Extractive Summarization of Long Documents by Combining Global and Local Context

by

Wen Xiao

B. Sc, University of Toronto, 2017

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Computer Science)

The University of British Columbia

(Vancouver)

August 2019

© Wen Xiao, 2019
The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

**Extractive Summarization of Long Documents by Combining Global and Local Context**

submitted by *Wen Xiao* in partial fulfillment of the requirements for the degree of **Master of Science in Computer Science**.

**Examiner Committee:**

Giuseppe Carenini, Computer Science  
*Supervisor*

Raymond Ng, Computer Science  
*Second Reader*
Abstract

In this thesis, we propose a novel neural single-document extractive summarization model for long documents, incorporating both the global context of the whole document and the local context within the current topic. We evaluate the model on two datasets of scientific papers, Pubmed and arXiv, where it outperforms previous work, both extractive and abstractive models, on ROUGE-1 and ROUGE-2 scores. We also show that, consistently with our goal, the benefits of our method become stronger as we apply it to longer documents. Besides, we also show that when the topic segment information is not explicitly provided, if we apply a pre-trained topic segmentation model that splits documents into sections, our model is still competitive with state-of-the-art models.
Lay Summary

The goal of this work is to automatically select sentences to form an extractive summary for a given long document, ideally with section information, like scientific papers. Our main idea is that to decide whether a sentence is representative for a document, there are three important factors to be considered, the sentence itself, the local context within the same section of that sentence, and the global context, i.e. what the document is talking about as a whole. To realize our idea, we build a model mainly applying the recurrent neural network and a technique called LSTM-Minus, which has been used in other domain. By the results of our empirical experiments, we show that we have achieved the stat-of-the-art on the scientific papers datasets, and the benefits of our method become stronger as we apply it to longer documents.
Preface

This dissertation is original, independent work by the author, W. Xiao. A compressed version of this dissertation has been accepted to be presented on the EMNLP-IJCNLP 2019 (2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing).
# Table of Contents

Abstract ........................................................................ iii

Lay Summary ................................................................. iv

Preface ........................................................................... v

Table of Contents .......................................................... vi

List of Tables ................................................................... viii

List of Figures .................................................................. ix

Acknowledgments ........................................................... x

1 Introduction ................................................................. 1

2 Related Work .............................................................. 4
   2.1 Traditional Summarization ........................................ 4
   2.2 Neural Extractive Summarization .............................. 5
   2.3 Datasets for long documents ................................. 8
   2.4 Neural Abstractive summarization ...................... 9
   2.5 LSTM-Minus ...................................................... 11
   2.6 Topic Segmentation ........................................... 12
   2.7 Statistical Significance test in NLP ..................... 14

3 Extractive Summarization Model for Long Documents 15
List of Tables

Table 2.1  Comparison of news datasets  . . . . . . . . . . . . . . . . . .  8
Table 4.1  Results on the arXiv dataset  . . . . . . . . . . . . . . . . . .  24
Table 4.2  Results on the Pubmed dataset  . . . . . . . . . . . . . . . . .  25
Table 4.3  Percentage relative improvement of our model  . . . . . . .  26
Table 4.4  Ablation study on Pubmed  . . . . . . . . . . . . . . . . . .  29
Table 4.5  Ablation study on arXiv  . . . . . . . . . . . . . . . . . .  29
Table 4.6  Results on the Bigpatent-A dataset.  . . . . . . . . . . . . .  30
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Extractors compared in [19]</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Extractors compared in [19]</td>
<td>6</td>
</tr>
<tr>
<td>2.3</td>
<td>The structure of Cohan’s model of discourse-aware abstractive summarization [8]</td>
<td>10</td>
</tr>
<tr>
<td>2.4</td>
<td>LSTM-Minus in dependency parsing</td>
<td>11</td>
</tr>
<tr>
<td>2.5</td>
<td>The supervised topic segmentation model proposed in [21]</td>
<td>13</td>
</tr>
<tr>
<td>2.6</td>
<td>The approximate randomization statistical significance test [31]</td>
<td>14</td>
</tr>
<tr>
<td>3.1</td>
<td>Structure of our model</td>
<td>16</td>
</tr>
<tr>
<td>3.2</td>
<td>Details of local context representation</td>
<td>17</td>
</tr>
<tr>
<td>4.1</td>
<td>Results with different length on Pubmed and arXiv</td>
<td>27</td>
</tr>
<tr>
<td>4.2</td>
<td>Relative position plot</td>
<td>28</td>
</tr>
<tr>
<td>4.3</td>
<td>Results with different length on Bigpatent-A</td>
<td>31</td>
</tr>
</tbody>
</table>
I am grateful for the funding received towards my Master program from Huawei.

First and foremost, I would like to thank my supervisor - Prof. Giuseppe Carenini, for his patience, encourage and helpful guidance. He is always willing to help me and share his ideas with me whenever I am stuck. Then I would like to thank my second reader - Prof. Raymond Ng for his time reading my thesis and providing me useful suggestions.

It’s my fortune to gratefully acknowledge the spiritual support of my best friends, Yuekun Gao, Jiayi Wu and Siqi Mei, even though they are not here with me physically, they always cheer me up and encourage me during my hard time.

My great appreciation goes to all my family members for their love and support all the time.

Next, I must express my very profound gratitude to my parents for showing faith in me and giving me liberty to choose what I desired, and how lucky I am to have the best parents in the world. This accomplishment would not have been possible without their unfailing support and continued encourage. There is another family member I would also like to thank, and he is my pet Little White, he has accompanied me for almost ten years, I can always feel better after playing with him, although he can not talk.

Finally, I owe thanks to a very special person, my husband, Zhenan Fan. It is him who always stands by my side supporting me and making me more confident when I feel tired, disappointed and lost. Without him, I would not have had the courage to embark on this journey in the first place.
Chapter 1

Introduction

Single-document summarization is the task of generating a short summary for a given document.\textsuperscript{4} Ideally, the generated summaries should be fluent and coherent, and should faithfully maintain the most important information in the source document. This is a very challenging task, because it arguably requires an in-depth understanding of the source document, and current automatic solutions are still far from human performance.\textsuperscript{1}

Single-document summarization can be either extractive or abstractive. Extractive methods typically pick sentences directly from the original document based on their importance, and form the summary as an aggregate of these sentences. Usually, summaries generated in this way have a better performance on fluency and grammar, but they may contain much redundancy and lack in coherence across sentences. In contrast, abstractive methods attempt to mimic what humans do by first extracting content from the source document and then produce new sentences that aggregate and organize the extracted information. Since the sentences are generated from scratch they tend to have a relatively worse performance on fluency and grammar. Furthermore, while abstractive summaries are typically less redundant, they may end up including misleading or even utterly false statements, because the methods to extract and aggregate information form the source document are still rather noisy.

In this thesis, we focus on extracting informative sentences from a given doc-

\textsuperscript{1}Sentence underlining and Roman numbering will be explained in the result sub-section 4.2.1.
ument (without dealing with redundancy), especially when the document is relatively long (e.g., scientific articles).

