score for each sentence in the document, and then use it as a distant supervised signal. Thus, the loss function for QMDS is changed as follows:

$$L_{sum}^{qa} = \frac{1}{|D|}\sum_{i \in |D|}(p_i - q(S_i|Q))^2 + \frac{1}{|D|}\sum_{i \in |D|}(p_i - r(S_i|S_{ref}))^2 \tag{7}$$

Table 1. Statistics of the DUC datasets.

Dataset	Clusters	Documents	Data Source
DUC 2005	50	1593	TREC
DUC 2006	50	1250	AQUAINT
DUC 2007	45	1125	AQUAINT

Table 2. Statistics of the DQA datasets. With the assumption that the sentence containing the answer span is correct, we convert the span-based SQuAD dataest to the sentence-based SQuAD† dataset.

Dataset	Split	#Documents	#Sentences
SQuAD†	TRAIN	87341	440573
	DEV	5273	26442
SelQA	TRAIN	5529	66438
	DEV	785	9377

where $q(S_i|Q)$ is the query Q relevance score of a sentence S_i predicted by the pre-trained DQA model. Finally, the learning of the QMDS model is formulated as follows:

$$Learn(D^{sum}; \theta^{sum}) = \arg\min_{\theta^{qa}} L^{qa}_{sum}(\theta^{qa}) \tag{8}$$

2.4 Sentence Selection

The summarization model we have described can estimate the importance of all sentences in the input documents. This section focus on creating a summary with the output of our summarization model.

Once we have assigned each sentence a score via a trained summarization model, we are ready to select a subset of sentences as the final summary. The method we used in our paper is similar to the proposed methods [2, 17]. We employ a simple greedy algorithm, similar to the MMR strategy [4]. The algorithm starts with the sentence of the predicted highest score. In each step, a new sentence S_i is added to the summary if it satisfies the following two conditions:

1. It has the highest score in the remaining sentences.
2. It contains significantly new bi-grams compared with the current summary content. We empirically set the cut-off of the new bi-gram ratio to 0.35.

3 Experiments

We describe the experimental setting and report empirical results in this section.

3.1 Datasets

In this paper, we focus on improving the performance on QMDS task with the help of the DQA task. The experiments are conducted on the public Document Understanding Conference (DUC) 2005, 2006 and 2007 datasets. All the documents in the dataset are from news websites and grouped into various thematic clusters. The DUC 2005, DUC 2006 and DUC 2007 datasets consist of 50, 50 and 45 topics respectively, and each topic includes a document set of 25 $\sim$ 50 news articles and a short description of a topical query. The task is to create a summary containing no more than 250 words for each document set to answer the query. There are at least four human written reference summaries provided in each document collection for evaluation. The datasets are briefly described in Table 1. We follow standard practice and train our QMDS models on two years of data and test on the third. It should be noted that 10 clusters are split from the training set to form the dev set.

Two different datasets are used for the DQA task: SelQA [7] and SQuAD [16]. Both datasets contain open-domain questions whose answers are extracted from Wikipedia articles. SelQA is a sentence-based DQA dataset, in which there is at least one correct sentence in the document for a question. The SQuAD is a span-based DQA dataset, and we could derive datasets for answer sentence selection from the original dataset. We assume that the sentences containing correct answer spans are correct, and vice versa. We merge them when training a DQA model. Table 2 shows the statistics of the two datasets above.

3.2 Evaluation Metrics

We employ the widely-adopted automatic evaluation metric ROUGE [3] for evaluation. We reported recall based ROUGE-1 and ROUGE-2 limited to 250 words. It automatically measures the quality of a summary by counting the number of overlapping units such as the n-gram, word sequences and word pairs between the candidate summary and reference summaries created by humans. ROUGE-2 recall is used as the main metric for comparison because it correlates well with human judgments.

3.3 Implementation Details

The proposed model is implemented with TensorFlow. The dimension of word embeddings is set to 300. The word embeddings are initialized with *300D GloVe* vectors [15], and out-of-vocabulary words in the training set are initialized randomly. We fix the embeddings during training. We train the model with Adam optimization algorithm [8] with a learning rate of 0.001, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. Our models are trained with a batch size of 5. We set the hidden unit size $d = 80$ for both sentence-level and document-level GRUs and all GRUs have one layer.

