Sliding Selector Network with Dynamic Memory for Extractive Summarization of Long Documents

Peng Cui and **Le Hu**
School of Computer Science and Technology
Harbin Institute of Technology, Harbin, China
{pcui,lhu}@insun.hit.edu.cn

Abstract

Neural-based summarization models suffer from the length limitation of text encoder. Long documents have to been truncated before they are sent to the model, which results in huge loss of summary-relevant contents. To address this issue, we propose the sliding selector network with dynamic memory for extractive summarization of long-form documents, which employs a sliding window to extract summary sentences segment by segment. Moreover, we adopt memory mechanism to preserve and update the history information dynamically, allowing the semantic flow across different windows. Experimental results on two large-scale datasets that consist of scientific papers demonstrate that our model substantially outperforms previous state-of-the-art models. Besides, we perform qualitative and quantitative investigations on how our model works and where the performance gain comes from.[1]

1 Introduction

Text summarization is an important task of natural language processing which aims to distil salient contents from a textual document. Existing summarization models can be roughly classified into two categories, which are abstractive and extractive. Abstractive summarization usually adopts natural language generation technology to produce a word-by-word summary. In general, these approaches are flexible but may yield disfluent summaries (Liu and Lapata, 2019a). By comparison, extractive approaches aim to select a subset of the sentences in the source document, thereby enjoying better fluency and efficiency (Cao et al., 2017).

Although many summarization approaches have demonstrated their success on relatively short documents, such as news articles, they usually fail to achieve desired performance when directly applied in long-form documents, such as scientific papers. This inferior performance is partly due to the truncation operation, which inevitably leads to information loss, especially for extractive models because parts of gold sentences would be inaccessible. In addition, the accurate modeling of long texts remains a challenge (Frermann and Klementiev, 2019).

Paragraph 1: Medical tourism is illustrated as occurrence in which individuals travel abroad to receive healthcare services. It is a multi-billion dollar industry and countries like India, Thailand, Israel, Singapore, …
Paragraph 2: The prime driving factors in medical tourism are increased medical costs, increased insurance premiums, increasing number of uninsured or partially insured individuals in developed countries, …
……
Paragraph 5: It is generally presumed in marketing that products with similar characteristics will be equally preferred by the consumers, however, attributes, which make the product similar to other products, will not….

Figure 1: An example where a paragraph-by-paragraph extraction will produce an incoherent summary.

A practical solution for this problem is to use a sliding window to process documents separately. This approach is used in other NLP tasks, such as machine reading comprehension (Wang et al., 2019b). However, such a paradigm is not suitable for summarization task because the concatenation of summaries that are independently extracted from local contexts is usually inconsistent with the gold summary of the entire document. Figure 1 shows an example to illustrate this problem. The core topic of the source document is "*medical tourism*," which is discussed in Paragraphs 1 and 2. How-

[1]Code will be released at https://github.com/pcui-nlp/SSN_DM

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5881–5891
June 6–11, 2021.

ever, the 5-th paragraph is mainly about "*consumer and product*." As a consequence, the paragraph-by-paragraph extraction approach might produce a both repetitive and noisy summary. Under this circumstance, the supervised signals will have a negative effect on model behaviors because understanding why Paragraph 5 should output an empty result without information conveying from previous texts is confused for the model.

In this paper, we propose a novel extractive summarization model for long-form documents. We split the input document into multiple windows and encode them with a sliding encoder sequentially. During this process, we introduce a memory to preserve salient information learned from previous windows, which is used to complete and enrich local texts. Intuitively, our model has the following advantages: 1) In each window, the text encoder processes a relatively short segment, thereby yielding more accurate representations. 2) The local text representations can capture beyond-window contextual information via the memory module. 3) The previous selection results are also parameterized in the memory block, allowing the collaboration among summary sentences.

To sum up, our contributions are threefold.

(1) We propose a novel extractive summarization model that can summarize documents of arbitrary length without truncation loss. Also, it employs the memory mechanism to address context fragmentation. To the best of our knowledge, we are the first to propose applying memory networks into extractive text summarization task.

(2) The proposed framework (i.e., a sliding encoder combined with dynamic memory) provides a general solution for summarizing long documents and can be easily extended to other abstractive and extractive summarization models.

(3) Our model achieves the state-of-the-art results on two widely used datasets for long document summarization. Moreover, we conduct extensive analysis to understand how our model works and where the performance gain comes from.

