Generating Extractive and Abstractive Summaries in Parallel from Scientific Articles Incorporating Citing Statements

Sudipta Singha Roy
The University of Western Ontario
London, ON, Canada
ssinghar@uwo.ca

Robert E. Mercer
The University of Western Ontario
London, ON, Canada
mercer@csd.uwo.ca

Abstract

Summarization of scientific articles often overlooks insights from citing papers, focusing solely on the document's content. To incorporate citation contexts, we develop a model to summarize a scientific document using the information in the source and citing documents. It concurrently generates abstractive and extractive summaries, each enhancing the other. The extractive summarizer utilizes a blend of heterogeneous graph-based neural networks and graph attention networks, while the abstractive summarizer employs an autoregressive decoder. These modules exchange control signals through the loss function, ensuring the creation of high-quality summaries in both styles.

1 Introduction

Text summarization automates condensing documents while preserving key information. Most neural summarization models, like those by Nallapati et al. (2016); Zhong et al. (2019), are designed for shorter texts, e.g., the CNN/Daily Mail dataset (Hermann et al., 2015). However, applying these models to longer documents, such as scientific research papers, remains limited. In scientific document summarization, it is common to focus solely on abstracts, introductions, and conclusions, as demonstrated in Yasunaga et al. (2019)'s work.

Summarizing scientific publications presents unique challenges due to their length, complex concepts, technical jargon, structured organization, and citations. These complexities make it a more daunting task compared to summarizing other types of documents. Additionally, the long-term impact of a scientific article may not be fully evident when it is first published, as its significance can evolve over time. While an abstract provides an initial overview from the authors' perspective, it may not capture the full extent of the paper's influence on the research community and its evolving impact (Yasunaga et al., 2019). As an example, we can consider the abstract from Bergsma and Lin (2006):

> We present an approach to pronoun resolution based on syntactic paths. . . . we learn the likelihood of coreference between a pronoun and a candidate noun based on the path in the parse tree between the two entities. . . . Highly coreferent paths also allow mining of precise probabilistic gender/number information. We combine statistical knowledge with well known features in a Support Vector Machine pronoun resolution classifier. Significant gains in performance are observed on several datasets.

This abstract gives insight into the methods the authors used. But the citations emphasize the corpus it presents. For example:

> We use the approach of Bergsma and Lin (2006), both because it achieves state-ofthe-art gender classification performance, and because a database of the obtained noun genders is available online. (Bergsma, 2005)

> For the gender task that we study in our experiments, we acquire class instances by filtering the dataset of nouns and their genders created by Bergsma and Lin (2006). (Bergsma and Van Durme, 2013)

Jaidka et al. (Jaidka et al., 2016, 2019) have identified this missing aspect in scientific document summarization and addressed it by introducing a shared task. This task aims to create summaries that take into account not only the information in the body of the documents but also the research community's overview of the documents over time. The work described here continues in this direction.

With the advancement of neural networks, there have been a few prominent research works in recent years for generating extractive (Yasunaga et al., 2019) and abstractive (Yu et al., 2020; Zhang et al.,

Proceedings of The 4th New Frontiers in Summarization Workshop, pages 75–86

2019) summaries from scientific documents (Cohan et al., 2018; Zhang et al., 2022). Extractive summarization recognizes key sentences from the source document as the summary but lack the flow of information, whereas the abstractive summarization technique generates new phrases using language models while preserving the semantics of the input document but may miss some important aspects of the text. This is a motivation for designing a model to generate both summaries in parallel and help the counterpart to achieve a performance boost with additional guidance.

A key step in extracting brief synopsis sentences from a manuscript is to map the cross-sentence correlations. A lot of recent prominent works (Nallapati et al., 2017) have tried to do so using recurrent neural networks (RNNs). However, because of using RNNs, these models fail to capture long-distance sentence-level dependencies. Another approach to preserve sentence-level dependencies from long documents is using graph-based neural networks. A few recent works (e.g., (Cohan and Goharian, 2018; Yasunaga et al., 2017)) have utilized discourse information in the article along with inter-sentence correlations for constructing graphs and summarizing document. Another approach is to construct a sentence-level fully connected graph. Zhong et al. (Zhong et al., 2019) and Liu et al. (Liu and Lapata, 2019) used transformer (Vaswani et al., 2017) encoders to determine how sentences interact with each other. Wang et al. (Wang et al., 2020) introduced an heterogeneous graph neural network for extractive summarization which used additional semantic units (words) as intermediate nodes to construct relationships between sentences.

Abstractive summarizers focus heavily on form, with the goal of producing a generalized summary, which tends to necessitate complex language-generating models. These models are typically based on sequence-to-sequence (seq2seq) architectures, in which a source document is seen as one sequence whereas its summary as another. The majority of previous research on neural abstractive summarization depended on large-scale, high-quality datasets of supervised document-summarization pairings (See et al., 2017). Recently, state-of-the-art solutions on abstractive summarization are built upon the transformer (Vaswani et al., 2017) and BERT (Devlin et al., 2019) models. These attention-based abstractive models are being used in different fields like clinical note summarization (Kanwal and Rizzo, 2022), scientific document summarization (Zhang et al., 2022), and lay-abstract generation (Yu et al., 2020).