Most recent works on neural extractive summarization have been rather successful in generating summaries of short news documents (around 650 words/document) [26] by applying neural Seq2Seq models [4]. However when it comes to long documents, the models tend to struggle with longer sequences because at each decoding step, the decoder needs to learn to construct a context vector capturing relevant information from all the tokens in the source sequence [35].

Long documents typically cover multiple topics. In general, the longer a document is, the more topics are discussed. As a matter of fact, when humans write long documents they organize them in chapters, sections etc.. Scientific papers are an example of longer documents and they follow a standard discourse structure describing the problem, methodology, experiments/results, and finally conclusions [38].

To the best of our knowledge only one previous work in extractive summarization has explicitly leveraged the section information to guide the generation of summaries [9]. However, the only information about sections that is fed into their sentence classifier is a categorical feature with values like Highlight, Abstract, Introduction, Results / Discussion / Analysis, Method, Conclusion, all else, depending on which actual section the sentence appears in. In contrast, in order to exploit section information, we propose to capture a distributed representation of both the global (the whole document) and the local context (e.g., the section/topic) when deciding if a sentence should be included in the summary.

The main contributions of this thesis are as follows:

- In order to capture the local context, we are the first to apply LSTM-minus to text summarization. LSTM-minus is a method for learning embeddings of text spans, which has achieved good performance in dependency parsing [43], in constituency parsing [10], as well as in discourse parsing [24]. With respect to more traditional methods for capturing local context, which rely on hierarchical structures, LSTM-minus produces simpler models i.e. with less parameters, and therefore faster to train and less prone to overfitting.

- We test our method on the Pubmed and arXiv datasets and results appear to
support our goal of effectively summarizing long documents. In particular, while overall we outperform the baseline and previous approaches only by a narrow margin on both datasets, the benefit of our method become much stronger as we apply it to longer documents.

- In order to evaluate our approach, we have created oracle labels for both Pubmed and arXiv [8], by applying a greedy oracle labeling algorithm. These two datasets annotated with extractive labels will be made public.

- When the topic segment information is not available, we first apply a topic segmentation model to split the documents into sections. And we have shown we can achieve a competitive result even using a pre-trained topic segmentation model trained on a completely different corpus.

In Chapter 2, we will introduce some related work on summarization models, datasets, as well as techniques we use in this thesis. The details of our own model will be described in Chapter 3. The implementation details and the experiments with empirical evidence are discussed in Chapter 4. The last Chapter presents our conclusion and possible future work.
Chapter 2

Related Work

2.1 Traditional Summarization

Before neural models, enabled by large datasets, became the most successful approach in summarization, researchers had applied more traditional techniques, like probabilistic models and graph-based methods. Mihalcea and Tarau [25] were the first to introduce graph theory and corresponding algorithms to the NLP area (including the summarization area). They propose to build a graph of text, where the node of the graph are text elements (e.g., the sentences). Then a node ranking technique can be applied to extract the most important information. They test their method on two tasks, keyword extraction and sentence extraction, i.e. extractive summarization. At the same time, Erkan et al. [15] propose a similar graph-based method for multi-document summarization. Later on, Garg et al. [16] improve the sentence extraction model on meeting transcripts by first clustering the sentences, and then build a similar graph as [25] for clusters, instead of single sentences. The reason is that the meeting transcripts are usually incomplete, ill-formed sentences with high redundancy, and also contain chit-chat, which has nothing to do with their main topic, by using cluster, they could avoid including such sentences. Besides, they use the cosine similarity of two sentences rather than directly counting the common words, and prove that this would lead to a better performance. Tixier et al. [39] propose a submodular based summarization model. They try to first extract keywords by building a graph of words and selecting important words[40].
then select sentences containing high scores keywords, and finally build the extractive summary by maximizing a submodular function within a certain budget. Unlike the ranking score used in Textrank, which tends to give nodes with more important connections higher scores in the graph, without any cohesiveness consideration, they hypothesize that important words are more likely to be found among the influential spreaders, which are the nodes that not only have many important connections but also in a dense substructures of these connections. Thus they use CoreRank[40] as score of each word, because it can capture the spreading influence of a node.

2.2 Neural Extractive Summarization

Benefiting from the success of neural sequence models in other NLP tasks, Cheng and Lapata [4] propose a novel approach to single-document extractive summarization based on neural networks and continuous sentence features, which outperforms traditional methods on the DailyMail dataset. In particular, they develop a general encoder-decoder architecture, where a CNN is used as sentence encoder, a uni-directional LSTM as document encoder, with another uni-directional LSTM as decoder. Besides, they extend their structure to an abstractive stage, by changing the sequence-labeling decoder to an attention-based word generator, but the result shows that the abstractive method is not as promising as the extractive model. To decrease the number of parameters while maintaining the accuracy, Nallapati et al. [27] present SummaRuNNer, a simple RNN-based sequence classifier without decoder, outperforming or matching the model of [4]. They take content, salience, novelty, and position of each sentence into consideration when deciding if a sentence should be included in the extractive summary. Yet, they do not capture any aspect of the topical structure, as we do in this thesis. So their approach would arguably suffer when applied to long documents, likely containing multiple and diverse topics.

While SummaRuNNer is tested only on news, Kedzie and McKeown [19] carry out a comprehensive set of experiments with deep learning models of extractive summarization across different domains, i.e. news, personal stories, meetings, and
Figure 2.1: One of the extractors compared in [19], model (a) is a simple RNN model, model (b) is an attention-based encoder-decoder model.

Figure 2.2: One of the extractors compared in [19], model (c) is the extractor proposed in [4], model (d) is the extractor proposed in [27].
medical articles, as well as across different neural architectures, in order to better understand the general pros and cons of different design choices. They compare different sentence encoders (CNN, RNN, and Average Word Embedding) and sentence extractors, and the extractors they compare are shown in Figure 2.1 - 2.2:

- Extractor (a): a simple bidirectional RNN model. First, the sentence embeddings are encoded through a bidirectional RNN, and then the forward and backward hidden states for each sentence are concatenated and passed through a Multi-Layer Perceptron with a logistic sigmoid. The final output is the probability that the sentence is selected.

- Extractor (b): an attention-based encoder-decoder model. The sentence embeddings are first encoded through a bidirectional RNN, which is the same as extractor (a), and then there is a bidirectional decoder with attention mechanism, in which the outputs of the decoder are the query vectors attending to the encoder. After that, the concatenation of attended encoder output and decoder output are concatenated and passed through a Multi-Layer Perceptron with a logistic layer. The final output is also the probability that the sentence is selected.

- Extractor (c): this extractor is first proposed in [4], which is an auto-regressive encoder-decoder structure. The sentences are encoded through a RNN encoder, with the last hidden state of the encoder as the initial hidden state of the decoder. The sentence embeddings are fed into the decoder and then passed to a Multi-Layer Perceptron with a logistic layer. The result is the probability that the sentence is part of the extractive summary as well.