2 Related Work

Neural Extractive Summarization. Neural-networks have become the dominant approach for extractive summarization. Existing studies usually formulate this task as sentence labelling (Dong et al., 2018; Nallapati et al., 2016; Zhang et al., 2019) or sentence ranking (Narayan et al., 2018). Among them, recurrent neural networks (Cheng and Lapata, 2016; Zhou et al., 2018), Transformer encoder (Wang et al., 2019a), or graph neural networks (Wang and Liu, 2020, Xu et al., 2020, Cui et al., 2020) (Wang et al., 2020; Xu et al., 2019; Cui et al., 2020) have been used to learn sentence representation.

Recently, pre-trained language model (e.g. BERT (Devlin et al., 2018)) has provided substantial performance gain for extractive summarization. Liu and Lapata (2019b) modified standard BERT for document modelling. Xu et al. (2019) used a span-BERT to perform span-level summarization. Zhong et al. (2020) regarded document summarization as a semantic matching task and used a Siamese-BERT as the matching model. However, the valid length of standard BERT is only 512, which means most of them can hardly generalize to long-form documents effectively.

Long Document Summarization. Recent years have seen a surge of interest on long document summarization, especially scientific publications. Celikyilmaz et al. (2018) used a multi-agent framework to boost the encoder performance. Cohan et al. (2018) proposed a hierarchical network that incorporates the discourse structures into the encoder and decoder. Xiao and Carenini (2019) proposed to model the local and global contexts jointly. Cui et al. (2020) proposed a hybrid model that employs a neural topic model (NTM) to infer latent topics as a kind of global information.

Despite their success, these approaches still face the input length limitation and the difficulty in encoding long texts accurately. In comparison, our model addresses these problems with a novel segment-wise extraction way and can summarize arbitrarily long documents without any content truncation.

Memory Networks. Memory network (Weston et al., 2015) is a general framework that employs a memory bank to model long-term information. Due to its flexible architecture and superior adaptability, it has been applied into various NLP scenarios, such as text classification (Zeng et al., 2018), question answering (Kumar et al., 2016; Xiong et al., 2016), and sentiment analysis (Tang et al., 2016). In this study, we leverage a memory module capture beyond-window when performing segment-level summarization. To the best of our knowledge, memory networks have never been applied into extractive summarization task.

3 Model

This section describes our model, namely, the *Sliding Selector Network with Dynamic Memory* (SSN-DM), of which Figure 2 gives an overall architecture. Formally, given a document D of arbitrary length, we first split D into multiple segments according to the pre-defined window length. Then, we use a BERT encoder to sequentially encode each segment and select salient sentences. During this process, a memory module is applied to achieve the information flow across different windows. Finally, the extracted sentences are aggregated to generate the final summary. We elucidate each module in the following subsections.

3.1 Sliding Encoder

Let $seg^k = s_1^k, s_2^k, \ldots, s_n^k$ be the kth window consisting of n sentences. We encode the window text with a pre-trained BERT, which has been proven effective on extractive summarization task (Liu and Lapata, 2019b; Xu et al., 2019; Cui et al., 2020). Following previous studies, we modify the standard BERT by inserting $[CLS]$ and $[SEP]$ tokens into the beginning and end of each sentence, respectively.

$$O_B = BERT(w_{1,CLS}^k, w_{1,2}^k, \ldots, w_{n,SEP}^k) \quad (1)$$

where $w_{i,j}^k$ denotes the jth word of the ith sentence. $O_B = \{h_{1,CLS}^k, h_{1,2}^k, \ldots, h_{n,SEP}^k\}$ denotes the representations of each token learned by BERT. We regard the hidden states of $[CLS]$ tokens $H^k = \{h_{1,CLS}^k, h_{2,CLS}^k, \ldots, h_{n,CLS}^k\}$ as the corresponding sentence representations.

On top of BERT encoder, we add an additional layer to incorporate two types of structural information. The first part is the position information of the current window. In our segment-wise encoding, the position embeddings equipped in BERT are recalculated in each window, thereby losing the exact position of each token in the entire document. This positional bias may lead to inferior performance (Zhong et al., 2019; Dai et al., 2019). To address this problem, we assign a window-level position encoding to each window as a complementary feature, indicating its relative position in the document.