In this paper, addressing the above-mentioned issues, we have built a standalone summarization model which can generate both extractive and abstractive summaries from scientific documents incorporating the citation network. Analyzing the citation network, citing statements from the citing articles are accumulated with the original text document to incorporate the research community's observation on that particular cited manuscript. These summaries are the abstracts of the original papers with additional information reflecting the research community's view. After that, we run the Longformer (Beltagy et al., 2020) encoder to generate sentence and word representations and train extractive and abstractive summarizers together. For the extractive summarizer, an heterogeneous graph neural network (Wang et al., 2020) is used as it has the ability to preserve sentence-level dependencies utilizing additional semantic units as intermediate nodes in the graph representation. Abstractive summaries are generated by the autoregressive decoder. The loss function is defined in such a way that both summarizers can achieve better ROUGE and METEOR scores. Furthermore, we have developed a corpus containing 10K research articles along with their corresponding citation statements and is a subset of the Semantic Scholar Network (SSN) corpus. The citation statements are collected utilizing the citation graph used in the SSN corpus. In short, the contributions of this work are:

- We have built a stand-alone summarizer model which can produce both extractive and abstractive summaries and each counterpart helps the other to generate better summaries.
- The summarizer model can work with long scientific text articles
- This model considers research communities' observations while generating the summaries
- We have proposed a new corpus containing 10K research articles along with the corresponding citing statements to incorporate the research communities' view.

2 Related Work

Text summarization aims to distill a document's essence efficiently. Recent NLP research has

yielded effective neural summarization models, particularly those using transformer and BERT-based architectures. Work summarizing lengthy scientific documents often focuses on specific sections rather than the entire text (Yu et al., 2020) or citation statements (An et al., 2021).

2.1 Extractive Text Summarization

Extractive text summarization models classify sentences in a document using labels that indicate whether or not a sentence ought to be included in the summary. Originally, these models were designed based on the encoder-decoder architecture using RNNs (Nallapati et al., 2017). Since transformer and BERT-based models provide a more enriched sentence encoding, they have become the foundation for the majority of extractive summarizer models in recent years. Liu and Lapata (2019) fine-tuned BERT with stacked layers of transformer to obtain the sentence vectors and then used a sigmoid classifier for identifying the sentences that would be included in the summary. Zhang et al. (2019) fine-tuned an hierarchical transformer (HIBERT) for the extractive summarization task. Another prominent approach for extractive summarization is using graph representations which can preserve sentence-level correlations. Later, the graph convolutional network (GCN) (Welling and Kipf, 2016) has been espoused for building different inter-sentence correlation graphs (Yasunaga et al., 2017) for this task. Wang et al. (2020) built an heterogeneous graph neural network for extractive summarization (HeterSumGraph) which takes into account additional semantic units at the word level for building the sentence-level correlation graph.

2.2 Abstractive Text Summarization

Abstractive text summarization models, unlike the extractive summarizers which work like classifiers, are intended to generate summaries comprising new sentences which may or may not be present in the body of the document. These models are mostly based on the encoder-decoder architecture of the sequence-to-sequence models and language models like BART (Lewis et al., 2020), BigBird (Zaheer et al., 2020), and T5 (Raffel et al., 2020). Aksenov et al. (2020) applied BERT-windowing to overcome the length limitation of the BERT model and summarize long documents. Gidiotis and Tsoumakas (2020) trained the summarizer model to generate separate abstractive summaries for small parts of the document. Pilault et al. (2020) combined both the extractive and abstractive summarization using a transformer language model and built an hybrid summarizer model. Yu et al. (2020) fine-tuned pretrained BERT as the abstractive summarizer for generating a lay summary from the document.

2.3 Scientific Article Summarization

Existing scientific article summarizers, in most cases, are extractive models designed on the idea of sentence selection (Cohan and Goharian, 2018). Cohan et al. (2018) developed the first abstractive summarizer for long scientific articles using an hierarchical encoder and discourse-aware attentive decoder. Mishra et al. (2022) applied citation contextualization to extract unique relevant sentences from the document and final summaries are generated using a multi-objective clustering approach. Gupta et al. (2022) applied BERT and graph-based approaches for biomedical document summarization. Li et al. (2020) fine tuned T5 for generating summaries from long scientific documents and implemented an extractive summarizer using GCN. Yasunaga et al. (2019) built a corpus (ScisummNet) that includes a citation network for scientific document summarization and extracted the summary-candidate sentences using a GCN. An et al. (2021) introduced a large corpus (SSN) with 141K research papers connected with a citation graph. They also proposed a graph-based summarization model (CGSUM) for extractive document summarization. This model can draw information from both the source and the citing texts.

3 Methodology

This section defines the problem of scientific document summarization using a citation graph. Then, the two benchmark datasets used for the scientific article summarization experiments are discussed along with the pre-processing procedures. Finally, the proposed deep learning model is explained.