[3] ROUGE-1.5.5 with options: -n 2 -l 250 -m -u -c 95 -x -r 1000 -f A -p 0.5 -t 0

Table 3. Results of different adaptation methods to the DQA on the DUC three datasets using the length of 250 words recall ROUGE-1 (R-1) and ROUGE-2 (R-2).

(a) on the DUC 2005.

Model	**R-1**	Δ/%	**R-2**	Δ/%
QA-None	33.96	–	6.04	–
QA-Loss	35.78	$1.82^{\uparrow}$	6.47	$0.43^{\uparrow}$
QA-Init	36.45	$2.49^{\uparrow}$	6.92	$0.88^{\uparrow}$
QA-Init&Loss	37.25	$3.29^{\uparrow}$	7.13	$1.09^{\uparrow}$

(b) on the DUC 2006.

Model	**R-1**	Δ/%	**R-2**	Δ/%
QA-None	36.05	–	6.64	–
QA-Loss	37.40	$1.35^{\uparrow}$	7.46	$0.82^{\uparrow}$
QA-Init	38.74	$2.69^{\uparrow}$	8.44	$1.80^{\uparrow}$
QA-Init&Loss	39.41	$3.36^{\uparrow}$	9.05	$2.41^{\uparrow}$

(c) on the DUC 2007.

Model	**R-1**	Δ/%	**R-2**	Δ/%
QA-None	37.54	–	8.51	–
QA-Loss	39.22	$1.68^{\uparrow}$	8.85	$0.34^{\uparrow}$
QA-Init	40.44	$2.90^{\uparrow}$	9.69	$1.18^{\uparrow}$
QA-Init&Loss	40.55	$3.01^{\uparrow}$	10.20	$1.69^{\uparrow}$

3.4 Comparison Systems

To evaluate the overall performance of the proposed method, we compare it with a variety of baseline methods, including some traditional baselines, and several recent extractive query-focused summarization systems, which are typically based on different neural network structures. We dont implement compared models and directly take the reported performance in the original papers. Compared methods includes *LEAD* [19], *QUERY-SIM* [2], *SVR* [14], *MultiMR* [18], *DocEmb* [9], *ISOLATION* [2], *AttSum* [2], *CRSum* [17].

3.5 Results

Effectiveness of DQA We firstly conduct experiments to verify the effectiveness of pre-trained DQA models. The proposed adaptation method in our paper can be divided into three ways to make use of the pre-trained DQA model. The first one is to pre-train our model on the DQA datasets and then continue to train the model on the QMDS datasets, denoted as *QA-Init* (see Equation 6). The second one is first to produce a query relevance score for each sentence in the QMDS datasets with a well-trained DQA model and then joint train the model with the supervised query relevance signal and sentence salience signal, denoted as *QA-Loss* (see Equation 7). The last one is to combine both *QA-Init* and *QA-Loss*, denoted as *QA-Init&Loss* (see Equation 8). We denote the model trained only with the QMDS datasets as *QA-None*. Table 3 shows the performances of different adaptation methods to use the pre-trained DQA model on three benchmarks.

As shown in Table 3, we can see that the three adaptation methods using DQA are quite effective to improve the QMDS task. Specifically, *QA-Loss* outperforms *QA-None* by 0.43, 0.82 and 0.34 in terms of ROUGE-2 and 1.82, 1.35

Table 4. Experimental results of the QMDS task on three benchmark datasets using the length of 250 words recall ROUGE-1 (R-1) and ROUGE-2 (R-2).