In addition, we further introduce a group of section (e.g., introduction, conclusion) embeddings to capture the discourse information, which has been proved an important feature for scientific papers summarization (Cohan et al., 2018). Combining these two parts, the structural encoding layer can be denoted as:

$$f_s(H^k) = tanh(W_1 H^k + W_2 e_w^k + W_3 e_s) \quad (2)$$

where e_w^k indicates the kth window-level position embedding, and e_s the section embedding. Both of them are randomly initialized and learned as a part of the model. Throughout the paper, W_* represents trainable parameter matrix.

Noticeably, the section features might not be generally available for long texts of other genres. Therefore, in our experiments, we consider e_s as an optional setting and conduct quantitative investigations on Section 5 to probe into its effect on model performance.

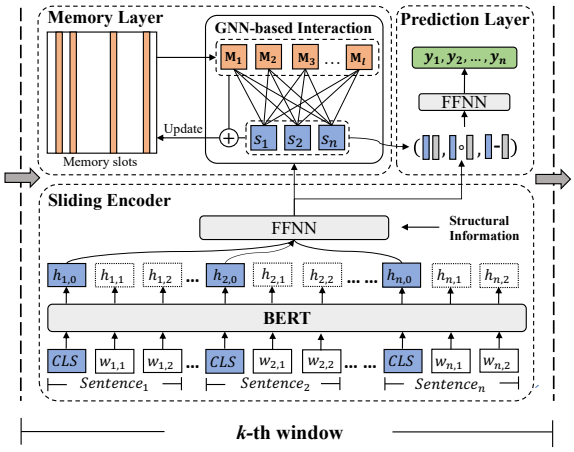

Figure 2: The framework of our model. There are three major components: (1) The sliding encoder generates representation of each sentence in the current window. (2) The memory layer infuses history information into sentence representations via graph neural networks. (3) The predication layer aggregates learned features to compute the binary sentence labels.

3.2 Graph-based Memory Interaction

After encoding the window text, we infuse the history information of previous texts into the learned representations H^k via a memory module. Let $M^k \in \mathbb{R}^{l \times d_m}$ be the memory block in the kth window that preserves salient information of previous $k-1$ windows, where l represents the number of memory slots and d_m represents the dimension of memory vector. M^0 is initialized with fixed values in the first window and then updated in the learning process dynamically. The detail of this part is explained in Section 3.4.

We use a graph neural network to model the interaction between memory module and the current window. Concretely, we first construct a bipartite

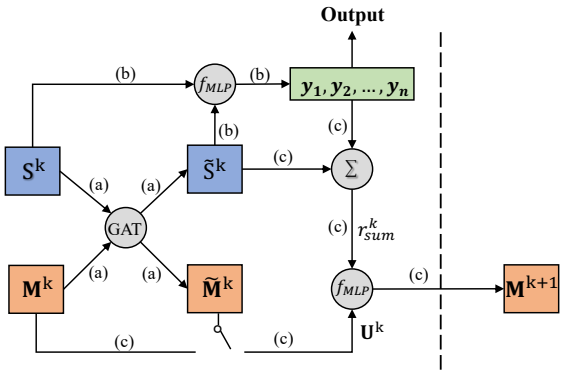

Figure 3: An illustration of the information flow in our model. Paths (a) denote the interaction between memory vectors (**M**) and sentence representations (**S**) via a GAT layer. Paths (b) denote the compution of sentence labels. Paths (c) denote the updating process of memory module.

graph that consists of l memory nodes and n sentence nodes, whose embeddings are initialized with M^k and H^k, respectively. Then, we use graph attention network (GAT; Velickovic et al., 2018) to encode this graph. Given a sentence node h_i, we update its representation by aggregating its neighboring nodes, as shown as follows,

$$\begin{aligned} z^k_{i,j} &= LeaklyRelu(W_a[h^k_i; SG(m^k_j)]), \\ \alpha_{i,j} &= \frac{exp(z^k_{i,j})}{\sum_{j=1}^{l} exp(z^k_{i,j})}, \\ \tilde{h}^k_i &= \|_{t=1}^{T} \sum_{j=1}^{l} tanh(\alpha^t_{i,j} W^t_c SG(m^k_j)), \end{aligned} \tag{3}$$

where $\alpha_{i,j}$ denotes the attention weight from node h^k_i to node m^k_j. Multi-head attention is applied to stabilize the calculation process. Function $SG(\cdot)$ stands for stop-gradient operation.

We refer $\tilde{H}^k$ and $\tilde{M}^k$ to the sentence representations and memory vectors after graph propagation, respectively. During the graph interaction, the sentence representations are completed and enriched by history information and vice versa.