3.1 Problem Formulation: Summarization Using Citation Graph

Scientific articles possess distinctive attributes, including citation linkages, that establish profound connections between their contents. These studies may also yield unforeseen impacts and evolve in importance as research progresses. In such cases, ideal summaries should encompass both the authors' key points and the perspectives of the scientific community, as reflected in cita-

tions (Yasunaga et al., 2019). To serve this intent we have utilized two resources: the citation graph provided in the Semantic Scholar Network (SSN) corpus (An et al., 2021), and the ScisummNet/CL-SciSumm-2020 (CL-SciSumm-2020) corpus (Chandrasekaran et al., 2020; Yasunaga et al., 2019) which supplies documents and their corresponding citing statements.

3.2 Description of the Datasets

As this work is focused on generating summaries from scientific articles that incorporate the research community's views, we have considered two benchmark datasets: ScisummNet/CL-SciSumm (Chandrasekaran et al., 2020; Yasunaga et al., 2019), and Semantic Scholar Network (SSN) (An et al., 2021) for the experiments done here. To the best of our knowledge, these are the only datasets for the summarization task that also provide citation information. The ScisummNet corpus consists of abstracts of the 1000 most cited research articles from the ACL Anthology Network (Radev et al., 2013) along with 15 citing statements per article. The gold standard summaries for these 1000 documents are manually summarized by domain experts. The CL-SciSumm-2020 corpus (Chandrasekaran et al., 2020) extends the ScisummNet corpus with 40 extra documents and human-generated summaries thereby providing 1040 documents, citation sentences, and summaries. For testing, we have used the test set comprising 200 scientific articles from the CL-SciSumm-2020 corpus. The other benchmark dataset used for this task is the SSN corpus. It includes 140,799 research articles culled from the Semantic Scholar Open Research Corpus (S20RC) (Lo et al., 2020) together with a large citation graph. This citation graph has each article as a node and 660,908 edges indicating the citations. This corpus covers research articles from three domains: physics, mathematics and computer science.

The primary objective of this study is to develop a deep learning model capable of generating summaries for lengthy scientific documents while incorporating insights from other researchers citing the document. While the ScisummNet/CL-SciSumm dataset provides citation statements, the SSN corpus lacks this information. Originally, the SSN corpus consisted of documents and their references, but for our purpose of including citing statements, modifications were necessary. We leveraged the citation graph to identify citing papers and manually extracted the statements referring to the cited articles. Given the substantial size of the SSN corpus, containing nearly 141K articles, we randomly selected 10K papers for summarization. These papers have body lengths ranging from 1000 to 3500 words (with background/related work sections removed), aligning with the capacity of the Long-Former model (as described in Section 4), which can handle a maximum of 4096 tokens at a time. The dataset was divided into training (8000), validation (1000), and testing (1000) articles to facilitate model development and evaluation.

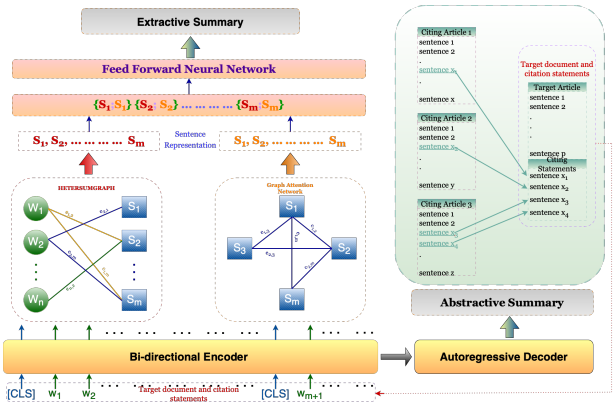

Figure 1: System architecture of the proposed model

Citations can convey positive, neutral, or negative intentions. To capture this diversity, we systematically categorized citing statements into these three classes after gathering them from citing articles. In cases where a paper had limited negative citations, we balanced the selection by including more neutral and positive citation statements. To classify these citation statements, we have employed RoBERTa trained on Athar (2014) following the approach used by Kundu (2023).

In the SSN corpus, the summaries are limited to the authors' perspectives as they consist of the paper abstracts. To create more comprehensive summaries, we employed a two-step approach. First, we used a fine-tuned T5 model (Raffel et al., 2020), trained on the CL-SciSumm-2020 corpus, to generate five summaries per document by inputting both the abstracts and corresponding citation statements. Then, we have employed a pre-trained RoBERTa architecture to obtain five vector representations for these summaries. The most similar summary to the reference summary, determined by cosine similarity, was selected as our T5-Generated Summary.

3.3 Model Overview

The investigated summarization model has two units: an extractive and an abstractive summarizer. The overall architecture of the model is portrayed in Figure 1. This section discusses the architecture and working principle of these two units.

While designing the extractive summarizer, we have considered two issues: how the sentences are connected to each other and how semantic units like words affect the sentence level correlations. To fulfill these purposes, we have utilized two different graph-based neural networks: an heterogeneous graph neural network (HeterSumGraph) (Wang et al., 2020) and a graph attention network (GAT) (Velickovic et al., 2018).