- Extractor (d): this extractor is proposed in [27]. The first step is also to encode sentence embeddings through a RNN encoder, and then there is a document representation by averaging the RNN output and a summary representation by taking the weighted sum of the RNN output until the current stage with weight being the extraction probability of corresponding sentence. The decision is made based on the RNN output, document representation, summary representation as well as positions of the sentence in the document.
<table>
<thead>
<tr>
<th>Datasets</th>
<th># docs</th>
<th>avg. doc. length (# words)</th>
<th>avg. summary length(# words)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>92K</td>
<td>656</td>
<td>43</td>
</tr>
<tr>
<td>Daily Mail</td>
<td>219K</td>
<td>693</td>
<td>52</td>
</tr>
<tr>
<td>NY Times</td>
<td>655K</td>
<td>530</td>
<td>38</td>
</tr>
<tr>
<td>PubMed</td>
<td>133K</td>
<td>3016</td>
<td>203</td>
</tr>
<tr>
<td>arXiv</td>
<td>215K</td>
<td>4938</td>
<td>220</td>
</tr>
<tr>
<td>Bigpatent</td>
<td>1,341K</td>
<td>3573</td>
<td>117</td>
</tr>
<tr>
<td>Bigpatent-A</td>
<td>193K</td>
<td>3521</td>
<td>110</td>
</tr>
</tbody>
</table>

Table 2.1: Comparison of news datasets, scientific paper datasets and the recently proposed patent dataset, [8][36]

They find that non auto-regressive sentence extraction performs at least as well as auto-regressive extraction in all domains, where by auto-regressive sentence extraction they mean the previous predictions is used to inform future predictions. Furthermore, they find that the Average Word Embedding sentence encoder works at least as well as encoders based on CNN and RNN. In light of these findings, our model is not auto-regressive and uses the Average Word Embedding encoder.

2.3 Datasets for long documents

[12] provide a comprehensive overview of the current datasets for summarization. Noticeably, most of the larger-scale summarization datasets consists of relatively short documents (less than 1000 words/document), like CNN/DailyMail [26] and New York Times [32]. One exception is [8] that recently introduce two large-scale datasets of long and structured scientific papers obtained from arXiv and PubMed. These two new datasets contain much longer documents than all the news datasets (See table 2.1) and are therefore ideal test-beds for the method we present in this thesis. Recently, a new dataset for summarization, Bigpatent, is proposed in [36]. The dataset consists of 1.3 million U.S. patent documents collected from Google Patents Public Datasets, and the authors use the patent’s abstract as summary while the corresponding description as the input document. Based on their result, they claim that when compared with other datasets (news and scientific papers), the summaries in their dataset contain a richer discourse structure with more repeating
entities, and the salient content is evenly distributed throughout the input. Besides, the documents tend to be as long as in the Pubmed dataset, while the summaries tend to be shorter, showing the higher compression ratio of this new dataset. There are 9 different categories in this dataset, and in this thesis, we only do the experiment on the first category (Bigpatent-A).

The main problem for experiments on this dataset is that there is no section information as in scientific papers. To solve the problem, we apply a pre-trained topic segmentation model to split the whole documents into sections. And we will introduce the previous works on topic segmentation in Sec 2.6.

2.4 Neural Abstractive summarization

Recently, more researchers have started to use Neural Network to generate abstractive summaries, especially after the large dataset CNN/Dailymail corpus is introduced by Nallapati et al. in [26] to the summarization community (it is originally used for the task of passage-based question answering([18])). They propose the encoder-decoder RNN model with attention, as well as several variants with neural tricks, like using pointer-generator switch and/or hierarchical structure. See et al. ([34]) apply the same idea as pointer-generator switch, and they propose the pointer-generator model for summarization, in which there is a generation probability determining whether generating words from the vocabulary, or copying words from the source documents. They show that they significantly outperform the abstractive state-of-the-art result at that time. However, the model works only for relatively short documents, for which the need for summarization is limited. As See et al. mention in their paper, they truncate the articles to 400 tokens.

While most current neural abstractive summarization models focus on summarizing relatively short news articles, few researchers have started to investigate the summarization of longer documents by exploiting their natural structure. Celikyilmaz et al. [3] present an encoder-decoder architecture to address the challenges of representing a long document for abstractive summarization. The encoding task is divided across several collaborating agents, each is responsible for a subsection of text through a multi-layer LSTM with word attention. Generally speaking, their model seems however overly complicated when it comes to the extractive summa-
Cohan et al. [8] also propose a model for abstractive summarization taking the structure of documents into consideration with a hierarchical approach, and test it on longer documents with section information, i.e. scientific papers. In particular, they apply a hierarchical encoder at the word and section levels. Then, in the decoding step, they combine the word attention and section attention to obtain a context vector at each state. Finally a score is computed based on the context vector, and the word is chosen as the one with highest score. The model is shown in Figure 2.3.

Figure 2.3: The structure of Cohan’s model of discourse-aware abstractive summarization [8]

This approach to capture discourse structure is however quite limited both in general and especially when you consider its application to extractive summarization. First, their hierarchical method has a large number of parameters and it is therefore slow to train and likely prone to overfitting. Secondly, it does not take the global context of the whole document into account, which is arguably critical in extractive methods, when deciding on the salience of a sentence (or even a word).

1To address this, they only process the first 2000 words of each document, by setting a hard threshold in their implementation, and therefore loosing information.
The extractive summarizer we present in this thesis does not suffer from these limitations by adopting the parameter lean LSTM-minus method, and by explicitly modeling the global context.

2.5 LSTM-Minus

The LSTM-Minus method is first proposed in [43] as a novel way to learn sentence segment embeddings for graph-based dependency parsing, i.e. estimating the most likely dependency tree given an input sentence. For each dependency pair, they divide a sentence into three segments (prefix, infix and suffix), and the LSTM-Minus is used to represent each segment. They apply a single LSTM to the whole sentence and use the difference between two hidden states $h_j - h_i$ to represent the segment from word $w_i$ to word $w_j$, as shown in Figure 2.4. This enables their model to learn segment embeddings from information both outside and inside the segments, enhancing their model’s ability to access sentence-level information. The intuition behind the method is that each hidden vector $h_t$ can capture useful information before and including the word $v_t$.

Shortly after, [10] use the same method on the task of constituency parsing, as the representation of a sentence span, extending the original uni-directional LSTM-Minus to the bi-directional case. More recently, inspired by the success of LSTM-Minus in both dependency and constituency parsing, [24] extend the
technique to discourse parsing. They propose a two-stage model consisting of an intra-sentential parser and a multi-sentential parser, learning contextually informed representations of constituents with LSTM-Minus, at the sentence and document level, respectively.

Similarly, in this thesis, when deciding if a sentence should be included in the summary, the local context of that sentence is captured by applying LSTM-Minus at the document level, to represent the sub-sequence of sentences of the document (i.e., the topic/section) the target sentence belongs to.