Empirical observations of prior research (Tang et al., 2016; Zeng et al., 2018) have shown that stacking multiple memory layers can bring further performance gain. Similarly, in our model, the multi-hops setting can be achieved by increasing the graph iteration number, i.e., repeating the GAT calculation process (Eq. 3).

3.3 Prediction Layer

We have obtained the sentence representations H^k derived from window text, and its extended version $\tilde{H}^k$ enriched by memory information. Given ith sentence, we send h^k_i and $\tilde{h}^k_i$ into a MLP classifier to compute its summary label.

$$\tilde{y}_i = f_o(\tilde{h}^k_i, h^k_i, |h^k_i - h^k_i|, \tilde{h}^k_i \circ h^k_i) \tag{4}$$

where $\tilde{y}_i$ represents the predicted probability of ith sentence, and $\circ$ represents the point-wise operation. f_o is a feed-forward network with three hidden layers. We construct interaction features between $\tilde{h}^k_i$ and h^k_i to capture the importance of ith sentence in both current segment and history context.

The training objective of the model is to minimize the binary cross-entropy loss given the predictions and ground truth sentence labels, i.e., $\mathcal{L} = -\sum y_i log(\tilde{y}_i) + (1 - y_i) log(1 - \tilde{y}_i)$

After processing the entire document, we rank all the sentences and select top-k as the final summary, where k is a hyperparameter set according to the average length of reference summaries. It worth noting that the memory module also acts as an intermediary to make the sentence scores of different windows comparable.

3.4 Dynamic Memory Updating

Now we explain the learning process of memory module. Figure 3 presents the information flow of our model. In each window, after the prediction layer, we update the memory values with two inputs.

First, recall that in GAT calculation, the updated memory vectors $\tilde{M}^k$ has also encoded the contextual information of the current window during the interaction with H^k. Therefore, we combine $\tilde{M}^k$ and M^k with gating mechanism (Chung et al., 2014).

$$\begin{aligned} \sigma^k_i &= tanh(W_m * \tilde{m}^k_i), \\ u^k_i &= \sigma^k_i \circ m^k_i + (1 - \sigma^k_i) \circ \tilde{m}^k_i \end{aligned} \tag{5}$$

where u^k_i is the liner interpolation between history memory m^k_i and the newly computed mi^k. $\sigma^k_i \in \mathbb{R}^{d_m}$ is an gate vector to modulates the information flow.

The second part refers to the extraction result of the current window. We first aggregate the sentence representations with their predicted probabilities

(Eq.4) to parameterize the selected sentences.

$$r^k_{sum} = \sum_{i=1}^{n} \tilde{y}_i * h^k_i. \quad (6)$$

Here, r^k_{sum} can be considered a sentence-level coverage vector (See et al., 2017) that records what contents has been extracted from the current window. This ensures that the following selection is informed by previous decisions.

Then, we use a single feedforward layer to generate new memory $M^{k+1} = \{m^{k+1}_1, \ldots, m^{k+1}_l\}$ for next window.

$$m^{k+1}_i = tanh(W_4 m^k_i + W_5 r^k_{sum}). \quad (7)$$

4 Experimental Setup

4.1 Datasets

Our model is particularly designed for long document summarization. For this reason, we do not conduct experiments on the widely explored news datasets consisting of relatively short documents. For example, the articles in DailyMail (Hermann et al., 2015) dataset have an average of 600 words, which can be effectively processed by most existing models. Instead, following prior research on long-form documents summarization(Cohan et al., 2018; Xiao and Carenini, 2019; Cui et al., 2020; Zhong et al., 2020), we evaluate our model on the following two large-scale scientific paper datasets.

Datasets	#Doc			Avg. Tokens	
	Train	Val.	Test	Doc	Sum
arXiv	203,037	6,436	6,440	4,938	220
PubMed	119,224	6,633	6,658	3,016	203

Table 1: The statistics of two datasets

arXiv and **PubMed** (Cohan et al., 2018) are two recently constructed datasets collected from arXiv.com and PubMed.com, respectively. Both of them consist of scientific papers, which are much longer than the common news articles. We preprocess and split datasets in accordance with (Cohan et al., 2018) and use the oracle labels created by (Xiao and Carenini, 2019). Their statistics is summarized in Table 1.

