SEHY: A Simple yet Effective Hybrid Model for Summarization of Long Scientific Documents

Zhihua Jiang[1], **Junzhan Yang**[1], **Dongning Rao**[2*]
[1] Department of Computer Science, Jinan University, Guangzhou 510632, P. R. China
[2] School of Computer, Guangdong University of Technology, Guangzhou 510006, P. R. China
tjiangzhh@jnu.edu.cn, junyz@stu2021.jnu.edu.cn, raodn@gdut.edu.cn

Abstract

Long-document summarization has been recently recognized as one of the most important natural language processing (NLP) tasks, yet one of the least solved ones. Extractive approaches attempt to choose salient sentences via understanding the whole document, but long documents cover numerous subjects with varying details and will not ease content understanding. Instead, abstractive approaches elaborate to generate related tokens while suffering from truncating the source document due to their input sizes. To this end, we propose a *S*imple yet *E*ffective *HY*brid approach, which we call *SEHY*, that exploits the discourse information of a document to select salient sections instead sentences for summary generation. On the one hand, *SEHY* avoids the full-text understanding; on the other hand, it retains salient information given the length limit. In particular, we design two simple strategies for training the extractor: extracting sections incrementally and based on salience-analysis. Then, we use strong abstractive models to generate the final summary. We evaluate our approach on a large-scale scientific paper dataset: arXiv. Further, we discuss how the disciplinary class (e.g., computer science, math or physics) of a scientific paper affects the performance of *SEHY* as its writing style indicates, which is unexplored yet in existing works. Experimental results show the effectiveness of our approach and interesting findings on arXiv and its subsets generated in this paper.

1 Introduction

Long-document tasks (e.g., scientific papers summarization (Cohan et al., 2018) and long-text reading comprehension (Wen et al., 2021)) have become one of long-term challenging tasks in Natural Language Processing (NLP) because long documents cover numerous subjects with varying details and will not ease content understanding. For example, scientific papers, whose abstracts can be used as ground-truth summaries, is a representative type of long documents with discourse information showing the hierarchical structure composed of tokens, sentences, paragraphs, and sections (K and Mathew, 2020). Extractive summarization approaches select important units such as phrases or sentences from the original text, but long documents cover numerous subjects with varying details and will not ease content understanding (Nallapati et al., 2017; Xiao and Carenini, 2020). Instead, abstractive summarization approaches concisely paraphrase the information content while suffering from truncating the source document due to their input sizes (Rohde et al., 2021; Guo et al., 2021).

Hybrid models exhibit a combination solution via first extracting salient sentences with an extractive model (i.e., extractor) and then generating a summary based on extracted sentences with an abstractive model (i.e., generator) (Gidiotis and Tsoumakas, 2020; Pilault et al., 2020). However, on the one hand, training an extractive model may be expensive due to the complex salience analysis; on the other hand, an abstractive model may generate inappropriate summary words due to the dependence on extracted sentences. Thus, pipeline-style errors can be propagated and accumulated, leading to hybrid models perform worse than current state-of-the-art (SoTA) abstractive models (Rohde et al., 2021; Guo et al., 2021). This suggests that exploring simple yet effective extractive approaches is crucial to improve the overall performance and decrease the training cost of a hybrid model.

Recently, the success of pre-trained language models (PTMs) such as Transformer (Vaswani et al., 2017) in NLP brings great gain for abstractive models in the summarization task. However, Transformer-based models usually suffer from the quadratic dependency on the sequence length due to their full attention mechanism. Sometimes, the model's performance is mainly con-

*Corresponding author: Dongning Rao.

Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022, pages 96–106
November 20–23, 2022.

strained by its limitation on the sequence length. For instance, the average document length on arXiv (Cohan et al., 2018) is more than 6000 tokens while BART (Lewis et al., 2020), which combines BERT (Devlin et al., 2019) and GPT (Radford and Narasimhan, 2018), has a comparatively smaller length limit, 1024 tokens. Besides, for a hybrid model, extracted sentences from its extractive model are often difficult to maintain the coherence of the source document, thus leading to the poor semantic representations by its abstractive model (Cai et al., 2019).

To alleviate these issues, we propose a novel *S*imple yet *E*ffective *HY*brid approach, which we call *SEHY*, that exploits the discourse information of a document to select salient sections instead sentences for summary generation. We use simple strategies for choosing sections, not only for decreasing the training cost of the extractor, but also for enhancing the input-sequence's coherence to the generator. Motivated by (Gidiotis and Tsoumakas, 2020), which identifies and selects specific sections that are more informative, we propose two strategies: choosing specific sections (e.g., Introduction or Conclusion) based on the salience analysis and using the beginning sections without concerning the salience. After this, we use strong abstractive models to generate the final summary.

To demonstrate the effectiveness of *SEHY*, we answer the following questions in this paper:

- Q1: which strategy is better?
- Q2: how do different abstractive models affect the overall performance of *SEHY*?
- Q3: can we have the equivalent result when summarizing different scientific papers?

As the contents indicate, Q1 is used to evaluate the two section-extraction strategies, Q2 is used to measure different abstractive models which are responsible to generate the final summary, and Q3 is used to estimate writing styles of scientific papers in different disciplines. The joint of Q1 and Q2 acts as ablation studies on the proposed hybrid model *SEHY*. While, Q3 is not explored yet in existing works where all scientific papers on arXiv are summarized without distinguishing their disciplinary properties (e.g., computer science, math or physics). For instance, a well-written computer science paper usually presents summary sentences in the Introduction or Conclusion section, but no experimental work has ever confirmed this.