2.6 Topic Segmentation

Topic segmentation is the task of dividing a document into segments, such that the sentences within each segment are topically cohesive, while the cut-off point should mark the change of topic. This provides a basic structure of documents that can be useful for summarization.

The traditional topic segmentation models are mostly unsupervised, due to the lack of large-scale labeled data. Riedl and Biemann[30] employ a method based on the topic assigned by the Bayesian Inference method of LDA. They define a coherence score between pairs of sentences, and identify segment boundaries by large score drops between pairs of adjacent sentences. Another noteworthy approach is GRAPHSEG[17], an unsupervised graph-based approach, which builds a semantic relatedness graph, where nodes represent sentences and edges are created by semantically related sentence pairs. Then they split the topic segments by finding maximal cliques in the relatedness graph. This is arguably better because instead of approximating the meaning of the sentence with its topics as in [30], it explicitly leverages the semantic relatedness between sentences.

Koshore et al. [21] introduce a large-scale natural dataset - WIKI-727K dataset, which is extracted from Wikipedia with sections as the topic segments, and propose a supervised hierarchical neural network model to solve the problem, as shown in Figure 2.5 For a given document \( d = (s_1, s_2, \ldots, s_n) \), the output of the model would be \((y_1, y_2, \ldots, y_n)\), where \( y_i \) indicates the probability that sentence \( s_i \) is the end of a topic segment. After that, the final decision, whether a sentence should be the boundary of two topics, is made by setting a threshold \( t \) on the value of \( y_i \).
essentially, only the sentences with probability higher than $t$ will be categorized as topic boundaries. In a series of experiments, they show that their model outperform other unsupervised models on natural datasets (except for one synthesized automatically[6]), and that it generalizes well to unseen natural text.

In this thesis, when topic segment information is not available, we will use the pre-trained model proposed in [21] to split the whole long documents into sections first.

**Figure 2.5:** The supervised topic segmentation model proposed in [21]
2.7 Statistical Significance test in NLP

Riezler and Maxwell [31] study some deficiencies in the discriminatory ability of Machine Translation evaluation metrics (NIST, BLEU, F1) and the accuracy of statistical significance tests. In particular, they show an example that if the difference between two system in BLEU is as small as 0.3%, then the confidence levels are assessed as 70%, in which case, the two system can not be considered to have significant difference. This highlight the fact that if the differences between the results of multiple systems are small, then a statistical significance test is critically needed. Based on their experiments, they show that approximate randomization[28] can estimate the p-value more conservatively, when compared with the popular statistical significance testing - bootstrap test [14], which increasing the likelihood of type-I error for the latter. Based on their findings, Kedzie et al. [19] choose the approximate randomization as the methods for statistical significance test.

Thus, following their work, we use the approximate randomization statistical significance test on all the results shown in this thesis. The algorithm to compute p-value is shown in figure 2.6.

Figure 2.6: The approximate randomization statistical significance test. [31]
Chapter 3

Extractive Summarization Model for Long Documents

In this thesis, we propose an extractive model for long documents, incorporating local and global context information, motivated by natural topic-oriented structure of human-written long documents. The architecture of our model is shown in Figure 3.1, each sentence is visited sequentially in the original document order, and a corresponding confidence score is computed expressing whether the sentence should be included in the extractive summary. Our model comprises three components: the sentence encoder, the document encoder and the sentence classifier.

3.1 Sentence Encoder

The goal of the sentence encoder is mapping sequences of word embeddings to a fixed length vector (See bottom center of Figure 3.1). There are several common methods to embed sentences. For extractive summarization, RNN were used in [27], CNN in [4], and Average Word Embedding in [19]. [19] experiment with all the three methods, and conclude that Word Embedding Averaging is as good or better than either RNNs or CNNs for sentence embedding across different domains and summarizer architectures. So in this work, we use the Average Word Embedding as our sentence encoder, by which a sentence embedding is simply the

\[^1\text{We do not deal with redundancy in this thesis.}\]
Figure 3.1: The structure of our model, $se_i, sr_i$ represent the sentence embedding and sentence representation of sentence $i$, respectively. The binary decision of whether the sentence should be included in the summary is based on the sentence itself (A), the whole document (B) and the current topic (C). The document representation is simply the concatenation of the last hidden states of the forward and backward RNNs, while the topic segment representation is computed by applying LSTM-Minus, as the details shown in Fig 3.2.
Figure 3.2: Detail of C, the topic segment representation is computed by applying LSTM-Minus. The RNN in red rectangle is the Document Encoder, the same as the one in the red rectangle in Fig. 3.1.

average of its word embeddings, i.e. $se = \frac{1}{n} \sum_{w_0}^{w_n} emb(w_i), se \in \mathbb{R}^{d_{emb}}$.

Besides, we also tried the popular pre-trained BERT sentence embedding [13], but initial results were rather poor. So we do not pursue this possibility any further.

3.2 Document Encoder

At the document level, a bi-directional recurrent neural network [33] is often used to encode all the sentences sequentially forward and backward, with such model
achieving remarkable success in machine translation [2]. As units, we selected gated recurrent units (GRU) [5], in light of favorable results shown in [7]. The GRU is represented in a standard fashion with \( r, z, n \) representing the reset, update, and new gates, respectively.

\[
\begin{align*}
    r_t &= \sigma(W_{ir}e_t + b_{ir} + W_{hr}h_{t-1} + b_{hr}) \\
    z_t &= \sigma(W_{iz}e_t + b_{iz} + W_{hz}h_{t-1} + b_{hz}) \\
    n_t &= \tanh(W_{in}e_t + b_{in} + r_t(W_{hn}h_{t-1} + b_{hn})) \\
    h_t &= (1 - z_t)n_t + z_t h_{t-1}
\end{align*}
\]

The output of the bi-directional GRU for each sentence \( t \) comprises two hidden states, \( h^f_t \in \mathbb{R}^{d_{hid}}, h^b_t \in \mathbb{R}^{d_{hid}} \) as forward and backward hidden state, respectively.

**A. Sentence representation** As shown in Figure 3.1(A), for each sentence \( t \), the sentence representation is the concatenation of both backward and forward hidden state of that sentence.

\[
sr_t = (h^f_t : h^b_t), sr_t \in \mathbb{R}^{d_{hid} \times 2}
\]

In this way, the sentence representation not only represents the current sentence, but also partially covers contextual information both before and after this sentence.