For any graph $G = \{V, E\}$, V denotes the nodes and E, the edges between them. HeterSumGraph defines $V = V_w \cup V_s$, V_w is the set of unique words and V_s is the set of sentences in the document. For a document with n unique words and m sentences, E is the edge weight matrix, where $e_{i,j}$ represents word i in sentence j, $(i \in \{1:n\}, j \in \{1:m\})$ (Wang et al., 2020). The nodes that represent the sentences are initialized with LongFormer [CLS] tokens. Because LogFormer generates a contextualized word embedding for each occurrence of the word in the document, all of the word embeddings for a word are averaged to initialize that particular word-representing node in the graph. The edges between the words and sentences are initialized with the corresponding TF-IDF values.

After the graph G is constructed, a graph attention network (GAT) is used to update the node feature values. Considering $h_i \in \mathbb{R}^{d_h}$ where $i \in \{1:(n+m)\}$ as the hidden states of the word and sentence nodes, the GAT layer is designed as:

$$\mathcal{T}_{i,j} = \textit{LeakyReLU}(\omega_a[\omega_q h_i; \omega_k h_j; e_{i,j}]) \quad (1)$$

$$\alpha_{i,j} = \frac{\exp(\mathcal{T}_{i,j})}{\sum_{l \in \mathcal{N}_i} \exp(\mathcal{T}_{i,l})} \quad (2)$$

$$u_i = \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{i,j} \omega_v h_j) \quad (3)$$

where ω_a, ω_q, ω_k and ω_v are learnable weight matrices. $\mathcal{N}_i$ denotes the list of the neighbor nodes. The attention value between h_i and h_j is denoted by $\alpha_{i,j}$. The GAT with multi-head attention (considering $\mathcal{K}$ attention heads) is designed as:

$$u_i = ||_{k=1}^{\mathcal{K}} \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{i,j}^k \omega^k h_i) \quad (4)$$

To prevent the gradient from vanishing, HeterSumGraph incorporates a residual connection and the final hidden state representation becomes:

$$h_i = u_i + h_i \quad (5)$$

Through the aforementioned GAT and position-wise feed-forward network (FFN) layer comprising two linear transformations (Wang et al., 2020), the sentence nodes are updated with their adjacent word nodes:

$$\mathcal{U}_{w \to s}^1 = \textit{GAT}(\mathcal{H}_s^0, \mathcal{H}_w^0, \mathcal{H}_w^0) \quad (6)$$

$$\mathcal{H}_s^1 = \textit{FFN}(\mathcal{U}_{w \to s}^1 + \mathcal{H}_s^0) \quad (7)$$

where $\mathcal{U}_{w \to s}^1 \in \mathbb{R}^{n \times d_h}$ $\mathcal{H}_w^1 = \mathcal{H}_w^0 = V_w$, and $\mathcal{H}_s^0 = V_s$. In Eq. 6, $\mathcal{H}_s^0$ is employed as the attention query and for both the attention key and value $\mathcal{H}_w^0$ is used. Then, the revised sentence nodes are used to generate new representations for the individual word nodes and continue to refine the revised sentence nodes in an iterative fashion. At each iteration, sentence-to-word and word-to-sentence updates continue to be processed. The process can be depicted as follows for the t-th iteration:

$$\mathcal{U}_{s \to w}^{t+1} = \textit{GAT}(\mathcal{H}_w^t, \mathcal{H}_s^t, \mathcal{H}_s^t) \quad (8)$$

$$\mathcal{H}_w^{t+1} = \textit{FFN}(\mathcal{U}_{s \to w}^{t+1} + \mathcal{H}_w^t) \quad (9)$$

$$\mathcal{U}_{w \to s}^{t+1} = \textit{GAT}(\mathcal{H}_s^t, \mathcal{H}_w^{t+1}, \mathcal{H}_w^{t+1}) \quad (10)$$

$$\mathcal{H}_s^{t+1} = \textit{FFN}(\mathcal{U}_{w \to s}^{t+1} + \mathcal{H}_s^t) \quad (11)$$

Once the model training is done, the sentence nodes' representations are used as the sentence vector representations.

For direct sentence-level interactions, we have also used a graph attention neural network (GAT). Here, for the graph $G = \{V, E\}$, $V = V_s$ where V_s is the set of all the sentences in the document. The edge weight matrix E preserves the semantic similarity values between sentences. The nodes are initialized in the same manner as the sentence nodes in HeterSumGraph. For initializing the edges between nodes, at first we have acquired the vector representations of the sentences using pre-trained LongFormer and then computed the cosine similarity between the sentences. The edges are initialized with the corresponding similarity values between sentences. However, as scientific documents come with many sentences, working with a fully connected graph is not computationally cost effective. To reduce the burden of computational overhead, we have dropped the edge connections between

nodes whose cosine similarity values are below a certain cut-off value. Throughout the conducted experiments, we have found that if we set the cut-off value below 0.3, the performance of the summarizer model remains the same.