2 Related Work

Automatic text summarization is the task of producing a concise and fluent summary while preserving key information content and overall meaning. It aims to transform lengthy documents into shortened versions, something which could be difficult and costly to undertake if done manually. In this section, we focus on recent summarization models. For more text summarization technologies, we refer interested readers to a survey on this (Allahyari et al., 2017).

2.1 Extractive Models

Extractive methods select important sentences and rearrange them as the summary, instead of generating summary tokens. LexRank (Erkan and Radev, 2011) is an early extractive model, which computes sentence importance based on the concept of eigenvector centrality in a graph representation of sentences. SummaRuNNer (Nallapati et al., 2017) is a Recurrent Neural Network (RNN) based sequence model for extractive summarization of documents. It has the additional advantage of being interpretable, since it allows visualization of its predictions broken up by abstract features, such as information content, salience, and novelty. Xiao et al. (Xiao and Carenini, 2020) found that redundancy is a very serious problem when summarizing long documents. They proposed ExtSum-LG+Rd, which achieved high ROUGE scores, while reducing redundancy significantly.

2.2 Abstractive Models

Early abstractive models include Pointer-Generator Networks (PGN) (See et al., 2017), which augments two shortcomings: inaccuracy and repetition, via copying words from the source text and using coverage to keep track of what has been summarized. Cohan et al. (Cohan et al., 2018) built two large-scale scientific-paper datasets: arXiv and Pubmed. They also proposed Discourse composed of a hierarchical encoder that models the discourse structure of a document and an attentive discourse-aware decoder that generates the summary. PEGASUS (Zhang et al., 2020) is a Transformer-based encoder-decoder model trained on massive text corpora with a new self-supervised objective.

Recent works improve the performance of Transformer-based models by increasing the input length or the model size. BigBird (Zaheer et al., 2020) exhibits a sparse attention mecha-

nism that reduced the quadratic dependency to linear. DeepPyramidion (Pietruszka et al., 2022) proposes representation pooling as a method to sparsify attention in Transformer by learning to select the most-informative token representations during the training process. HAT-BART (Rohde et al., 2021) proposes a new Hierarchical Attention Transformer-based architecture into the denoising auto-encoder BART (Lewis et al., 2020). LongT5 (Guo et al., 2021) attempts to increase both at the same time. Specifically, it integrates attention ideas from long-form transformer (Beltagy et al., 2020a), and adopts pretraining strategies from PEGASUS into the scalable T5 architecture (Raffel et al., 2020a). Top Down Transformer (Pang et al., 2022) updates token representations in a bottom-up and top-down manner: token representations are first inferred in the bottom-up pass and then updated in the top-down pass to capture long-range dependency.

Even though Top Down Transformer is at the top of the arXiv leaderboard[1] while LongT5 takes the second place, the authors of Top Down Transformer did not release their model or code yet. Thus, we regard LongT5 as the current SoTA with respect to all open-sourced document summarization models.

2.3 Hybrid Models

A hybrid approach takes advantage of extractive and abstractive approaches. DANCER (Gidiotis and Tsoumakas, 2020) proposes a divide-and-conquer algorithm, which breaks a long document and its summary into multiple source-target pairs and uses them for training a model that learned to summarize each part of the document. TLM-I+E (Pilault et al., 2020) performs a simple extractive step, which is used to condition the transformer language model on relevant information before being tasked with generating a summary. Although mostly follows the abstractive approach, Top Down Transformer connects to the hybrid models via learning and assigning importance weight with the importance tagger resembles an extractive step.

2.4 Paper Abstract Generation

Scientific papers are representatives of long documents with discourse information, where their abstracts can be used as ground-truth summaries. Wang et al. (Wang et al., 2018) presented a paper abstract writing system based on an attentive neural sequence-to-sequence model that can take a title as input and automatically generate an abstract. They designed a novel Writing-editing Network that can attend to both the title and the previously generated abstract drafts and then iteratively revise and polish the abstract. Next year, they further developed a Paper-Robot (Wang et al., 2019) which performs as an automatic research assistant by incrementally writing some key elements of a new paper based on memory-attention networks. Demir et al. (Demir et al., 2019) proposed a dataset with LaTeX source files on recent open-source computer vision papers and experimented with recent methods such as Transformer and Transformer-XL (Dai et al., 2019) to generate consistent LaTeX code.

[1] https://paperswithcode.com/dataset/arxiv

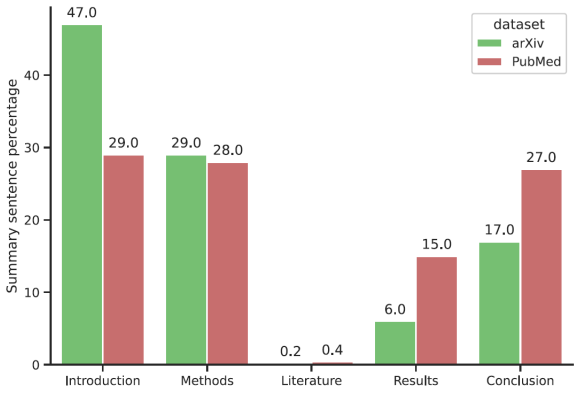

Figure 1: The distribution of summary sentences per section type, cited from (Gidiotis and Tsoumakas, 2020).

3 Method

In this section, we first present two strategies to implement our extractive model (for answering Q1), then describe multiple paired abstractive models (for answering Q2), and finally explain how to generate data subsets with regard to disciplinary categories of scientific papers (for answering Q3).