**B. Document representation** The document representation provides global information on the whole document. It is computed as the concatenation of the final state of the forward and backward GRU, labeled as B in Figure 3.1 [22]

\[
d = (h^f_n : h^b_0), d \in \mathbb{R}^{d_{hid} \times 2}
\]

**C. Topic segment representation** To capture the local context of each sentence, namely the information of the topic segment that sentence falls into, we apply the LSTM-Minus method\(^2\), a method for learning embeddings of text spans. LSTM-Minus is shown in detail in Figure 3.2, each topic segment is represented as the subtraction between the hidden states of the start and the end of that topic. As illustrated in Figure 3.2 the representation for section 2 of the sample document (containing three sections and eight sentences overall) can be computed as \([f_5 −\]

\(^2\)In the original paper, LSTMs were used as recurrent unit. Although we use GRUs here, for consistency with previous work, we still call the method LSTM-Minus
\[ f_2, b_3 - b_6 \], where \( f_5, f_2 \) are the forward hidden states of sentence 5 and 2, respectively, while \( b_3, b_6 \) are the backward hidden states of sentence 3 and 6, respectively. In general, the topic segment representation \( l_t \) for segment \( t \) is computed as:

\[
\begin{align*}
    f_t &= h_{end_t}^f - h_{start_t-1}^f, f_t \in \mathbb{R}^{d_{hid}} \\
    b_t &= h_{start_t}^b - h_{end_t+1}^b, b_t \in \mathbb{R}^{d_{hid}} \\
    l_t &= (f_t : b_t), l_t \in \mathbb{R}^{d_{hid} \times 2}
\end{align*}
\]

where \( start_t, end_t \) is the index of the beginning and the end of topic \( t \), \( f_t \) and \( b_t \) denote the topic segment representation of forward and backward, respectively. The final representation of topic \( t \) is the concatenation of forward and backward representation \( l_t \). To obtain \( f_t \) and \( b_t \), we utilize subtraction between GRU hidden vectors of \( start_t \) and \( end_t \), and we pad the hidden states with zero vectors both in the beginning and the end, to ensure the index can not be out of bound. The intuition behind this process is that the GRUs can keep previous useful information in their memory cell by exploiting reset, update, and new gates to decide how to utilize and update the memory of previous information. In this way, we can represent the contextual information within each topic segment for all the sentences in that segment.

### 3.3 Decoder

Once we have obtained a representation for the sentence, for its topic segment (i.e., local context) and for the document (i.e., global context), these three factors are combined to make a final prediction \( p_t \) on whether the sentence should be included in the summary. We consider two ways in which these three factors can be combined.

**Concatenation** We can simply concatenate the vectors of these three factors as,

\[
input_t = (d \cdot l_t \cdot sr_t), input_t \in \mathbb{R}^{d_{hid} \times 6}
\]

where sentence \( i \) is part of the topic \( t \), and \( input_t \) is the representation of sentence \( i \) with topic segment information and global context information.

**Attentive context** As local context and global context are all contextual informa-
tion of the given sentence, we use an attention mechanism to decide the weight of each context vector, represented as

\[
\begin{align*}
score^d_i &= v^T \tanh(W_a(d : sr_i)) \\
score^l_i &= v^T \tanh(W_a(l_i : sr_i)) \\
weight^d_i &= \frac{score^d_i}{score^d_i + score^l_i} \\
weight^l_i &= \frac{score^l_i}{score^d_i + score^l_i} \\
context_i &= weight^d_i \ast d + weight^l_i \ast l_i \\
input_i &= (sr_i : context_i), \ input_i \in \mathbb{R}^{d_{hid+4}}
\end{align*}
\]

where the \( context_i \) is the weighted context vector of each sentence \( i \), and assume sentence \( i \) is in topic \( t \).

Then there is a final multi-layer perceptron(MLP) followed with a sigmoid activation function indicating the confidence score for selecting each sentence:

\[
\begin{align*}
h_i &= \text{Dropout}(\text{ReLU}(W_{mlp}input_i + b_{mlp})) \\
p_i &= \sigma(W_hh_i + b_h).
\end{align*}
\]
Chapter 4

Experiments

To validate our method, we set up experiments on the two scientific paper datasets (arXiv and PubMed). With ROUGE scores and METEOR score as automatic evaluation metrics, we compare with previous works, both abstractive and extractive. Besides, we also do a series experiments on an additional dataset, Bigpatent, in which that documents do not contain the section information.

4.1 General Experiment Settings

In this section, we will introduce the details and general settings of our experiments on all the datasets.

4.1.1 Training

We minimized the weighted negative log-likelihood during training, where the weight is computed as \( w_{pos} = \frac{\#negative}{\#positive} \), to solve the problem of highly imbalanced data (typical in extractive summarization).

\[
\mathcal{L} = - \sum_{d=1}^{N} \sum_{i=1}^{N_d} (w_{pos} \cdot y_i \log p(y_i | W, b) + (1 - y_i) \log p(y_i | W, b))
\]

where \( y_i \) represent the ground-truth label of sentence \( i \), with \( y_i = 1 \) meaning sentence \( i \) is in the gold-standard extract summary.
4.1.2 Extractive Label Generation

In the Pubmed and arXiv datasets, the extractive summaries are missing. So we follow the work of [19] on extractive summary labeling, constructing gold label sequences by greedily optimizing ROUGE-1 on the gold-standard abstracts, which are available for each article. \(^1\) The pseudo code is shown in Algorithm 1.

Algorithm 1 Extractive label generation

```python
function LABEL_GENERATION(Reference, sentences, lengthLimit)
    hyp = ''
    wc = 0
    picked = []
    highest_r1 = 0
    sid = -1
    while wc <= lengthLimit do
        for i in range(len(sentences)) do
            score = ROUGE(hyp + sentences[i], ref)
            if score > highest_r1 then
                highest_r1 = score
                sid = i
            end if
        end for
        if sid != -1 then
            picked.append(sid)
            hyp = hyp + sentences[sid]
            wc += NumberOfWords(sentences[sid])
        else
            break
        end if
    end while
    return picked
end function
```

\(^1\) For this, we use a popular python implementation of the ROUGE score to build the oracle. Code can be found here, [https://pypi.org/project/py-rouge/](https://pypi.org/project/py-rouge/)
4.1.3 Implementation Details

We train our model using the Adam optimizer [20] with learning rate 0.0001. We use a mini-batch with a batch size of 32 documents, and the size of the GRU hidden states is 300. The pre-trained word embedding we use is GloVe [29] with dimension 300, trained on the Wikipedia and Gigaword. The vocabulary size of our model is 50000. And the drop out rate we use in the experiments is 0.3. All the above parameters were set based on [19] without any fine-tuning. Again following [19], we train each model for 30 epochs, and the best model is selected with early stopping on the validation set according to Rouge-2 F-score.

4.1.4 Models for Comparison

We perform a systematic comparison with previous work in extractive summarization. For completeness, we also compare with recent neural abstractive approaches. In all the experiments, we use the same train/val/test splitting.

- Traditional extractive summarization models: SumBasic [41], LSA [37], and LexRank [15] (Only available on scientific paper datasets)
- Neural abstractive summarization models: Attn-Seq2Seq [26], Pntr-Gen-Seq2Seq [34] and Discourse-aware [8] (Only available on scientific paper datasets)
- Neural extractive summarization models: Cheng&Lapata [4] and SummaRuNNer [27]. Based on [19], we use the Average Word Encoder as sentence encoder for both models, instead of the CNN and RNN sentence encoders that were originally used in the two systems, respectively.  
- Baseline: Similar to our model, but without local context and global context, i.e. the input to MLP is the sentence representation only.
- Lead: Given a length limit of $k$ words for the summary, Lead will return the first $k$ words of the source document.
- Oracle: uses the Gold Standard extractive labels, generated based on ROUGE (Sec. 4.1.2).

\[ https://github.com/kedz/nnsum/tree/emnlp18-release \]
4.2 Experiments on Scientific Paper Datasets

In this section, we will show the results of the experiments on the two scientific paper datasets - Pubmed and arXiv.