Considering node features $h = \{h_1, h_n, ..., h_m\}$ as the input, GAT applies a self attention on the nodes and computes the attention coefficients as follows:

$$\mathcal{T}_{i,j} = a(\omega h_i, \omega h_j) \tag{12}$$

where a is a single-layer feed forward neural network with the *LeakyReLU* activation function, and ω is a learnable parameter. This attention coefficient shows node j's importance on node i and it is computed only for the corresponding one-hop neighbour nodes ($j \in \mathcal{N}_i$). This attention coefficient value is normalized to compute the attention values as follows:

$$\alpha_{i,j} = \frac{\exp(\mathcal{T}_{i,j})}{\sum_{l \in \mathcal{N}_i} \exp(\mathcal{T}_{i,l})} \tag{13}$$

The multi-head attention is computed in the same way it has been done for HeterSumGraph (Eq. 4).

Once the sentence representations from both the HeterSumGraph and GAT are computed, they are concatenated and fed to the feed-forward neural network layer. This is a two-layer position-wise feed-forward layer (Wang et al., 2020) for labeling the sentences with 1 or 0; 1 indicates that particular sentence is included in the extractive summary.

The abstractive summary is generated by the LongFormer decoder. To train the summarizer units in parallel, the training mechanism in Yu et al. (2020) is used. The overall loss L of the model is:

$$L = L_{ext} + L_{abs} \tag{14}$$

where L_{ext} and L_{abs} represent the cross-entropy losses of the extractive and abstractive summarizers, respectively.

4 Experimental Results and Analysis

This section gives a brief description of the model parameters used in the experiments as well as the results achieved on CL-SciSumm-2020 and the customized SSN datasets.

4.1 Model Parameters and Training Details

We have trained our model on a 48GB NVIDIA RTX A6000 GPU. The batch size has been set to 1 as the length of input documents plus the citation statements is large. Since all the experiments are done on a small batch-size, we have followed the training procedure of Sefid and Giles (2022) and accumulated gradients for 10 steps and updated the parameters. The NOAM scheduler has been utilized to adjust the learning rate and gradients are clipped so that exploding gradients during training can be prevented. The model has been trained for 20,000 epochs. The extractive summarizer is initialized with the LongFormer embeddings. Following that, the LongFormer encoder-decoder architecture for the abstractive summarizer and the extractive summarizer units' forward passes are trained separately. Once both of the forward passes are done for each iteration and the individual losses are calculated, the model's overall loss is calculated. If either of the two unit's validation loss continues to go down for 5 epochs, the parameter settings for that particular unit are saved and that unit's training is postponed for the next 10 epochs. The number of attention-heads for multi-head attention has been set to 8. The stop words and punctuation have been filtered out when pre-processing the word nodes in the graph. Following Wang et al. (2020), 10% of the words in the vocabulary having low TF-IDF values have been further filtered out. The word and sentence nodes have been initialized with 768-dimensional vectors. And the sentence representations from both the HeterSumGraph and GAT are 512-dimensional vectors. So, the final sentence vectors after the concatenation step are 1024-dimensional vectors. The Feed Forward Network hidden layer size is 512.

4.2 Performance Analysis of the Model

We have performed experiments on two datasets: modified SSN and CL-SciSumm-2020. The results achieved by our models are reported as overlapping unigrams, bigrams, and the longest common sequence between the generated summaries and the reference summaries by means of R-1, R-2, and R-L metrics; and semantic compatibility between the reference and generated summaries by means of METEOR metric, respectively, for the modified SSN corpus. R-1, and R-2 show the informativeness, and R-L shows the fluency of the generated summary. The metrics used for analyzing the model performance on CL-SciSumm-2020 are R-2 and R-SU4, which indicate the proportion of bigram overlap and unigram plus skipgram of

Table 1: Results on the modified SSN corpus. The results consider both the abstracts and the T5-generated summaries incorporating citation statements as the reference summaries. The best results are boldfaced.

Models	On Abstracts as Summaries				On T5-Generated Summaries			
	R-1	R-2	R-L	METEOR	R-1	R-2	R-L	METEOR
Extractive								
BERTSumExt	42.92	14.19	39.01	33.09	43.11	14.21	39.12	33.07
HeterSumGraph	44.27	14.52	39.73	33.18	44.30	14.53	39.74	33.18
GRETEL	**45.22**	**15.19**	**40.23**	**36.87**	**45.23**	15.19	**40.24**	**36.88**
Proposed Model (Extractive)	45.19	15.18	40.21	36.83	45.19	**15.21**	40.23	36.85
Abstractive								
PTGen+Cov	41.66	13.08	36.95	32.44	41.60	13.10	36.72	32.40
BERTSumAbs	42.06	14.52	38.17	32.49	42.04	14.56	38.17	32.49
BERT+CopyTransformer	42.43	15.01	39.03	32.88	42.44	15.05	39.04	32.91
Proposed Model (Abstractive)	**44.82**	**15.19**	**39.31**	**36.50**	**44.83**	**15.19**	**39.30**	**36.51**

4 tokens overlap, respectively, between the reference and generated summaries. The performance here is also analyzed with the METEOR metric. As the Bi-directional encoder and autoregressive decoder we have also experimented with BigBird. However, the better performance was found with LongFormer. That is why in the final model, we have used LongFormer in all the cases for initial encoding and generating abstractive summaries.