3.1 Two Extraction Strategies

Long documents introduce a lot of noise to the summarization process. Indeed, one of the major difficulties in summarizing a long document is that large parts of the document are not really key to its narrative and thus should be ignored. Following DANCER (Gidiotis and Tsoumakas, 2020), we identify and select specific sections that are more informative. This reduces the noise and the computational cost in processing a long document. Figure 1 demonstrates the distribution of summary sentences per section type. We observe that the ma-

jority of summary sentences, for the arXiv dataset, are assigned to the *introduction* section followed by the *methods* and *conclusion* sections. Based on that observation, we select and use only the sections that are classified *introduction*, *methods*, and *conclusion* ignoring the others. This simple method very effectively allows us to filter out parts of the article that are less important for the summary and leads to summaries that are more focused. Another benefit of selecting sections instead of sentences is that, the number of sections is much smaller than that of sentences, which decreases the number of combinations dramatically.

In particular, we use the following two strategies for selecting sections. Formally, supposing there are *N* sections in a source document *Doc*:

- $P_{sal}(Sec)$: using all the sections included in $Sec = \{sec_1, sec_2, ..., sec_{|Sec|}\}$ where $|Sec| \leq N$;
- $P_{inc}(k)$: only using the first *k* sections where $1 \leq k \leq N$ is a positive integer.

We sequentially concatenate selected sections from the beginning of a document as the above strategies indicate. If exceeding the length limit, the concatenated sequence will be truncated; otherwise, it will be padded with zero. All section headings can be conveniently identified from the LaTeX source files. On the one hand, to simplify the salience analysis of $P_{sal}(Sec)$, we focus on the first section (i.e., Head Section), the last section (i.e., Tail Section), and the combination of these two (i.e., Head+Tail Section), for the target of determining Sec. On the other hand, we can set $k > 1$ for $P_{inc}(k)$ to cover *introduction* and *methods* as shown in Figure 1. However, the actual values of k are usually no more than the relative ratio of the length limit divided by the average section-length on experimental datasets, because larger k values will not bring greater gain due to the truncation mechanism of the abstractive model.

Obviously, one weakness of this method is that, although these section categories are meaningful when working on academic articles, if the proposed method is extended to different domains (e.g. financial documents), then a new categorization of sections would be required. Thus, exploring more sophisticated methods that use machine learning to identify the type of each section should be explored in future work.

Table 1: Examples of the head and tail section names of scientific papers on arXiv.

Head Section Name	Tail Section Name
Introduction	Conclusion
Related Works	Conclusions
Introduction and related work	Discussion
Motivation	Future Work
Background and Introduction	Further Work
Motivation and Background	Observations
Motivating Work	Concluding remarks

3.2 Tested Abstractive Model

We test five strong abstractive models introduced in the Related Work section, whose actual parameter settings are shown in Table 7.

- T5 (Raffel et al., 2020b). T5 introduces a unified framework that converts all text-based language problems into a text-to-text format and combines the insights from the exploration with scale and the new corpus.
- BART (Lewis et al., 2020). BART is a denoising auto-encoder for pre-training sequence-to-sequence models. It is trained by corrupting text with an arbitrary noising function, and learning a model to reconstruct the original text.
- LED (Beltagy et al., 2020b). LED is a Longformer (Beltagy et al., 2020a) variant for supporting long document generative tasks. The Longformer's attention mechanism scales linearly with sequence length, making it easy to process super-long documents.
- BigBird (Zaheer et al., 2020). Bigbird introduces a sparse attention mechanism that reduces the quadratic dependency to linear. It reveals some benefits of having global tokens (e.g., CLS), that attend to the entire sequence as part of the sparse attention mechanism.
- PEGASUS (Zhang et al., 2020). Pegasus is a pre-training large Transformer-based encoder-decoder models on massive text corpora with a new self-supervised objective. Important sentences are removed or masked from an input document and generated together as one output sequence from the remaining sentences.

3.3 Data Subset Generation

Academic papers of the arXiv dataset are collected from the scientific repository arXiv.org and are writ-

Table 2: The number of disciplinary papers for the Train/Dev/Test split.

Discipline \ Split[1]	Train	Dev	Test
Physics	146628	5145	5193
Mathematics	19146	296	257
Computer Science	9600	361	339
Statistics	2354	80	77
Quantitative Biology	1492	54	60
Quantitative Finance	612	19	25
E.E.S.S.	259	5	10
Economics	14	1	2
Total (the full arXiv)	203038	6437	6640

[1] E.E.S.S. is shorthand for Electrical Engineering and Systems Science.

Table 3: The average length of Abstract, Head Section and Tail Section on arXiv and its subsets.

Dataset \ Section[1]	Abstract	Head	Tail
Full (the full arXiv)	151	748	724
CS (Computer Science)	158	857	537
Math (Mathematics)	122	1036	1059
Phy (Physics)	154	645	720

[1] Head and Tail indicate Head Section and Tail Section, respectively.

ten in LaTeX[2]. Following previous work (Cohan et al., 2018; Demir et al., 2019), we extract the top-level section headings from the LaTeX source files using Pandoc[3]. We collect various section heading names and classify them into equivalent categories. For instance, names of Head Section and Tail Section are shown in Table 1.

The arXiv dataset covers various disciplines, including physics, mathematics, computer science, quantitative biology, and economics, etc. We statistics the paper numbers of different disciplines following the train/dev/test split of (Cohan et al., 2018), as shown in Table 2. It shows that the arXiv papers are primarily collected from three disciplines: Physics, Mathematics and Computer Science. Thus, to answer Q3, we generate three subsets of the full arXiv dataset[4]: CS (Computer Science), Math (Mathematics) and Phy (Physics). For the convenience of writing, we use "Full" to indicate the full arXiv dataset in this paper. To better determine the super-parameters of $P_{sal}(Sec)$ and $P_{inc}(k)$, we calculate the average lengths of Head Section (H) and Tail Section (T) of Full, CS, Math, Phy, as shown in Table 3.