Methods	DUC 2005		DUC 2006		DUC 2007	
	R-1	R-2	R-1	R-2	R-1	R-2
LEAD [2]	29.71	4.69	32.61	5.71	36.14	8.12
QUERY-SIM [2]	32.95	5.91	35.52	7.10	36.32	7.94
SVR [2]	36.91	7.04	39.24	8.87	43.42	11.10
MultiMR [2]	35.58	6.81	38.57	7.75	41.59	9.34
DocEmb [2]	30.59	4.69	32.77	5.61	33.88	6.46
ISOLATION [2]	35.72	6.79	40.58	8.96	42.76	10.79
AttSum [2]	37.01	6.99	**40.90**	**9.40**	**43.92**	**11.55**
CRSum [17]	36.96	7.01	39.51	9.19	41.20	11.17
Our model	**37.25**	**7.13**	39.41	9.05	40.55	10.20

and 1.68 in terms of ROUGE-1 on the DUC 2005, 2006 and 2007 datasets, respectively. *QA-Init* achieves a performance gain of 0.88, 1.80 and 1.18 ROUGE-2 points and 2.49, 2.69 and 2.90 ROUGE-1 points over *QA-None*. And *QA-Init&Loss* yields 1.09, 2.41 and 1.69 ROUGE-2 improvements and 3.29, 3.36 and 3.01 improvements. As can be seen, the improvements on the DUC 2005 dataset is smaller than that on the DUC 2006 and 2007 datasets, which may be because of the differences in numbers of documents under a topic cluster. In the DUC 2005 dataset, a topic cluster contains 32 documents on average, while in the other two datasets the number is 25 documents on average. It becomes hard when the number of candidate sentences increases. Among the three adaptations, *QA-Init&Loss* achieves the best performance than the others and *QA-Init* is better than *QA-Loss*.

In the following, we compare our best model *QA-Init&Loss* against several recent models.

Performance Comparison For the compared approaches, we list the best results reported in the original literature. The overall performance comparisons on the DUC 2005, DUC 2006 and DUC 2007 datasets are shown in Table 4. Our proposed method obtains the state-of-the-art performance on the DUC 2005 dataset and achieves comparable results on the DUC 2006 and 2007 datasets.

The first block in Table 4 represents non-neural network methods, which apply manual features or well-defined rules. Recent neural network methods are shown in the second block of Table 4. On the DUC 2005 dataset, our model outperforms the previous best method *AttSum* by 0.24 ROUGE-1 points and exceeds the previous best method *SVR* by 0.09 ROUGE-2 points. On the DUC 2006 dataset, our model outperforms the feature-based methods in terms of ROUGE-1 and ROUGE-2 and achieves comparable performances with neural network-based methods. On the DUC 2007 dataset, our model is on par with

Table 5. Statistics of the query with question type in the DUC datasets.

Dataset	Clusters	Question Type	Proportion
DUC 2005	50	30	60.0%
DUC 2006	50	24	48.0%
DUC 2007	45	11	24.4%

the public methods except feature-based methods *SVR* and neural network-based methods *AttSum* and *CRSum*. It is noted that *SVR* heavily depends on hand-crafted features, while our model does not use any manual features. Our proposed neural network model is designed to be suitable for both QMDS and DQA tasks, which is different from the QMDS-specific models (e.g., *AttSum* and *CRSum*). Moreover, *CRSum* also extract word-level features via convolutional neural networks. There are two kinds of query type in the DUC (2005-2007) datasets, namely description type query and question type query. As shown in Table 5, we found that the question type queries has a high proportion of clusters on the DUC 2005 dataset (60.0%) and low proportion on the other datasets (only 24.4% on the DUC 2007 dataset), which may explain the a little bit worse performance of our proposed method on the DUC 2007 dataset. QA based initialization and training objective tends to improve question type queries.

4 Related Work

As a challenging issue for text understanding, automatic document summarization has been studied for a long period . Except for computing sentence salience, QMDS also needs to concern the query-sentence relevance, which makes it harder than the Generic MDS. Cao et al. [2] propose a neural network model (AttSum) which jointly handles sentence salience ranking and query relevance ranking. It automatically generates distributed representations for sentences as well as the document cluster. Meanwhile, it applies an attention mechanism that tries to simulate human attentive reading behavior when a query is given. Ren et al. [17] find that sentence relations in the document play an essential role, and so propose a Contextual Relation-based Summarization model (CRSum), which firstly uses sentence relations with a word-level attentive pooling convolutional network to construct sentence representations and then use contextual relations with a sentence-level attentive pooling recurrent neural network to construct context representations. Finally, CRSum automatically learns useful contextual features by jointly learning representations of sentences and similarity scores between a sentence and its contexts. Inspired by these two works, we design a hierarchical neural network model, which is not only able to capture sentence relations via a Bi-GRU structure, but also pays attention to the query with the attention mechanism. It should be noted that our proposed model is end-to-end and does not require manual features. So, the proposed model with manual features (CRSum+SF+QF) in the paper [17] is not referred to in our experiment. Meanwhile, our proposed model is also suitable for DQA, which benefits the adaptation from the DQA task to the QMDS task.