4.2.1 Results and analysis

<table>
<thead>
<tr>
<th>Model</th>
<th>Rouge-l</th>
<th>Rouge-2</th>
<th>Rouge-L</th>
<th>Meteor</th>
</tr>
</thead>
<tbody>
<tr>
<td>SumBasic*</td>
<td>29.47</td>
<td>6.95</td>
<td>26.30</td>
<td>-</td>
</tr>
<tr>
<td>LSA*</td>
<td>29.91</td>
<td>7.42</td>
<td>25.67</td>
<td>-</td>
</tr>
<tr>
<td>LexRank*</td>
<td>33.85</td>
<td>10.73</td>
<td>28.99</td>
<td>-</td>
</tr>
<tr>
<td>Attn-Seq2Seq*</td>
<td>29.30</td>
<td>6.00</td>
<td>25.56</td>
<td>-</td>
</tr>
<tr>
<td>Pntr-Gen-Seq2Seq*</td>
<td>32.06</td>
<td>9.04</td>
<td>25.16</td>
<td>-</td>
</tr>
<tr>
<td>Discourse-aware*</td>
<td>35.80</td>
<td>11.05</td>
<td>31.80</td>
<td>-</td>
</tr>
<tr>
<td>Baseline</td>
<td>42.91</td>
<td>16.65</td>
<td>28.53</td>
<td>21.35</td>
</tr>
<tr>
<td>Cheng &amp; Lapata</td>
<td>42.24</td>
<td>15.97</td>
<td>27.88</td>
<td>20.97</td>
</tr>
<tr>
<td>SummaRuNNer</td>
<td>42.81</td>
<td>16.52</td>
<td>28.23</td>
<td>21.35</td>
</tr>
<tr>
<td>Ours-attentive context</td>
<td><strong>43.58</strong></td>
<td><strong>17.37</strong></td>
<td><strong>29.30</strong></td>
<td><strong>21.71</strong></td>
</tr>
<tr>
<td>Ours-concat</td>
<td><strong>43.62</strong></td>
<td><strong>17.36</strong></td>
<td><strong>29.14</strong></td>
<td><strong>21.78</strong></td>
</tr>
<tr>
<td>Lead</td>
<td>33.66</td>
<td>8.94</td>
<td>22.19</td>
<td>16.45</td>
</tr>
<tr>
<td>Oracle</td>
<td>53.88</td>
<td>23.05</td>
<td>34.90</td>
<td>24.11</td>
</tr>
</tbody>
</table>

Table 4.1: Results on the arXiv dataset. For models with an *, we report results from [8]. Models are traditional extractive in the first block, neural abstractive in the second block, while neural extractive in the third block. The Oracle (last row) corresponds to using the ground truth labels, obtained (for training) by the greedy algorithm, see Section 4.1.2. Results that are not significantly distinguished from the best systems are bold.

For evaluation, we follow the same procedure as in [19]. Summaries are generated by selecting the top ranked sentences by model probability \( p(y_i|W,b) \), until the length limit is met or exceeded. Based on the average length of abstracts in these two datasets, we set the length limit to 200 words. We use ROUGE scores\(^3\) [23] and METEOR scores [11] between the model results and ground-

\(^3\)We use a modified version of rouge\_papier, a python wrapper of ROUGE-1.5.5, [https://github.com/kedz/rouge\_papier](https://github.com/kedz/rouge_papier). The command line is ‘Perl ROUGE-1.5.5 -e data -a -n 2 -r 1000 -f A -z SPL config_file’
<table>
<thead>
<tr>
<th>Model</th>
<th>Rouge-1</th>
<th>Rouge-2</th>
<th>Rouge-L</th>
<th>Meteor</th>
</tr>
</thead>
<tbody>
<tr>
<td>SumBasic*</td>
<td>37.15</td>
<td>11.36</td>
<td>33.43</td>
<td>-</td>
</tr>
<tr>
<td>LSA*</td>
<td>33.89</td>
<td>9.93</td>
<td>29.70</td>
<td>-</td>
</tr>
<tr>
<td>LexRank*</td>
<td>39.19</td>
<td>13.89</td>
<td>34.59</td>
<td>-</td>
</tr>
<tr>
<td>Attn-Seq2Seq*</td>
<td>31.55</td>
<td>8.52</td>
<td>27.38</td>
<td>-</td>
</tr>
<tr>
<td>Pntr-Gen-Seq2Seq*</td>
<td>35.86</td>
<td>10.22</td>
<td>29.69</td>
<td>-</td>
</tr>
<tr>
<td>Discourse-aware*</td>
<td>38.93</td>
<td>15.37</td>
<td>35.21</td>
<td>-</td>
</tr>
<tr>
<td>Baseline</td>
<td>44.29</td>
<td>19.17</td>
<td>30.89</td>
<td>20.56</td>
</tr>
<tr>
<td>Cheng &amp; Lapata</td>
<td>43.89</td>
<td>18.53</td>
<td>30.17</td>
<td>20.34</td>
</tr>
<tr>
<td>SummaRuNNer</td>
<td>43.89</td>
<td>18.78</td>
<td>30.36</td>
<td>20.42</td>
</tr>
<tr>
<td>Ours-attentive context</td>
<td><strong>44.81</strong></td>
<td><strong>19.74</strong></td>
<td><strong>31.48</strong></td>
<td><strong>20.83</strong></td>
</tr>
<tr>
<td>Ours-concat</td>
<td><strong>44.85</strong></td>
<td><strong>19.70</strong></td>
<td><strong>31.43</strong></td>
<td><strong>20.83</strong></td>
</tr>
<tr>
<td>Lead</td>
<td>35.63</td>
<td>12.28</td>
<td>25.17</td>
<td>16.19</td>
</tr>
<tr>
<td>Oracle</td>
<td>55.05</td>
<td>27.48</td>
<td>38.66</td>
<td>23.60</td>
</tr>
</tbody>
</table>

Table 4.2: Results on the Pubmed dataset. For models with an *, we report results from [8]. See caption of Table 4.1 above for details on compared models. Results that are not significantly distinguished from the best systems are bold.

The performance of all models on arXiv and Pubmed is shown in Table 4.1 and Table 4.2, respectively. We use the approximate randomization as the statistical significance test method [31] with the Bonferroni correction to the multiple comparison problem, at the confident level 0.01 ($p < 0.01$).