Figure 4 shows the position distributions of ground-truth sentences of the two datasets, where we can see the importance of the long text processing ability for extractive summarization models. For example, the maximum length of standard BERT is 512, which means that a large proportion

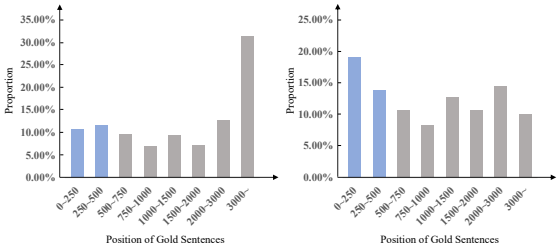

Figure 4: Position distribution of gold sentences on two datasets.

(colored in grey) of ground-truth sentences would be inaccessible for existing state-of-the-art BERT-based summarization models.

4.2 Models for Comparison

We compare our model with the following state-of-the-art summarization approaches.

Pointer Generator Network (PGN; See et al., 2017) extends the standard seq2seq framework with attention, coverage, and copy mechanism.

Discourse-Aware (Cohan et al., 2018) is an abstractive model particularly designed for summarizing long-form document with discourse structure. It employs a hierarchical encoder and explicitly introduces the section information of scientific papers.

Seq2seq-local&global (Xiao and Carenini, 2019) is also an extractive model for long document summarization that jointly encodes local and global contexts.

Match-Sum (Zhong et al., 2020) is a state-of-the-art BERT-based summarization model. It performs summary-level extraction based on the matching scores between candidate summary and the source document.

Topic-GraphSum (Cui et al., 2020) introduces a joint neural topic model to explore latent topics as a kind of global information to help summarize long documents. Since Cui et al. (2020) used different data preprocessing, we repeat the experiments using the model released by the authors and preprocess the data in accordance with previous studies (Cohan et al., 2018; Xiao and Carenini, 2019) to make the results comparable.

4.3 Implementation Details

For the sliding encoder, we use the "bert-base-uncased" version with the hidden size of 768 and fine-tune it for all experiments. The maximum length of window is set to 512, and we segment the

Models	arXiv			PubMed		
	R-1	R-2	R-L	R-1	R-2	R-L
Lead	33.66	8.94	22.19	35.63	12.28	25.17
LexRank+	33.85	10.73	28.99	39.19	13.89	34.59
LSA+	29.91	7.42	25.67	33.89	9.93	29.70
Oracle*	53.88	23.05	34.90	55.05	27.48	38.66
Seq2seq-attention+	29.30	6.00	25.56	31.55	8.52	27.38
PGN+	32.06	9.04	25.16	35.86	10.22	29.69
Disourse-aware+	35.80	11.05	31.80	38.93	15.37	35.21
Cheng & Lapta (2016)*	42.24	15.97	27.88	43.89	18.53	30.17
SummaRuNNer*	42.81	16.52	28.23	43.89	18.78	30.36
Seq2seq-local&global*	43.62	17.36	29.14	44.85	19.70	31.43
Match-Sum	40.59	12.98	32.64	41.21	14.91	**36.75**
Topic-GraphSum	44.03	18.52	32.41	45.95	20.81	33.97
SSN-DM	**45.03**	19.03	32.58	**46.73**	**21.00**	34.10
SSN-DM + discourse	44.90	**19.06**	**32.77**	46.52	20.94	35.20

Table 2: Rouge results on two dataets. Apart from the baselines mentioned in Section 4.2, we also collected the public results reported by previous studies. Oracle represents the results of ground truth sentences extracted by the greedy algorithm, usually as the upper bound. Results with + are token from Cohan et al. (2018), and results with * are token from Xiao and Carenini (2019).

documents with sentence as the smallest unit to alleviate semantic fragility. For the memory module, we set the number of slots to 50 and the dimension of the memory vector to 768, same with the hidden size of the encoder. The iteration number of GAT is set to 2. We use Rouge (Lin, 2004) as the evaluation metric and select the hyperparameters by grid search based on the "Rouge-2" performance on validation sets. Further analysis about the impacts of hyperparameters are discussed in Section 5.2.

We train our model with 2 NVIDIA V100 cards with a small batch size of 16. During the training, we use Adam (Kingma and Ba, 2015) to optimize parameters with a learning rate of 5e-4. An early-stop strategy (Caruana et al., 2000) is applied when valid loss is no longer decent. The extracted sentence number is set to 7 for arXiv dataset and 6 for PubMed dataset according to their average summary length. We report the average results over 5 runs.

5 Results and Analysis

5.1 Main Results

Table 2 presents the results of different models on two datasets. The first section includes traditional approaches and the Oracle; the second and the third sections includes abstractive and extractive models, respectively; and the last section reports ours. Our model with discourse represents that we leverage section information as additional feature (Eq. 2). Several observations deserve to be mentioned.