Speaking of domain adaptation, works from SDS to MDS is emerging in recent years due to the insufficient labeled data in MDS. Lebanoff et al. [10] describe a novel adaptation method (PG-MMR), which combines an extractive summarization algorithm (MMR) for sentence extraction and an abstractive model (PG) to fuse source sentences. Zhang et al. [20] add a document set encoder to their hierarchical summarization framework and propose three strategies to improve the model performance further. Baumel et al. [1] try to apply the 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 every single document until the length limit is reached. Different from them, we explore how to apply DQA to improve QMDS. To the best of our knowledge, we are the first to do like this.

5 Conclusion

We propose a novel adaptation of applying DQA to improve the model's performance on the QMDS task. Our proposed network is designed to fit both tasks, which includes a sentence encoder, a query filter and a document encoder. Extensive experiments demonstrate that our proposed method can indeed improve over a variety of baselines and yields comparable results with state-of-the-art methods. The method we have proposed is one of the many possible methods for utilizing DQA datasets (and models). In the future, we plan to explore other adaptation methods, like meta-learning [6] and investigating more tasks related to the QMDS task.

Acknowledgment This work is supported by the National Natural Science Foundation of China (61773026, 61572245).

References

1. Baumel, T., Eyal, M., Elhadad, M.: Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. arXiv preprint arXiv:1801.07704 (2018)
2. Cao, Z., Li, W., Li, S., Wei, F.: Attsum: Joint learning of focusing and summarization with neural attention. CoRR **abs/1604.00125** (2016), http://arxiv.org/abs/1604.00125
3. Cao, Z., Wei, F., Li, S., Li, W., Zhou, M., Houfeng, W.: Learning summary prior representation for extractive summarization. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). vol. 2, pp. 829–833 (2015)

4. Carbonell, J., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. pp. 335–336. ACM (1998)
5. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
6. Gu, J., Wang, Y., Chen, Y., Cho, K., Li, V.O.: Meta-learning for low-resource neural machine translation. arXiv preprint arXiv:1808.08437 (2018)
7. Jurczyk, T., Zhai, M., Choi, J.D.: Selqa: A new benchmark for selection-based question answering. In: Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on. pp. 820–827. IEEE (2016)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Kobayashi, H., Noguchi, M., Yatsuka, T.: Summarization based on embedding distributions. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1984–1989 (2015)
10. Lebanoff, L., Song, K., Liu, F.: Adapting the neural encoder-decoder framework from single to multi-document summarization. arXiv preprint arXiv:1808.06218 (2018)
11. Li, C., Qian, X., Liu, Y.: Using supervised bigram-based ilp for extractive summarization. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 1004–1013 (2013)
12. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
13. Ouyang, Y., Li, S., Li, W.: Developing learning strategies for topic-based summarization. In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. pp. 79–86. ACM (2007)
14. Ouyang, Y., Li, W., Li, S., Lu, Q.: Applying regression models to query-focused multi-document summarization. Information Processing & Management **47**(2), 227–237 (2011)
15. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
16. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250 (2016)
17. Ren, P., Chen, Z., Ren, Z., Wei, F., Ma, J., de Rijke, M.: Leveraging contextual sentence relations for extractive summarization using a neural attention model. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 95–104. SIGIR '17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/3077136.3080792, http://doi.acm.org/10.1145/3077136.3080792
18. Wan, X., Xiao, J.: Graph-based multi-modality learning for topic-focused multi-document summarization. In: IJCAI. pp. 1586–1591 (2009)
19. Wasson, M.: Using leading text for news summaries: Evaluation results and implications for commercial summarization applications. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2. pp. 1364–1368. Association for Computational Linguistics (1998)
20. Zhang, J., Tan, J., Wan, X.: Towards a neural network approach to abstractive multi-document summarization. arXiv preprint arXiv:1804.09010 (2018)