IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 May 2023, accepted 16 June 2023, date of publication 4 July 2023, date of current version 30 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3292300

RESEARCH ARTICLE

Generation of Highlights From Research Papers Using Pointer-Generator Networks and SciBERT Embeddings

TOHIDA REHMAN[1], DEBARSHI KUMAR SANYAL[2], SAMIRAN CHATTOPADHYAY[1,3], PLABAN KUMAR BHOWMICK[4], AND PARTHA PRATIM DAS[5,6], (Member, IEEE)

[1]Department of Information Technology, Jadavpur University, Salt Lake Campus, Kolkata, West Bengal 700106, India
[2]School of Mathematical and Computational Sciences, Indian Association for the Cultivation of Science, Kolkata, West Bengal 700032, India
[3]Techno India University, West Bengal, Kolkata 700091, India
[4]G S Sanyal School of Telecommunication, Indian Institute of Technology Kharagpur, Kharagpur, West Bengal 721302, India
[5]Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, Kharagpur, West Bengal 721302, India
[6]Department of Computer Science, Ashoka University, Sonipat, Haryana 131029, India

Corresponding author: Tohida Rehman (tohidarehman.it@jadavpuruniversity.in)

The work of Debarshi Kumar Sanyal was supported in part by the Department of Science and Technology, Government of India, under Grant CRG/2021/000803.

ABSTRACT Nowadays many research articles are prefaced with research highlights to summarize the main findings of the paper. Highlights not only help researchers precisely and quickly identify the contributions of a paper, they also enhance the discoverability of the article via search engines. We aim to automatically construct research highlights given certain segments of a research paper. We use a pointer-generator network with coverage mechanism and a contextual embedding layer at the input that encodes the input tokens into SciBERT embeddings. We test our model on a benchmark dataset, CSPubSum, and also present MixSub, a new multi-disciplinary corpus of papers for automatic research highlight generation. For both CSPubSum and MixSub, we have observed that the proposed model achieves the best performance compared to related variants and other models proposed in the literature. On the CSPubSum dataset, our model achieves the best performance when the input is only the abstract of a paper as opposed to other segments of the paper. It produces ROUGE-1, ROUGE-2 and ROUGE-L F1-scores of 38.26, 14.26 and 35.51, respectively, METEOR score of 32.62, and BERTScore F1 of 86.65 which outperform all other baselines. On the new MixSub dataset, where only the abstract is the input, our proposed model (when trained on the whole training corpus without distinguishing between the subject categories) achieves ROUGE-1, ROUGE-2 and ROUGE-L F1-scores of 31.78, 9.76 and 29.3, respectively, METEOR score of 24.00, and BERTScore F1 of 85.25.

INDEX TERMS Deep learning, natural language generation, pointer-generator network, SciBERT, scientific data.

I. INTRODUCTION

Scientific publications are growing at an exponential rate [1]. It has been reported that the number of scientific articles doubles roughly every nine years [2]. Even in a limited subfield, scientists find it very challenging to keep track of the cutting edge of research. Therefore, to make it easier for researchers to appreciate the main import of a paper, publishers have adopted many novel presentation techniques. One recent trend is to complement the abstract of a paper with *research highlights*, a list of points summarizing the main findings of the paper. Research highlights are typically written by the author along with the abstract. They are often easier to read and grasp than a longer paragraph, especially on hand-held devices. Moreover, research highlights can be used by search engines for indexing the articles and subsequently, retrieve or recommend them to the appropriate users. Yet, not all scholarly articles contain research highlights written by the authors.

Research highlights and abstract are both *summaries* of the research paper. Text summarization is a process to present the gist of a source document or a set of related documents.

The associate editor coordinating the review of this manuscript and approving it for publication was Li Zhang.

The main benefit of text summarization is that it reduces the amount of time the reader has to spend to extract the main information in the document. Extractive summarization and abstractive summarization are two broad approaches used in automatic text summarization [3], [4]. Extractive approaches [5] simply copy some relevant sentences from the documents and ignore the rest. Abstractive approaches [6] can induce new relevant words in the summary in the same way that a person does – they first read the entire text, comprehend it, and then summarize using suitable new words. Therefore, abstractive approaches typically provide better summaries compared to those produced by extractive methods.

In this paper, we aim to extract research highlights from a research paper using abstractive approaches. From a simple manual analysis, we found that most of the information present in research highlights occur in the abstract, introduction, and conclusion sections of a paper. Therefore, we provide these sections and their combinations as inputs to our summarizer. Our model is an adaptation of the pointer-generator network with coverage mechanism [7]. However, unlike the original model, we use an additional embedding layer at the input. This layer encodes each word of the input document with embeddings from SciBERT, which is a BERT model trained on a large corpus of scientific documents. We expect these contextual embeddings to help the model generate better quality abstractive summary compared to that produced by the vanilla model.

The main contributions of this paper are:

1) Our method automatically generates research highlights from a scientific research paper. We propose a technique to combine a SciBERT [8] pre-trained layer of word embeddings with a pointer-generator network that also uses a coverage mechanism. To the best of our knowledge, this work is the first attempt to use SciBERT with a pointer-generator model augmented with coverage mechanism [7] to generate research highlights.
2) We present a new multi-disciplinary dataset named MixSub that contains research papers (with author-written highlights) from different subject domains.
3) For one of the datasets, namely, the CSPubSum dataset, we analyze the performance of generating research highlights for the following different input types: (a) the input is the abstract only, (b) the input is the introduction only, (c) the input is the conclusion only, (d) the input comprises the abstract and the conclusion, (e) the input comprises the introduction and the conclusion. For the MixSub dataset, we use only the abstract as the input. We use ROUGH [9], METEOR [10], and BERTScore [11] metrics to evaluate the performance of the models. We show that our model performs better than existing baselines proposed for this task.