As we can see from these tables, on both datasets, the neural extractive models outperform the traditional extractive models on informativeness (ROUGE-1,2)
<table>
<thead>
<tr>
<th>Dataset-versus</th>
<th>Rouge-1(%)</th>
<th>Rouge-2(%)</th>
<th>Rouge-L(%)</th>
<th>Meteor(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>arXiv-SR</td>
<td>+1.9</td>
<td>+5.1</td>
<td>+3.2</td>
<td>+2.0</td>
</tr>
<tr>
<td>arXiv-BSL</td>
<td>+1.7</td>
<td>+4.3</td>
<td>+2.1</td>
<td>+2.0</td>
</tr>
<tr>
<td>Pubmed-SR</td>
<td>+2.2</td>
<td>+4.9</td>
<td>+3.5</td>
<td>+2.0</td>
</tr>
<tr>
<td>Pubmed-BSL</td>
<td>+1.3</td>
<td>+2.8</td>
<td>+1.7</td>
<td>+1.3</td>
</tr>
<tr>
<td>Macro avg-SR</td>
<td>+2.0</td>
<td>+5.0</td>
<td>+3.4</td>
<td>+2.0</td>
</tr>
<tr>
<td>Macro avg-BSL</td>
<td>+1.5</td>
<td>+3.5</td>
<td>+1.9</td>
<td>+1.7</td>
</tr>
</tbody>
</table>

Table 4.3: Percentage relative improvement of our model, when compared with the SummaRuNNer (SR) and Baseline (BSL) models on both datasets (first and second block). The third block shows Macro average relative improvement across the two datasets.

by a wide margin, but results are mixed on ROUGE-L. Presumably, this is due to the neural training process, which relies on a goal standard based on ROUGE-1. Exploring other training schemes and/or a combination of traditional and neural approaches is left as future work. Similarly, the neural extractive models also dominate the neural abstractive models on ROUGE-1,2, but these abstractive models tend to have the highest ROUGE-L scores, possibly because they are trained directly on gold standard abstract summaries.

Compared with other neural extractive models, our models (both with attentive context and concatenation decoder) have better performances on all three ROUGE metrics as well as METEOR score. In particular, the improvements over the Baseline model show that the local and global contextual information does help to identify the most important sentences. Interestingly, just the Baseline model already achieves a slightly better performance than previous works; possibly because the auto-regressive approach used in those models is even more detrimental for long documents. The details of these key comparisons are revealed in Table 4.3 which shows the percentage relative improvements of our model over the Baseline and SummaRuNNer on both datasets, as well as their macro averages.

Figure 4.1 shows the most important result of our analysis: the benefits of our method, explicitly designed to capture global and local context for dealing with longer documents, do actually become stronger as we apply it to longer documents. As it can be seen in the Figure, the performance gain of our model with respect to
Figure 4.1: A Comparison between our model, SummaRuNNer and Oracle when applied to documents with increasing length, left-up: ROUGE-1 on Pubmed dataset, right-up: ROUGE-2 on Pubmed dataset, left-down: ROUGE-1 on arXiv dataset, right-down: ROUGE-2 on arXiv dataset.

its closest neural competitor is more pronounced for documents with $\geq 3000$ words.

Finally, the result of Lead (Table 4.1, 4.2) shows that scientific papers have less position bias than news; i.e., the first sentences of these papers are not a good choice to form an extractive summary.

Figure 4.2 shows the relative position of our predicted sentences, oracle sentences and the section borders in the documents, with the documents uniformly sampled from the highest ROUGE score(left) to the lowest ROUGE score(right). The interesting result is that our method has the preference on the first sentences and last sentences from the first section and the last section, which are most likely Introduction and Conclusion of the scientific papers, respectively, even though there is no explicit information on the positions.

As a teaser for the potential and challenges that still face our approach, its out-
Figure 4.2: The relative position in documents of our predicted sentences, oracle sentences, and the section borders, and the documents are sampled uniformly from the highest ROUGE score (left) to lowest ROUGE score (right). The upper figure shows the position distribution of Pubmed, and the lower one shows the position distribution of arXiv.

(Paragraph: put (i.e., the extracted sentences) when applied to this thesis is underlined and the order in which the sentences are extracted is marked with the Roman numbering. They are all located in the Introduction chapter, and distributed from the introductory paragraph to the contribution part. It can be found that the most confident three sentences are the ones stating the motivation, explaining the intuition, and describing the experiments. If we increase the length limit to the number of words in our abstract, three more sentences are extracted, which do seem to provide useful complementary information. Not surprisingly, some redundancy is present, as dealing explicitly with redundancy is not a goal of our current proposal and left as future work.)
Table 4.4: Ablation study on the Pubmed dataset. Baseline is the model with sentence representation only, Baseline+segment is the model with sentence and local topic information, Baseline+doc is the model with sentence and global document information, and the last one is the full model with concatenation decoder. Results that are not significantly distinguished from the best systems are bold.

<table>
<thead>
<tr>
<th>Model</th>
<th>Rouge-1</th>
<th>Rouge-2</th>
<th>Rouge-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>44.29</td>
<td>19.17</td>
<td>30.89</td>
</tr>
<tr>
<td>Baseline+local</td>
<td>44.85</td>
<td>19.77</td>
<td>31.51</td>
</tr>
<tr>
<td>Baseline+global</td>
<td>44.06</td>
<td>18.83</td>
<td>30.53</td>
</tr>
<tr>
<td>Baseline+global+local(concat)</td>
<td>44.85</td>
<td>19.70</td>
<td>31.43</td>
</tr>
</tbody>
</table>

Table 4.5: Ablation study on the arXiv dataset. The model descriptions refer to Table 4.4. Results that are not significantly distinguished from the best systems are bold.

<table>
<thead>
<tr>
<th>Model</th>
<th>Rouge-1</th>
<th>Rouge-2</th>
<th>Rouge-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>42.91</td>
<td>16.65</td>
<td>28.53</td>
</tr>
<tr>
<td>Baseline+local</td>
<td>43.57</td>
<td>17.35</td>
<td>29.29</td>
</tr>
<tr>
<td>Baseline+global</td>
<td>42.90</td>
<td>16.58</td>
<td>28.36</td>
</tr>
<tr>
<td>Baseline+global+local(concat)</td>
<td>43.62</td>
<td>17.36</td>
<td>29.14</td>
</tr>
</tbody>
</table>

4.2.2 Ablation Study

To investigate the influence that each part of our model makes, we do the ablation study on the concatenation decoder with the ROUGE scores as evaluation metric, and the results are shown in Table 4.4, 4.5. The same as Section 4.2.1, we use the approximate randomization as the statistical significance test method [31] with the Bonferroni correction to the multiple comparison problem, at the confident level 0.01 ($p < 0.01$).