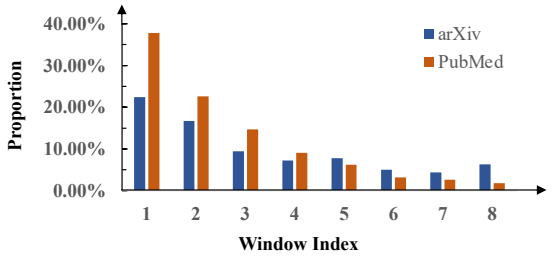

Figure 5: Proportion of sentences selected by each window.

• ***Encoding long texts for abstractive summarization is a challenge***. The vanilla seq2seq with attention model and the pointer network perform rather poorly on the two datasets. A possible reason is that most encoders experience difficulties in modeling long-range contextual dependency when encoding long texts (Vaswani et al., 2017; Frermann and Klementiev, 2019), thereby leading to the inferior performance during the generation (decoding) process.

• ***Global Information Modeling is important for summarizing long documents***. We also observe that Seq2seq-local&global and Topic-

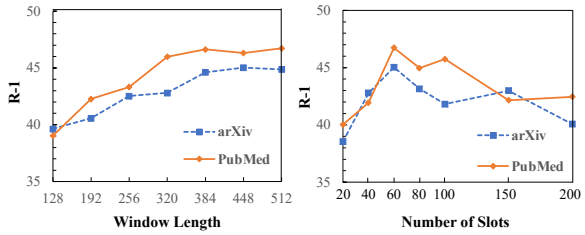

Figure 6: Impact of window length (left) and slot number (right) on model performance (R-1).

Iteration Numbers	Rouge-1	
	arXiv	PubMed
$t = 0$	44.79	46.42
$t = 1$	44.95	46.69
$t = 2$	**45.03**	46.73
$t = 3$	44.97	**46.74**
$t = 4$	45.01	46.71

Table 3: R-1 results on varying iteration numbers t of GAT.

GraphSum show promising results on the two datasets. Both of them explicitly model the global information (e.g., latent topics). Such observation provides a useful instruction for designing the summarization model for long documents.

• ***Our framework is effective***. Our two models substantially outperform all the baselines on two datasets. Figure 5 shows the proportion of sentences selected by each window, where we can see that our model can extract contents from any position of an entire document. By contrast, BERT-Sum and Topic-GraphSum, two BERT-based strong baselines, can only select sentences from the first 512 or 768 words because their truncation setting. This superiority endows our model a higher upper bound when summarizing long documents.

• ***Discourse structure is automatically captured***. The last section of Table 3 shows that the incorporation of discourse information brings no substantial performance gain for our model, though observations in previous studies (Cohan et al., 2018; Xiao and Carenini, 2019) have shown it an effective feature on arXiv and PubMed datasets. A possible reason is that our window-level position encoding has already learned such discourse information because it indicates the window's relative position in the document, while scientific papers are generally organized in specific and relatively fixed structure. This observation implies that the performance of our model does not rely on prior information of datasets. As a result, our model could be easily generalized to long texts of other genres.

5.2 Results on Varying Hyperparameters

We conduct experiments to probe into the impact of several important hyperparameters on model performance, including window length, number of memory slots, and number of memory hops (i.e., iteration number of GAT).

Impact of Window Length. Intuitively, a shorter window means more accurate text encoding. However, it will result in more segments, which is demanding for memory module. Therefore, it is important to find a balanced window length. Figure 6 (left) shows that the overall performance is enhanced when the window length increases from a small value (128). This is because that too short windows suffer from semantic fragility. However, when the window length is set to 368-512, the performance shows a stable trend, implying that the step number and text length are both in a suitable range. For the sake of efficiency, we set the window length to 512 in our experiments.

Impact of Slots Numbers. Figure 6 (right) presents the Rouge-1 results on varying slot numbers. As can be seen, the curves on the two datasets are not monotonous and show a similar trend. In particular, within a particular range where l is relatively small, more slots produce better performance because the memory capacity is improving. However, such increasing trend will reach a saturation when slot number exceeds a threshold, which is 60 in our experiments.