[2]https://www.latex-project.org/

[3]https://pandoc.org

[4]We use the article ID extracted from the LaTeX file of a scientific paper to determine its discipline class. Specifically, we search the article ID on arXiv and get the "class" field of the returned result page as the discipline class.

4 Experiment

4.1 Settings

We conduct all experiments on a local machine (Windows 10 + GTX 1060 3GB) and a workstation (Ubuntu18.04, a NVIDIA Tesla V100 36G GPU, and a Intel(R) Xeon(R) E5-2698 v4 @ 2.20GHz CPU). Our code is written in Python 3.7. The deep learning platform is Pytorch 1.8.0. We use the huggingface-transformers[5] for pre-training and fine-tuning summary models. The actual parameter settings of all tested models are shown in Table 7.

We evaluate multiple variants of our approach on the largest-scale scientific-paper dataset: arXiv, with ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) as the measurement metric. We report the F1 scores of ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L), using the pyrouge package[6]. ROUGE is suitable for summarization of scientific papers, whose human-written abstracts can be used as ground-truth summaries. We do not include human evaluation, following the previous works such as LongT5 (Guo et al., 2021), BigBird (Zaheer et al., 2020) and PEGASUS (Zhang et al., 2020), etc. It is quite challenging to run human evaluations for scientific papers, as it requires participants to possess sophisticated domain-specific background knowledge.

4.2 Results and Analysis

In this section, we exhibit the evaluation results of multiple variants of our approach *SEHY* equipped with different section-selection strategies and different summary-generation models. We also answer the mentioned-above three questions (Q1, Q2, and Q3) to reveal interesting experimental findings.

Evaluation results of $P_{sal}(Sec)$. We report the ROUGE scores of *SEHY* using $P_{sal}(Sec)$ paired with three base models (Table 4) and three large models (Table 5) on arXiv (D_{Full}) and its three disciplinary subsets (D_{CS}, D_{Math} and D_{Phy}), respectively.

In Table 4, we find that: (1) all tested base models paired with $P_{sal}(H+T)$ obtain the highest scores, showing the advantage of using both of Head Section and Tail Section against using only one of them; (2) most tested base models paired with $P_{sal}(H)$ perform better than the same models paired with $P_{sal}(T)$, demonstrating that Head Section (usually *introduction*) contributes more than

[5]https://github.com/huggingface/transformers

[6]https://pypi.org/project/pyrouge

Table 4: Evaluation results of *SEHY* using the policy P_{sal} paired with *base* abstractive models on arXiv and its subsets. ROUGE scores (%) are reported. Best results in each group are in bold.

Dataset+Policy[1] \ Model	T5-base R-1 / R-2 / R-L	LED-base R-1 / R-2 / R-L	BART-base R-1 / R-2 / R-L
$D_{Full} + P_{sal}(H)$	38.75 / 13.93 / 34.50	43.67 / 16.87 / 39.29	43.48 / 16.25 / 38.86
$D_{Full} + P_{sal}(T)$	39.71 / 14.86 / 35.53	42.02 / 15.81 / 37.80	42.85 / 16.13 / 38.42
$D_{Full} + P_{sal}(H+T)$	**47.09 / 19.84 / 42.30**	**47.55 / 19.99 / 42.88**	**44.84 / 17.37 / 40.11**
$D_{CS} + P_{sal}(H)$	43.00 / 15.90 / 38.69	43.23 / 16.14 / 39.54	44.53 / 16.57 / 40.65
$D_{CS} + P_{sal}(T)$	40.22 / 14.71 / 36.13	40.91 / 15.04 / 37.18	41.93 / 16.08 / 38.10
$D_{CS} + P_{sal}(H+T)$	**47.58 / 19.91 / 43.11**	**46.67 / 18.86 / 42.93**	**45.46 / 17.32 / 41.59**
$D_{Math} + P_{sal}(H)$	39.93 / 15.62 / 36.06	41.28 / 16.75 / 37.67	40.83 / 15.41 / 36.69
$D_{Math} + P_{sal}(T)$	30.81 / 9.31 / 27.71	33.44 / 10.60 / 30.38	34.37 / 11.90 / 30.86
$D_{Math} + P_{sal}(H+T)$	**44.05 / 18.77 / 39.68**	**43.18 / 18.15 / 39.25**	**41.82 / 15.83 / 37.63**
$D_{Phy} + P_{sal}(H)$	38.22 / 13.57 / 33.96	41.04 / 15.03 / 36.69	43.10 / 16.11 / 24.83
$D_{Phy} + P_{sal}(T)$	39.88 / 15.00 / 35.64	42.39 / 16.11 / 38.11	43.44 / 16.40 / 38.89
$D_{Phy} + P_{sal}(H+T)$	**46.76 / 19.75 / 41.93**	**47.20 / 19.88 / 42.52**	**44.39 / 17.14 / 39.59**

[1] "Full" indicates the full arXiv dataset. CS, Math and Phy are shorthand for Computer Science, Mathematics and Physics, respectively. H and T are shorthand for Head Section and Tail Section. "H+T" indicates the concatenation of H and T.

Table 5: Evaluation results of *SEHY* using the policy P_{sal} paired with *large* abstractive models on arXiv and its subsets. ROUGE scores (%) are reported. Best results in each group are in bold.