4.3 Results: Modified SSN Corpus

To compare the performance of our model with the existing extractive models, we train and test the following extractive summarizer models on our modified corpus: (1) BERTSumEXT (Liu and Lapata, 2019): a BERT-based model; (2) HeterSumGraph (Wang et al., 2020): a heterogeneous graph-based approach that considers the cross-sentence correlations using additional semantic units; and (3) GRETEL: fuses semantic information from the document context and gold summary using a hierarchical transformer encoder and graph contrastive learning. For the abstractive summarization baseline, we have experimented with: (1) PTGen+Cov (See et al., 2017): based on a hybrid pointer generator network to copy words from the source text, (2) BERTSumAbs (Liu and Lapata, 2019): a BERT-based model; and (3) BERT+CopyTransformer (Aksenov et al., 2020): applies BERT-windowing for processing data longer than the BERT window.

The performance of the existing models and our proposed models are shown in Table 1. As reference summaries, we have considered both the paper abstracts as well as the summaries we have generated from the abstracts plus the citing statements using T5.

Although BERTSumExt and BERTSumAbs perform very well with short documents, their performance metrics are not at that level when summarizing scientific documents. The main reason for this is their limitation to working with a maximum 512 input tokens, but scientific documents are much longer. For this, they have applied the greedy algorithm introduced by Nallapati et al. (2016). HeterSumGraph considers direct relationships between words and sentences on texts with a 50-sentence maximum, whereas our proposed model considers direct cross-sentence correlations, as well, and can deal with longer text spans (up to 3500 words). These additional features, together with LongFormer's enriched word and sentence features, gives our model a performance boost, but our model requires more computational time and resources. Our model performs better by a good margin compared to the other models apart from GRETEL. Our extractive summarizer shows slightly lower performance compared to GRETEL which is a more complex model. Still, because of the parallel training approach, our model has achieved comparable results. Our abstractive summarizer model outperforms the other experimental abstractive summarizers by large margins: PTGen+Cov by 2.36, BertSumAbs by 1.14, and BERT+CopyTransformer by 0.28 R-L scores. The METEOR scores achieved by our model are 36.83 and 36.50 for extractive and abstractive summaries, respectively, when tested over the T5-generated summaries. In the experiment with the abstracts as summaries, the METEOR scores are 36.51 and 36.85 for the abstractive and extractive summaries, respectively. Looking at the METEOR scores achieved by the other models (see Table 1), it is clearly visible that both the extractive and abstractive summarizer units of our model have outper-

Table 2: Model performance analysis on two CL-SciSumm-2020 summary categories. All values are F-1 scores.

Models	Abstracts as Summaries			Human-created Summaries		
	R-2	R-SU4	METEOR	R-2	R-SU4	METEOR
Jaccard-focused GCN	0.19931	0.09956	-	0.2042	0.14162	-
Clustering	0.1959	0.0962	-	0.1749	0.1169	-
MMR2	0.15067	0.07851	-	0.15073	0.10237	-
LSTM+BabelNet	0.329	0.172	-	0.241	0.171	-
Proposed Model						
Extractive Summarizer	0.43	0.266	31.12	0.42	0.249	30.18
Abstractive Summarizer	0.43	0.250	30.98	0.41	0.234	30.06

formed them by at least 3. This observation indicates that the summaries generated by our proposed model are more semantically similar to the reference summaries. To see the importance of the individual units, please check the ablation study in the appendix.

4.4 Results: CL-SciSumm-2020 Corpus

For analyzing our proposed model's performances on CL-SciSumm-2020 Corpus, we have used R-2 and R-SU4 F-1 scores (as the other comparable models are reported with these metrics) We have experimented to generate abstract and human summaries. As benchmarks, we have selected the research works submitted to CL-SciSumm-2019/2020: (1) Jaccard-focused GCN (Umapathy et al., 2020): an extractive summarizer utilizing cross-sentence graph and graph attention networks, (2) Clustering (Mishra et al., 2020): based on different clustering algorithms followed by sentence-scoring functions, (3) MMR2 (Reddy et al., 2020): based on the maximal marginal relevance technique, and (4) LSTM+BabelNet (Chiruzzo et al., 2019): BabelNet vectors were used to train the LSTM. The CL-SciSumm task provides a performance metric evaluation script which is used to calculate the R-2 and R-SU4 values for the model-generated summaries against the test set.

Results on CL-SciSumm-2020 are reported in Table 2. Looking at the results, it is clear that our model outperforms the other existing extractive models on every measure. The R-2 and R-SU4 achieved for both of our model-generated extractive and abstractive summaries are very high compared to the other existing extractive models. And this is the case for both the original abstracts and the human-created summaries as reference summaries. For the human-created reference summaries, our extractive and abstractive summarizers have achieved 0.078 and 0.063 R-SU4 F-1 score gains, respectively, compared to the LSTM+BabelNet model, which comes with the best result among the other considered models. While considering the abstracts of the papers as reference summaries, these gains are 0.094 and 0.078, respectively. For the abstractive summaries, the METEOR score achieved by our model is 30.18 whereas for the extractive summaries, it has achieved a 30.06 METEOR score on the human-generated summaries. Over the abstracts of the papers, these scores are 31.12 and 30.98, respectively.