Impact of Iteration Numbers. Recall that in memory layer, we employ a GAT to calculate the interaction between the memory and the window texts. To select the best iteration number (hop number) t, we compare the performance of different t on the validation sets of two datasets. Table 3 shows when t goes from 0 to 2, the performance is slightly boosted. However, this increasing trend is not always monotonous, and a larger t does not bring further substantial gain. To balance the time cost and performance, we select t=2 for the two datasets.

5.3 Effect of Dynamic Memory

In this subsection, we perform quantitative and qualitative investigations to understand the effect

1-th window	4-th window	5-th window
Our full model		
Social isolation and exclusion are associated with poor health status and premature death, while social cohesion, the quality of social relationships and the existence of trust, mutual obligations, and respect in communities, helps to protect people and their health. ..... Good nutrition is important for health and well-being at all stages of the life course; however, its determinants change with age. ..... **Their results suggest that participation in social and cultural activities is beneficial for health, since it helps people to remain active and socially connected, fighting social isolation**.....	..... **We decided to take a snapshot of the metropolitan area of the city of @entity investigating the relationship between adherence to diet or nutritional regimen, BMI, and subjective well-being and the impact of social and cultural participation**. Engagement with community activities, friendships, and meaningful volunteer work are perceived as strategies for maintaining social participation, especially for people with a chronic disease. Thus, encouraging participation in social and cultural activities could be a key tool to fight social isolation and its health detrimental outcomes.....	..... Availability and access to cultural and social activities are a key element of healthy environment, especially of urban environment. **Social isolation can have a negative effect on nutrition, and thus we speculated that social and cultural participation might influence adherence to diet**. ..... Subjective well-being significantly correlates with high self-esteem, and self-esteem shares significant variance in both mental well-being and happiness. Self-esteem has been found to be the most dominant and powerful… …
Ablated model (w/o memory)		
Social isolation and exclusion are associated with poor health status and premature death, while social cohesion, the quality of social relationships and the existence of trust, mutual obligations, and respect in communities, helps to protect people and their health..... Good nutrition is important for health and well-being at all stages of the life course; however, its determinants change with age. ..... **Their results suggest that participation in social and cultural activities is beneficial for health, since it helps people to remain active and socially connected, fighting social isolation**.	..... **We decided to take a snapshot of the metropolitan area of the city of @entity investigating the relationship between adherence to diet or nutritional regimen, BMI, and subjective well-being and the impact of social and cultural participation**. Engagement with community activities, friendships, and meaningful volunteer work are perceived as strategies for maintaining social participation, especially for people with a chronic disease. Thus, encouraging participation in social and cultural activities could be a key tool to fight social isolation and its health detrimental outcomes.	..... Availability and access to cultural and social activities are a key element of healthy environment, especially of urban environment. **Social isolation can have a negative effect on nutrition, and thus we speculated that social and cultural participation might influence adherence to diet**. ..... Subjective well-being significantly correlates with high self-esteem, and self-esteem shares significant variance in both mental well-being and happiness. Self-esteem has been found to be the most dominant and powerful… …

Figure 7: Comparison between the output of our full model (top) and the ablated model (bottom). We use underlined text to denote model-selected sentences and **bold text** to denote the ground truth sentences. The ablated model selects repetitive contents in 4-th window and noisy contents in 5-th window.

of memory module. To this end, we construct an ablated version by removing the memory module and then seek to observe the result difference.

Case Study. Figure 7 provides a case study that compares the selection results of the ablated model and our full model. In 4-th window, the ablated model selects a repetitive sentence, whereas our full model avoids such error. This positive effect is brought by the extraction results preserved in memory module, which serve as a reminder of what information has already been selected. We also note that the ablated model selects wrong sentences in 5-th window. This is because that the model mistakes the "*self-esteem*" as the salient information. By contrast, our model, being aware of previous texts, correctly captures the "*social isolation*" as the core topic and filters the noisy sentences.

Quantitative analysis. In Figure 8, we compare the Rouge scores between our full model and the ablated one. As can be seen, the performance declines dramatically on both datasets when the memory module is removed. This proves that the dynamic memory indeed plays a necessary role in our model.

We further analyze the effecf of memory module in better granularity. Intuitively, the memory module should enhance our model in the following aspects: (1) **Reducing Redundancy**. Our memory module explicitly records the previous predictions and functions like a sentence-level coverage mechanism, which is expected to reduce repetition. (2) **Avoiding Noise**. As discussed in Section 1, segment-wise extraction tend to mistake locally important content as summary sentences due to the lack of global context. Our memory module allows the cross-window information flow and therefore should alleviate this problem. (3) **Perceiving Sentence Length**. The awareness of previous selections may also allow the model to capture sentence length information (Zhong et al., 2019). Ideally, our model is able to adaptively change the change the length of extracted sentence, thereby achieving better performance.