Dataset+Policy \ Model[1]	BART-large R-1 / R-2 / R-L	BigBird-large R-1 / R-2 / R-L	PEGASUS-large R-1 / R-2 / R-L
$D_{Full} + P_{sal}(H)$	45.06 / 17.18 / 40.38	35.95 / 12.01 / 30.69	43.28 / 16.50 / 38.57
$D_{Full} + P_{sal}(T)$	**47.34 / 19.24 / 42.47**	28.49 / 7.75 / 24.58	40.43 / 14.88 / 35.73
$D_{Full} + P_{sal}(H+T)$	46.84 / 18.56 / 42.01	**47.33 / 19.57 / 39.97**	**45.23 / 18.22 / 40.42**
$D_{CS} + P_{sal}(H)$	47.78 / 18.66 / 43.68	46.31 / 19.11 / 40.84	46.05 / 18.65 / 42.12
$D_{CS} + P_{sal}(T)$	46.78 / 18.63 / 42.70	40.67 / 14.49 / 35.27	41.63 / 15.51 / 36.99
$D_{CS} + P_{sal}(H+T)$	**48.22 / 19.19 / 44.14**	**49.37 / 20.69 / 42.99**	**47.71 / 19.62 / 43.52**
$D_{Math} + P_{sal}(H)$	44.52 / 16.79 / 40.21	43.20 / 18.03 / 37.65	43.85 / 18.27 / 39.73
$D_{Math} + P_{sal}(T)$	42.54 / 15.33 / 38.04	32.91 / 10.50 / 28.17	32.79 / 10.62 / 28.77
$D_{Math} + P_{sal}(H+T)$	**44.53 / 16.97 / 40.49**	**46.05 / 19.67 / 39.48**	**44.62 / 18.84 / 39.94**
$D_{Phy} + P_{sal}(H)$	45.23 / 17.05 / 40.25	42.92 / 16.15 / 36.19	43.14 / 16.37 / 38.33
$D_{Phy} + P_{sal}(T)$	**47.83 / 19.12 / 42.80**	28.80 / 7.94 / 24.77	40.92 / 15.18 / 36.12
$D_{Phy} + P_{sal}(H+T)$	45.20 / 17.42 / 40.20	**47.42 / 19.66 / 39.93**	**45.25 / 18.32 / 40.37**

[1] Both of Bigbird-Pegasus-large (Zaheer et al., 2020) and Pegasus-large (Zhang et al., 2020) have been fine-tuned on arXiv, quoted from their original literature.

Table 6: Comparisons between *SEHY* and other summarization approaches on the full arXiv dataset D_{Full}. ROUGE scores (%) are reported. The three highest scores are in bold.

Type	Approach[1]	R-1 / R-2 / R-L[2]
Abstractive	PGN** (See et al., 2017)	32.06 / 9.04 / 25.16
	Discourse* (Cohan et al., 2018)	35.80 / 11.05 / 31.80
	PEGASUS* (Zhang et al., 2020)	44.67 / 16.95 / 38.83
	BigBird* (Zaheer et al., 2020)	46.63 / 19.02 / 41.77
	HAT-BART* (Rohde et al., 2021)	46.68 / 19.07 / 42.17
	DeepPyramidion* (Pietruszka et al., 2022)[3]	47.15 / **19.99**†† / -
	LongT5* (Guo et al., 2021)[4]	**48.35**† / **21.92**† / **44.27**†
Extractive	LexRank** (Erkan and Radev, 2011)	33.85 / 10.73 / 28.99
	SummaRuNNer** (Nallapati et al., 2017)	42.81 / 16.52 / 28.23
	ExtSum-LG+Rd* (Xiao and Carenini, 2020)	44.01 / 17.79 / 39.09
Hybrid	DANCER* (Gidiotis and Tsoumakas, 2020)	45.01 / 17.60 / 40.56
	TLM-I+E* (Pilault et al., 2020)	41.62 / 14.69 / 38.03
Ours	*SEHY*:$D_{Full} + P_{sal}(H+T)$+T5-base	47.09 / **19.84**††† / 42.30
	SEHY:$D_{Full} + P_{sal}(H+T)$+LED-base	**47.55**†† / **19.99**†† / **42.88**††
	SEHY:$D_{Full} + P_{sal}(H+T)$+BART-base	44.84 / 17.37 / 40.11
	SEHY:$D_{Full} + P_{sal}(T)$+BART-large	**47.34**††† / 19.24 / **42.47**†††
	SEHY:$D_{Full} + P_{sal}(H+T)$+BigBird-large	47.33 / 19.57 / 39.97
	SEHY:$D_{Full} + P_{sal}(H+T)$+PEGASUS-large	45.23 / 18.22 / 40.42

[1] * indicates the results are from leaderboard (https://paperswithcode.com/dataset/arxiv). ** indicates the results are from their original papers.
[2] The †, †† and ††† indicate the highest, the second high and the third high score, respectively.
[3] DeepPyramidion only reported the R-1 and R-2 scores in its original paper (Pietruszka et al., 2022), so far on leaderboard.
[4] LongT5 is the current state of the art (SoTA) among all open-source summarization models.

Table 7: Parameter settings of abstractive models.

Parameter \ Model	T5	BART	LED	BigBird	PEGASUS
Version	base	base	base	large	large
Batch	8	6	7	6	6
Layer	12	6	6	16	16
Epoch	3	3	3	1	1
Min Loss[1]	1.84	2.29	1.96	-	-
Length_limit	-	1024	16384	4096	1024

[1] We fine-tuned all base models on D_{Full} and reported the final loss.

Tail Section (usually *conclusion*) on summarizing well-organized scientific papers; (3) there is a slight difference of performances between different models, but no model dominates all the others. For instance, LED-base performs better than T5-base on D_{Full} and D_{Phy} while T5-base performs better than LED-base on D_{CS} and D_{Math}.