5 Conclusion and Future Work

In this paper, we have introduced a summarizer model considering two intentions: first, summarize scientific documents incorporating citation contexts, and second, build a summarizer model which can generate both extractive and abstractive summaries by means of parallel training so that both counterparts can gain a performance boost. For this, we have utilized both the sentence-sentence and sentence-word correlations. Furthermore, we have constructed a corpus comprising 10K scientific articles with their corresponding citation statements for the summarization task. The experimental results show that our model performs well compared to other well-known methods. Though this work considers the research community's observations (citing statements), it doesn't consider the background information (references presented in the target article). In our future work, we are planning to use both sides of the citation graph (references as the background knowledge and the citing statements as the research community's views) while summarizing a scientific article.

Acknowledgements

We would like to thank all reviewers for their comments. This research is partially funded by The Natural Sciences and Engineering Research Council of Canada (NSERC) through a Discovery Grant to R. E. Mercer.

Limitations

Our experiments are limited to summarize long scientific texts only. We have not conducted any experiments with short target texts, consequently we are not sure how well the model may perform while summarizing short texts. We are also unsure how well this model may perform for extreme summary generation like TLDR (Cachola et al., 2020). Moreover, we have trained both the extractive and abstractive summarizer units for a large number of epochs. Though to prevent any unit from being over-fitted we have checked the curve of validation loss after every 5 epochs. This is very computationally expensive and demands a longer period of time for model training. Furthermore, no tests have been performed to see how the abstractive summarizer unit suffers from hallucination.

Ethics Statement

We do not see any ethics issue here in this paper.

References

Dmitrii Aksenov, Julian Moreno Schneider, Peter Bourgonje, Robert Schwarzenberg, Leonhard Hennig, and Georg Rehm. 2020. Abstractive text summarization based on language model conditioning and locality modeling. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6680–6689.

Chenxin An, Ming Zhong, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2021. Enhancing scientific papers summarization with citation graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12498–12506.

Awais Athar. 2014. Sentiment analysis of scientific citations. university of cambridge. *Computer Laboratory, Stroudsburg, PA, USA*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Shane Bergsma. 2005. Automatic acquisition of gender information for anaphora resolution. In *Proceedings of the 18th Conference of the Canadian Society for Computational Studies of Intelligence*, pages 342–353.

Shane Bergsma and Dekang Lin. 2006. Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 33–40.

Shane Bergsma and Benjamin Van Durme. 2013. Using conceptual class attributes to characterize social media users. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 710–720.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel S Weld. 2020. Tldr: Extreme summarization of scientific documents. *arXiv preprint arXiv:2004.15011*.

Muthu Kumar Chandrasekaran, Guy Feigenblat, Eduard Hovy, Abhilasha Ravichander, Michal Shmueli-Scheuer, and Anita De Waard. 2020. Overview and insights from scientific document summarization shared tasks 2020: CL-SciSumm, LaySumm and LongSumm. In *Proceedings of the First Workshop on Scholarly Document Processing (SDP 2020)*.

Luis Chiruzzo, Àlex Bravo, Horacio Saggion, et al. 2019. Lastus-taln+ inco@ cl-scisumm 2019. In *BIRNDL@ SIGIR*.

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

Arman Cohan and Nazli Goharian. 2018. Scientific document summarization via citation contextualization and scientific discourse. *International Journal on Digital Libraries*, 19(2):287–303.

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 Volume 1 (Long and Short Papers)*, pages 4171–4186.

Alexios Gidiotis and Grigorios Tsoumakas. 2020. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:3029–3040.

Supriya Gupta, Aakanksha Sharaff, and Naresh Kumar Nagwani. 2022. Biomedical text summarization: a graph-based ranking approach. In *Applied Information Processing Systems*, pages 147–156. Springer.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. 2016. Overview of the CL-SciSumm 2016 shared task. In *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*, pages 93–102.

Kokil Jaidka, Michihiro Yasunaga, Muthu Kumar Chandrasekaran, Dragomir Radev, and Min-Yen Kan. 2019. The CL-SciSumm shared task 2018: Results and key insights. In *CEUR Proceedings*, volume 2132.

Neel Kanwal and Giuseppe Rizzo. 2022. Attention-based clinical note summarization. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 813–820.

Souvik Kundu. 2023. Citation polarity identification from scientific articles using deep learning methods. Master's thesis, The University of Western Ontario.

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*, pages 7871–7880.

Lei Li, Yang Xie, Wei Liu, Yinan Liu, Yafei Jiang, Siya Qi, and Xingyuan Li. 2020. Cist@ cl-scisumm 2020, longsumm 2020: Automatic scientific document summarization. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 225–234.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S Weld. 2020. S2orc: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983.

Santosh Kumar Mishra, Harshavardhan Kundarapu, Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. 2020. Iitp-ai-nlp-ml@ cl-scisumm 2020, cl-laysumm 2020, longsumm 2020. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 270–276.