From these tables, we can see that the performances significantly improve with the additional topic information, for both Baseline and Baseline+global models. It indicates that the topic information do have relation to deciding if a sentence should be part of the summary, and the LSTM-Minus method helps to catch such information. But adding the global information does not always improve the performances, in contrast, the performance is even worse when adding the global in-
### Table 4.6: Results on the Bigpatent-A dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Rouge-1</th>
<th>Rouge-2</th>
<th>Rouge-L</th>
<th>Meteor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>35.44</td>
<td>10.79</td>
<td>23.95</td>
<td>15.08</td>
</tr>
<tr>
<td>Baseline + local</td>
<td>35.62</td>
<td>10.86</td>
<td>24.05</td>
<td>15.19</td>
</tr>
<tr>
<td>Baseline + global</td>
<td>35.75</td>
<td>10.95</td>
<td>24.06</td>
<td>15.26</td>
</tr>
<tr>
<td>Cheng &amp; Lapata</td>
<td>35.77</td>
<td>10.86</td>
<td>24.07</td>
<td>15.22</td>
</tr>
<tr>
<td>SummaRuNNer</td>
<td>35.79</td>
<td>10.94</td>
<td>24.04</td>
<td>15.25</td>
</tr>
<tr>
<td>Ours-attentive context</td>
<td>35.62</td>
<td>10.82</td>
<td>23.96</td>
<td>15.17</td>
</tr>
<tr>
<td>Ours-concat</td>
<td>35.62</td>
<td>10.84</td>
<td>24.02</td>
<td>15.16</td>
</tr>
<tr>
<td>Lead</td>
<td>31.27</td>
<td>8.64</td>
<td>21.58</td>
<td>12.93</td>
</tr>
<tr>
<td>Oracle</td>
<td>45.92</td>
<td>16.32</td>
<td>29.95</td>
<td>18.51</td>
</tr>
</tbody>
</table>

formation to the Baseline model. It might because that global information we use is always the same for all the sentences in one document, which might not be useful to distinguish the summary sentences from other sentences. To solve the problem, exploring a more sentence-specific global representation is needed, and we would leave it as one of the future work.

### 4.3 Experiment on Bigpatent - Long documents without topic segment information

For documents in the Bigpatent corpus, there is no natural topic segment information as scientific papers do, so in this case, we need to first employ a topic segmentation methods to split documents into sections. In essence, this experiment is a preliminary exploration of whether our method can still deliver useful results when topic segmentation is done automatically. The topic segmentation method we use in this experiment is the pre-trained model proposed by [21], trained on the WIKI-727k corpus. And the dataset we use is the BigPatent-A (Human Necessities), a subset of the new dataset proposed in [36]. For all the models, we set the length limit of result summaries to be 100 words, based on the average length of the ground truth summaries.

All the results on this dataset are shown in Table 4.6. There is no model that significantly outperforms the others. The Figure 4.3 shows the ROUGE scores of documents with different lengths. Although in this experiment, the performance of our
model is not improving, compared with others, as the documents being longer, our model is still competitive with the state-of-the-art extractive summarization model. One possible reason for this is that the pre-trained topic segmentation model does not work well on this dataset, since it was trained on Wikipedia, which is quite different from patents. For instance, there is an obvious distinct between sections in Wikipedia, while the patent documents tend to cover topics that are not sufficiently distinct. Furthermore, since we do not have ground truth topic segment information for this dataset, we can not evaluate how accurate the result of topic segment model is. We will leave the study of this issue for future work.
Chapter 5

Conclusion and Future work

In this thesis, we propose a novel extractive summarization model especially designed for long documents, by incorporating the local context within each topic, along with the global context of the whole document. Our approach is based on the fact that when human write long documents, they tend to include multiple topics and organize them in a structured way. Technically, our proposal integrates recent findings on neural extractive summarization in a parameter-lean and modular architecture.

Our main contribution is that we apply the LSTM-Minus method to the extractive summarization model, as the way to generate the representation of one topic segment, e.g. the sections in a scientific paper. This technique had been successfully used in graph-based dependency parsing [43], constituency parsing [10] and discourse parsing [24] as a representation of a text span. In this thesis, we show that it can be effectively applied to extractive summarization.

We evaluate our model and compare with previous works in both extractive and abstractive summarization on two large scientific paper datasets, Pubmed and arXiv, which contain documents that are much longer than in previously used corpora, and with natural topic segment information (section). Our model not only achieves state-of-the-art on these two datasets, but in an additional experiment, in which we consider documents with increasing length, it becomes more competitive for longer documents. Besides, although we do not explicitly use the position information of each sentence and section, the result shows that our model prefers the
sentences located at the beginning or the end of the first and last sections, which is most likely the introduction and conclusion, respectively. We also performed an ablation study to test the effect of each module in our proposed model, and the results suggest that the local context itself could improve the baseline model significantly, while adding the global context does not have an obvious effect.

We also test our model on a new dataset, Bigpatent, with long documents but without natural topic segment information. In this case, we first apply a pre-trained topic segmentation model to split the documents, and then apply our model to generate extractive summary. Despite the fact that the topic segmentation model may be inaccurate, our summarizer still achieves result competitive with the state-of-the-art model. However, when the length of documents increase, we do not find the same benefits as in the scientific paper datasets.

For future work, we will try to deal with redundancy of our generated summaries, one possible way is to have a summary representation storing the information of the summary at each timestep[27]. After that, it could be beneficial to integrate explicit features, like sentence position and salience, into our neural approach. As another venue for future work, we will also explore how to apply hierarchical model that can leverage the section information, like Hierarchical Attention Networks(HAN) [44], or hierarchical transformer[42]. More generally, we plan to combine traditional and neural models, as suggested by our results.

Furthermore, we would like to explore more sophisticated structure of documents, like discourse tree, instead of rough topic segments. Besides, we would also like to explore ways to combine the pre-trained multi-task language models (like BERT[13] or XLNet[45]) with the natural structure of documents to generate extractive summaries.

More long term, we will study how extractive/abstractive techniques can be integrated. Initially, the output of an extractive system could be fed into an abstractive one, training the two jointly. Then, we would consider a finer-grain integration, where the combination of abstractive/extractive techniques is tailored to the particular source document.
Bibliography


document summarization with explicit information selection modeling. In
Language Processing, pages 1787–1796, Brussels, Belgium, Oct.-Nov.
2018. Association for Computational Linguistics. URL
https://www.aclweb.org/anthology/D18-1205. → page 18


Empirical Methods in Natural Language Processing, pages 1289–1298,
Copenhagen, Denmark, Sept. 2017. Association for Computational
Linguistics. doi:10.18653/v1/D17-1133. URL
https://www.aclweb.org/anthology/D17-1133. → pages 2, 11, 32

Language Processing, 2004. URL
http://www.aclweb.org/anthology/W04-3252 → page 4

Abstractive text summarization using sequence-to-sequence RNNs and
beyond. In Proceedings of The 20th SIGNLL Conference on Computational
Association for Computational Linguistics. doi:10.18653/v1/K16-1028
URL https://www.aclweb.org/anthology/K16-1028. → pages 2, 8, 9, 23

network based sequence model for extractive summarization of documents.
In Proceedings of the Thirty-First AAAI Conference on Artificial
Intelligence, AAAI’17, pages 3075–3081. AAAI Press, 2017. URL
http://dl.acm.org/citation.cfm?id=3298483.3298681 → pages
5, 6, 7, 15, 23, 33