In Table 5, equivalent results can be found when using large models. Generally, given the same model, the large version obtains higher scores than the base version, showing the stronger ability of addressing this task due to the model size. Particularly, BigBird-large performs best in this part, probably because of its comparatively larger input length (4096, see Table 7) derived by the sparse attention mechanism. However, one exception is BART-large, which behaves consistently with others on D_{CS} and D_{Math} but doing best by using $P_{sal}(T)$ on D_{Full} and D_{Phy}.

For answering Q3, we focus on evaluation results on D_{CS}, D_{Math} and D_{Phy} in Table 4 and 5. We find that $P_{sal}(H+T)$ almost obtains higher scores than either $P_{sal}(H)$ or $P_{sal}(T)$ on D_{CS}, D_{Math} and D_{Phy}, no matter that which abstractive model is used. Further, it is encouraging that *SEHY* using $P_{sal}(H+T)$ paired with BigBird-large obtains the highest score (49.37 / 20.69 / 42.99) on D_{CS} (Table 5) in our experiments, showing that, comparatively speaking, the policy $P_{sal}(Sec)$ is most suitable for scientific papers in Computer Science.

Besides, we exhibit the fine-tuning time of base models on all experimental datasets in Table 8. We did not do these for the large models because they have been fine-tuned on arXiv, quoted from their original papers. It is found that training our hybrid model *SEHY*, even though leveraging simple extraction strategies, is still time-expensive because arXiv is super large-scale. The training time increases dramatically with the growth of the dataset size, especially on D_{Full}.

Table 10 shows examples of summaries generated by our models by using $P_{sal}(H+T)$, paired with the above base and large models.

Table 8: The fine-tuning time (hours) of base models on datesets.

Dataset+Policy \ Model	T5-base	LED-base	BART-base
$D_{Full}+P_{sal}(H)$	23.27	40.15	11.40
$D_{Full}+P_{sal}(T)$	22.53	21.56	11.62
$D_{Full}+P_{sal}(H+T)$	58.58	41.78	12.29
$D_{CS}+P_{sal}(H)$	1.07	1.00	0.54
$D_{CS}+P_{sal}(T)$	1.04	0.98	0.51
$D_{CS}+P_{sal}(H+T)$	2.64	2.00	0.55
$D_{Math}+P_{sal}(H)$	2.16	1.96	1.06
$D_{Math}+P_{sal}(T)$	2.10	1.96	1.05
$D_{Math}+P_{sal}(H+T)$	5.59	3.88	1.08
$D_{Phy}+P_{sal}(H)$	16.21	15.43	8.05
$D_{Phy}+P_{sal}(T)$	16.50	15.49	8.01
$D_{Phy}+P_{sal}(H+T)$	41.63	30.02	8.41

Table 9: Evaluation results of *SEHY* using $P_{inc}(k)$ paired with BigBird. Best results in each group are in bold.

Dataset+Policy \ Model	BigBird–large R-1 / R-2 / R-L
$D_{Full}+P_{inc}(1)$	35.95 / 12.01 / 30.69
$D_{Full}+P_{inc}(2)$	44.52 / 17.31 / 37.45
$D_{Full}+P_{inc}(3)$	44.73 / 17.42 / 37.45
$D_{Full}+P_{inc}(4)$	**44.80 / 17.56 / 37.48**
$D_{CS}+P_{inc}(1)$	46.31 / 19.11 / 40.84
$D_{CS}+P_{inc}(2)$	47.47 / 19.68 / 41.70
$D_{CS}+P_{inc}(3)$	48.33 / 20.52 / 42.36
$D_{CS}+P_{inc}(4)$	**48.52 / 20.67 / 42.45**
$D_{Math}+P_{inc}(1)$	43.20 / 18.03 / 37.65
$D_{Math}+P_{inc}(2)$	45.31 / 19.75 / 39.28
$D_{Math}+P_{inc}(3)$	45.59 / 19.95 / 39.02
$D_{Math}+P_{inc}(4)$	**45.71 / 19.95 / 39.27**
$D_{Phy}+P_{inc}(1)$	42.92 / 16.15 / 36.19
$D_{Phy}+P_{inc}(2)$	44.33 / 17.18 / 37.16
$D_{Phy}+P_{inc}(3)$	44.56 / 17.27 / 37.19
$D_{Phy}+P_{inc}(4)$	**44.60 / 17.40 / 37.18**

Evaluation results of $P_{inc}(k)$. We measure $P_{inc}(k)$ on D_{Full}, D_{CS}, D_{Math} and D_{Phy}. This strategy can validate the contributions of middle sections such as *methods* (Figure 1) on the generated summary. We conduct this part of experiments by only using BigBird-large because it performs best in above experiments. We set the largest value of *k* to 4 because the length limit of BigBird-large is 4096 and the average section-length on D_{Full}, D_{CS}, D_{Math} is more than 1000 (see Table 3). Evaluation results of $P_{inc}(k)$ are reported in Table 9, showing that the ROUGE scores are increased with the growth of k values (i.e., more first sections are used). However, e.g., on D_{Full}, the best result of $P_{inc}(k)$ (44.80 / 17.56 / 37.48) is much worse than that of $P_{sal}(Sec)$ (47.55 / 19.99 / 42.88).

Comparison of $P_{sal}(Sec)$ and $P_{inc}(k)$. For answering Q1, we compare $P_{sal}(Sec)$ and $P_{inc}(k)$ with regard to all experimental options. Results are shown in Figure 2, 3 and 4. Obviously, $P_{sal}(Sec)$

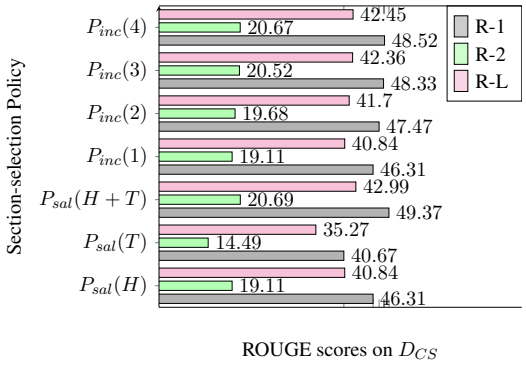

Figure 2: Comparison of section-selection strategies of *SEHY* paired with BigBird-large on the dataset D_{CS}.