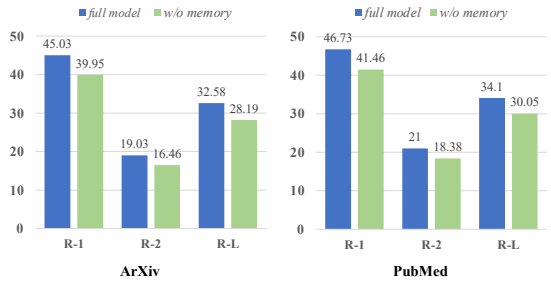

Figure 8: Rouge results of our full model and the ablated version on the two datasets.

To verify our hypothesis, we design three measurements to quantitatively evaluate the model per-

Models	S_{Rep}	S_{Noise}	S_{Len}
arXiv			
w/o memory	0.105	0.118	1.247
Full model	0.033	0.011	0.295
PubMed			
w/o memory	0.107	0.097	1.106
Full model	0.031	0.008	0.343

Table 4: Comparison between our full model and the ablated version. S_{Rep}, S_{Noise} and S_{Len} are the metrics of repetition, noise, and length deviation. Lower is better.

formance on above aspects. Similar to (Zhong et al., 2019), we use $S_{Rep} = 1 - \frac{CountUniq(ngram)}{Count(ngram)}$ to measure the degree of repetition, where $Count(ngram)$ and $CountUniq(ngram)$ are the total and unique number of ngrams of selected sentences. For the noise measurement, we have $S_{noise} = \frac{Count(NoisySent)}{Count(ExtractSent)}$, where $NoisySent$ are the sentences with "R-1" smaller than a threshold. For the length deviation, we have $S_{Len} = \frac{(|sum|-|ref|)}{|ref|}$, where $|sum|$ and $|ref|$ denote the length of model-produced summary and reference summary, respectively.

Table 4 presents the comparison results. The model achieves better performance in three indicators when combined with memory mechanism, consistent with aforementioned analysis.

6 Conclusion and Future Work

In this study, we propose a novel extractive summarization that can summarize long-form documents without content loss. We conduct extensive experiments on two well-studied datasets that consist of scientific papers. Experimental results demonstrate that our model outperforms previous state-of-the-art models. In the future, we will extend our framework (i.e., a sliding encoder combined with long-range memory modeling) to abstractive summarization models.

References

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2017. Faithful to the original: Fact aware neural abstractive summarization. In *AAAI*, pages 4784–4791.

Rich Caruana, Steve Lawrence, and C. Lee Giles. 2000. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in Neural Information Processing Systems 13*, volume 13, pages 402–408.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1662–1675.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 484–494.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 615–621.

Peng Cui, Le Hu, and Yuanchao Liu. 2020. Enhancing extractive text summarization with topic-aware graph neural networks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5360–5371.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. Banditsum: Extractive summarization as a contextual bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748.

Lea Frermann and Alexandre Klementiev. 2019. Inducing document structure for aspect-based summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6263–6273.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, volume 28, pages 1693–1701.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *ICLR 2015 : International Conference on Learning Representations 2015*.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: dynamic memory networks for natural language processing. In *ICML'16 Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, pages 1378–1387.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proc. Workshop on Text Summarization Branches Out, Post Conference Workshop of ACL 2004*, pages 74–81.

Yang Liu and Mirella Lapata. 2019a. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081.

Yang Liu and Mirella Lapata. 2019b. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3728–3738.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2016. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, pages 3075–3081.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

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

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 214–224.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, volume 30, pages 5998–6008.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219.

Danqing Wang, Pengfei Liu, Ming Zhong, Jie Fu, Xipeng Qiu, and Xuanjing Huang. 2019a. Exploring domain shift in extractive text summarization. *arXiv: Computation and Language*.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019b. Multi-passage bert: A globally normalized bert model for open-domain question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5877–5881.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *ICLR 2015 : International Conference on Learning Representations 2015*.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3009–3019.

Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *ICML'16 Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, pages 2397–2406.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Discourse-aware neural extractive model for text summarization.

Jichuan Zeng, Jing Li, Yan Song, Cuiyun Gao, Michael R. Lyu, and Irwin King. 2018. Topic memory networks for short text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3120–3131.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 654–663.