Santosh Kumar Mishra, Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. 2022. Scientific document summarization in multi-objective clustering framework. *Applied Intelligence*, 52(2):1520–1543.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first AAAI conference on artificial intelligence*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Christopher 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)*, pages 9308–9319.

Dragomir R Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. 2013. The ACL anthology network corpus. *Language Resources and Evaluation*, 47(4):919–944.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Saichethan Reddy, Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. 2020. Iiitbh-iitp@ cl-scisumm20, cl-laysumm20, longsumm20. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 242–250.

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

Athar Sefid and C Lee Giles. 2022. Scibertsum: Extractive summarization for scientific documents. In *International Workshop on Document Analysis Systems*, pages 688–701. Springer.

Anjana Umapathy, Karthik Radhakrishnan, Kinjal Jain, and Rahul Singh. 2020. Citeqa@ clscisumm 2020. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 297–302.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations, ICLR*.

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

Max Welling and Thomas N Kipf. 2016. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7386–7393.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462.

Tiezheng Yu, Dan Su, Wenliang Dai, and Pascale Fung. 2020. Dimsum@ laysumm 20: Bart-based approach for scientific document summarization. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 303–309.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297.

Haopeng Zhang, Semih Yavuz, Wojciech Kryscinski, Kazuma Hashimoto, and Yingbo Zhou. 2022. Improving the faithfulness of abstractive summarization via entity coverage control. *arXiv preprint arXiv:2207.02263*.

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

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. *arXiv preprint arXiv:1907.03491*.

A Appendix

A.1 Ablation Study

Table 3: Ablation Study: Rows labeled with † indicate the extractive summaries and rows labeled with ∗ indicate abstractive summaries.

Discarded Unit	On T5-Generated Summaries			
	R-1	R-2	R-L	METEOR
GAT†	44.86	14.9	39.96	36.52
HeterSumGraph†	44.78	14.81	39.84.	36.49
Extractive Summarizer∗	43.01	15.02	38.99	35.92
Abstractive Summarizer†	44.91	14.95	39.96	36.50

To portray a better grasp of each component's contribution in our suggested model, we have experimented with different units of our model separately and the results are reported in Table 3. All of these experiments are performed on the T5-generated corpus which combines the abstract of the paper along with the citation statements.

In our first experiment, we have discarded the GAT unit which works with cross-sentence relationships and kept only the HeterSumGraph for extractive summary generation. This time the performances of the model are lower than the reported results in Tables 1 (R-1: 44.86, R-2: 14.91, R-L: 39.96, and METEOR: 36.52) for our generated extractive summaries. Still, these results are higher compared to the original HeterSumGraph model. It shows, using the LongFormer encoder in the beginning and using the collective loss function for both the abstractive and extractive summarizer units play a significant role in the performance boost. And it also indicates that taking direct cross-sentence correlations into consideration provides some additional features to enrich the model which helps the model's performance to improve.

In the second experiment, we have discarded the HeterSumGraph unit and used only GAT in the extractive summarization unit. This time the performance metrics for extractive summaries are R-1: 44.78, R-2: 14.81, R-L: 39.84,and METEOR: 36.4. These values are comparably lower than we gained in the last experiment. The reason behind this incident is, though no direct cross-sentence relationships are present, HeterSumGraph, by 2-hop distance, considers the correlations between sentences.

The third experiment discards the extractive summarizer unit. The LongFormer abstractive summarizer unit achieves very poor R-1: 43.01, R-2: 15.02, R-L: 38.99, and METEOR: 35.92 scores compared to the proposed model. This poor performance demonstrates the importance of the information that the extractive summarizer provides the abstractive summarizer through the combined loss function.

Finally, we have discarded the abstract summarizer unit and used the combination of HeterSumGraph and GAT for extractive summary generation. During this experiment, the achieved R-1, R-2 R-L, and METEOR scores are 44.91, 14.95, 39.96, and 36.50, respectively, which are more than the cases for the three above-mentioned ablation experiments. It indicates the significance of training the abstractive summarization unit in parallel as well as using the cross-sentence and semantic unit-sentence correlations at the same time.

A.2 Validity Check of the Proposed Corpus

To ascertain the corpus's quality, a rigorous analysis was conducted on a statistically significant subset of the dataset, with a confidence level of 95% and a margin of error of 3%, aided by three human annotators. Within the vast pool of 10,000 summarization samples, a random selection of 400 was subject to annotation for this statistical inquiry.

Each annotator was tasked with evaluating whether the summaries generated by the T5 model effectively encapsulated the same information as the combination of the abstract and the citing statements. The first annotator affirmed that 374 samples achieved this concurrence, the second annotator concurred with 368, and the third annotator with 371.

When comparing the assessments of the first and second annotators, it was determined they agreed that 368 samples were appropriately summarized, while 16 were not, resulting in a substantial Cohen's κ of 0.89. In the comparison between the second and third annotators, a significant concurrence emerged for 396 samples, where 368 were accurately summarized, and 28 were not, yielding κ value of 0.93. Similarly, when examining the assessments of the first and third annotators, agreement was established for 398 summaries, with 370 being correctly summarized and 27 not, resulting in κ of 0.94.