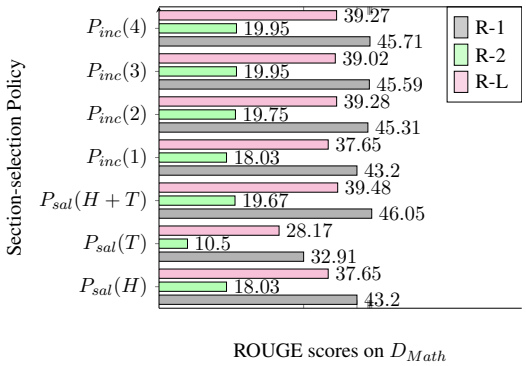

Figure 3: Comparison of section-selection strategies of *SEHY* paired with BigBird-large on the dataset D_{Math}.

performs better than $P_{inc}(k)$. Besides, from Table 4 and 5, we find that different pre-trained models do not significantly affect the performance of our approach for answering Q2.

Comparisons of *SEHY* with other approaches. We collect the best results of *SEHY* by using $P_{sal}(Sec)$ from Table 4 and 5 and compare them with those of other 12 summarization models (including 7 abstractive models, 3 extractive models and 2 hybrid models) on the full arXiv dataset D_{Full}. Evaluation results are presented in Table 6. Experimental findings are as follows: (1) even though not exceeding LongT5 (the current open-source SoTA), multiple variants of *SEHY* obtain competitive scores, i.e., the second and third highest scores on Learderboard. (2) all variants of *SEHY* except for the one paired with BART-base perform better than DANCER, which is the most related work to ours due to using section-selection strategies and training a hybrid model. (3) Apart from LongT5, *SEHY* obtains better results than the other compared models, demonstrating the effectiveness of our approach.

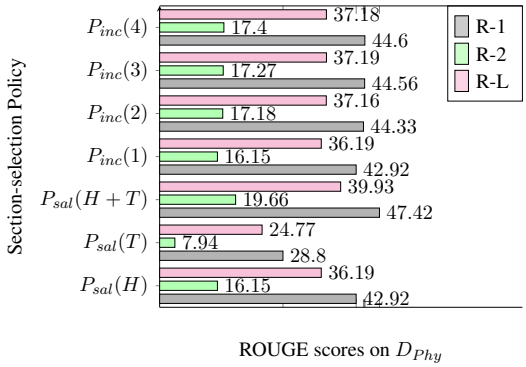

Figure 4: Comparison of section-selection strategies of *SEHY* paired with BigBird-large on the dataset D_{Phy}.

5 Conclusion

Long documents introduce a lot of noise to the summarization process and large parts of the document are not really key to its narrative and thus should be ignored. In this paper, we propose a simple yet effective hybrid model called *SEHY* for summarizing long scientific documents. In particular, we design two simple strategies for selecting sections: $P_{sal}(Sec)$ and $P_{inc}(k)$, and use strong abstractive models for generating the final summary. We conduct excessive experiments with multiple variants of *SEHY* on the full arXiv dataset publicly available and three disciplinary sub-datasets generated by ourselves. Experimental results show that $P_{sal}(Sec)$ is more effective than $P_{inc}(k)$ and our best models obtain the competitive results with regard to the current SoTA on arXiv.

Future works are as follows. First, we will evaluate the generalization of our approach on other academic-paper datasets such as ScisummNet (Yasunaga et al., 2019), which is a large-scale scientific paper summarization dataset with citation networks. Second, we will develop more flexible and sophisticated extraction strategies such as using reinforce learning (Chali and Mahmud, 2021), to further improve our approach's performance.

Acknowledgements

This paper is supported by Guangdong Basic and Applied Basic Research Foundation, China (Grant No. 2021A1515012556).

Ground-truth summary	We study a novel class of numerical integrators, the adapted nested force-gradient schemes, used within the molecular dynamics step of the hybrid monte carlo (hmc) algorithm. We test these methods in the schwinger model on the lattice, a well-known benchmark problem. We derive the analytical basis of nested force-gradient type methods and demonstrate the advantage of the proposed approach, namely reduced computational costs compared with other numerical integration schemes in hmc.
Summary generated by T5-base	We present a novel class of numerical time integration schemes for the hybrid monte carlo algorithm (hmc), often used to study quantum chromodynamics (qcd) on the lattice. The adapted nested-force-gradient approach is applied to the model problem of quantum electrodynamics (qed) in two dimensions, the schwinger model. We compare the behavior of the adapted nested-force-gradient approach and the adapted version introduced in @xcite. We investigate the computational costs needed to perform numerical calculations, as well as the effort required to achieve a satisfactory acceptance rate during the hmc evolution. We compare the behavior of the nested-force-gradient approach and the adapted version introduced in @xcite.
Summary generated by BART-base	We consider the hybrid monte carlo algorithm (hmc), often used to study quantum chromodynamics (qcd) on the lattice. We compare the behavior of numerical time integration schemes currently used for hmc with the nested force-gradient integrator and the adapted version introduced in @xcite. We investigate the computational costs needed to perform numerical calculations, as well as the effort required to achieve a satisfactory acceptance rate during the hmc evolution. Our goal is to find a numerical scheme which would provide a sufficiently high acceptance rate while not drastically increasing the simulation time.
Summary generated by LED-base	We present the nested force - gradient approach and its adapted version applied to a model problem in quantum field theory, the schwinger model. We investigate the computational costs needed to perform numerical calculations, as well as the effort required to achieve a satisfactory acceptance rate during the hmc evolution. We also compare the behavior of the adapted force-gradient scheme with the existing ones and propose a novel class of numerical schemes which would provide a sufficiently high acceptance rate while not drastically increasing the simulation time.
Summary generated by Pegasus-large	The hybrid monte carlo algorithm (hmc), often used to study quantum chromodynamics (qcd) on the lattice, has proved to be an efficient numerical time integration scheme. Omelyan integration schemes of a force - gradient type have proved to be an efficient choice, since it is easy to obtain higher order schemes that demand a small additional computational effort. These schemes use higher - order information from force-gradient terms to both increase the convergence of the method and decrease the size of the leading error coefficient.
Summary generated by BigBird-large	We present a new class of numerical time integration schemes for the hybrid monte carlo algorithm, which are based on the idea of the nested force-gradient method. We compare the behavior of these new schemes with the standard hybrid monte carlo algorithm and discuss which scheme is most suitable for the simulation of the two-dimensional lattice schwinger model. Hybrid monte carlo algorithm, numerical time integration, nested force-gradient method.
Summary generated by BART-large	In this paper we present a new class of numerical schemes for the hybrid monte carlo algorithm (hmc), often used to study quantum chromodynamics (qcd) on the lattice, which are optimal in terms of computational costs per trajectory for a given acceptance rate. We compare the behavior of numerical time integration schemes currently used for hmc with the adapted nested force-gradient integrator and the adapted version introduced in @xcite. Our goal is to find a numerical scheme for the hmc algorithm which would provide a sufficiently high acceptance rate while not drastically increasing the simulation time. We chose the model problem of quantum electrodynamics (qed) in two dimensions, the schwinger model, since it is well-suited as a test case for new concepts and ideas which can be subsequently applied to more computationally demanding problems. As a lattice quantum field theory, it has many of the properties of more sophisticated models such as qcd, for example the numerical cost is still dominated by the fermion part of the action. The fact that this model, with far fewer degrees of freedom makes it the perfect choice for testing purposes.

Table 10: Examples of summaries generated by our models by using $P_{sal}(H + T)$. For the limitation of space, the original paper is omitted.

References

Mehdi Allahyari, Seyed Amin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys J. Kochut. 2017. Text summarization techniques: A brief survey. *CoRR*, abs/1707.02268.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020a. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020b. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Tian Cai, Mengjun Shen, Huailiang Peng, Lei Jiang, and Qiong Dai. 2019. Improving transformer with sequential context representations for abstractive text summarization. In *Natural Language Processing and Chinese Computing - 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9-14, 2019, Proceedings, Part I*, volume 11838 of *Lecture Notes in Computer Science*, pages 512–524. Springer.

Yllias Chali and Asif Mahmud. 2021. Query-based summarization using reinforcement learning and transformer model. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 129–136.

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, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 615–621. Association for Computational Linguistics.

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 Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988. Association for Computational Linguistics.

Samet Demir, Uras Mutlu, and Özgür Özdemir. 2019. Neural academic paper generation. *CoRR*, abs/1912.01982.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2011. Lexrank: Graph-based lexical centrality as salience in text summarization. *CoRR*, abs/1109.2128.

Alexios Gidiotis and Grigorios Tsoumakas. 2020. A divide-and-conquer approach to the summarization of long documents. *IEEE ACM Trans. Audio Speech Lang. Process.*, 28:3029–3040.

Mandy Guo, Joshua Ainslie, David C. Uthus, Santiago Ontañón, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2021. Longt5: Efficient text-to-text transformer for long sequences. *CoRR*, abs/2112.07916.

Sheena Kurian K and Sheena Mathew. 2020. Survey of scientific document summarization techniques. *Comput. Sci.*, 21(2).

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3075–3081. AAAI Press.

Bo Pang, Erik Nijkamp, Wojciech Kryscinski, Silvio Savarese, Yingbo Zhou, and Caiming Xiong. 2022. Long document summarization with top-down and bottom-up inference. *CoRR*, abs/2203.07586.

Michal Pietruszka, Lukasz Borchmann, and Lukasz Garncarek. 2022. Sparsifying transformer models with trainable representation pooling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8616–8633. Association for Computational Linguistics.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Chris Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9308–9319. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020a. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020b. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Tobias Rohde, Xiaoxia Wu, and Yinhan Liu. 2021. Hierarchical learning for generation with long source sequences. *CoRR*, abs/2104.07545.

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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Qingyun Wang, Lifu Huang, Zhiying Jiang, Kevin Knight, Heng Ji, Mohit Bansal, and Yi Luan. 2019. Paperrobot: Incremental draft generation of scientific ideas. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1980–1991. Association for Computational Linguistics.

Qingyun Wang, Zhihao Zhou, Lifu Huang, Spencer Whitehead, Boliang Zhang, Heng Ji, and Kevin Knight. 2018. Paper abstract writing through editing mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 260–265. Association for Computational Linguistics.

Haoyang Wen, Anthony Ferritto, Heng Ji, Radu Florian, and Avi Sil. 2021. VAULT: variable unified long text representation for machine reading comprehension. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 1035–1042.

Wen Xiao and Giuseppe Carenini. 2020. Systematically exploring redundancy reduction in summarizing long documents. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020*, pages 516–528. Association for Computational Linguistics.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R. Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7386–7393. AAAI Press.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.