The rest of this article is organized as follows. Section II is an overview of prior work in the field. Section III describes the proposed model. Section IV describes the experimental setup, the datasets we used, including a new dataset called MixSub which we construct as part of this work. Section V reports the results we obtained by using the CSPubSum dataset and our new MixSub dataset. Within this section, we compare our method with a few competitive pre-trained models in the literature, and analyze the energy consumption of the various models. Section VI depicts several case studies. Finally, the paper concludes in Section VII.

II. LITERATURE REVIEW

The advancement of sequence-to-sequence models [12] has significantly improved the state-of-the-art in abstractive summarization [13]. Attention-based encoder of with a beam-search decoder has achieved significant performance in abstractive text summarization on DUC 2004 dataset [14]. Convolutional attention-based conditional recurrent neural network was used to further improve the performance on Gigaword Corpus and DUC 2004 dataset [15]. Nallapati et al. [16] proposed a model for abstractive text summarization based on attentional encoder decoder recurrent neural networks. To remove out-of-vocabulary words (OOV) and repeating words, a hybrid approach called pointer-generator network with coverage mechanism has been proposed [7]. It can copy words from the source text by pointing and uses coverage to keep track of what is summarized to avoid repetition. To represent the semantic information of words more correctly, Anh and Trang [17] have used two pre-trained word embeddings, namely, word2vec and FastText, with a pointer-generator model for the CNN/Daily Mail dataset and achieved an impressive performance. Du et al. [18] proposed a model to extract summary based on fuzzy logic rules, multi-feature set and genetic algorithm on DUC2002 dataset.

Recently, pre-trained language models that generate contextual embeddings have become extremely popular and shown to achieve state-of-the-art results in many NLP tasks. Their mode of operation is as follows: train the model on a large corpus and then fine-tune it on various downstream task in NLP. Radford et al. [19] proposed Generative Pre-Training (GPT), which combines unsupervised pre-training and supervised fine-tuning, to improve language understanding. The implementation of the transformer architecture and its bidirectional encoder model BERT resulted in improved performance in downstream NLP tasks including text summarization [20]. BERT trains a deep bidirectional transformer encoder, which learns interactions between left and right context, using a masked language modelling objective [20]. For a new corpus, BERT can be fine-tuned for sentence-label and token-label tasks. Knowledge graphs (KGs) can be combined with BERT to capture the lexical, syntactic, and knowledge information at the same time [21]. Researchers have also made available a few large deep neural models that are pre-trained specifically for the summarization task. A pre-trained model PEGASUS, trained using large pre-training corpora and a gap sentence generating task has been evaluated on 12 downstream summarization tasks [22]. To overcome the disadvantages of limited input size in a BERT-based architecture, the BERT windowing method can be used [23]. Raffel et al. [24] proposed T5, which is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised workloads, with each task transformed to text-to-text processing. BART [25] is a transformer encoder-encoder (seq2seq) model with a bidirectional BERT encoder and an autoregressive decoder (more specifically, generative pre-trained transformer or GPT). The pre-training task in BART entails changing the sequence of the original phrases at random and using a novel in-filling strategy that replaces text spans with a single mask token. BART is especially effective when fine-tuned for text generation, but it also performs well for comprehension tasks.

Early work on extractive summarization of scientific documents was done with limited datasets, such as one of 188 document and summary pairs [5] where all the documents were gathered from 21 scientific/technical publications. A summarizing technique that focuses on the rhetorical status of assertions in 80 scientific articles, part of a larger corpus of 260 articles, has been developed by Teufel and Moens [26]. A sentence-based automatic summarizing system has been built based on feature extraction and query-focused methods [27]. Lloret et al. [28] have proposed a method to automatically generate the abstract of a research paper in the biomedical domain. They used two approaches – extractive as well as abstractive. To better deal with the long text of a research paper in abstractive summarization, a multiple timescale model of the gated recurrent unit (MTGRU) has been used in [29]. They have contributed a new corpus containing pairs of (introduction, abstract) of computer science papers from arXiv.org. Souza et al. [30] have proposed a multi-view extractive text summarization approach for long scientific texts.

Recent advancements have attempted to summarize entire research papers, focusing specifically on the generation of the paper title from the abstract (title-gen) and the generation of the abstract from the body of the paper (abstract-gen) in biomedical domain [31]. Since the keyphrases in a paper may be assumed to capture the main aspects of the paper, extraction [32], [33] or generation [34], [35] of keyphrases is a related area of research. However, we do not discuss it further as it does not produce sentential forms typically observed in a summary.

Generating research highlights from scientific articles is different than document summarization. Collins et al. [36] have developed supervised machine learning methods to identify relevant highlights from the full text of a paper using a binary classifier. They also contributed a new benchmark dataset of URLs, which includes approximately 10,000 articles from computer science domain, labelled with relevant author-written highlights. Using multivariate regression methods for the same problem, Cagliero and Quatra in [37] selected the top-k most relevant sentences from a paper as research highlights, unlike a simple binary classification of sentences as highlights or not. Note that this is also extractive in nature. Rehman et al. [38] proposed an abstractive summarization model based on pointer-generator network with coverage and GloVe embeddings to generate research highlights from abstracts. Later, Rehman et al. [39] combined named entity recognition with pointer-generator networks to improve the performance of their method. In contrast to previous works, in the current one we use pre-trained SciBERT word embeddings and propose a new dataset.

III. METHODOLOGY

We use pointer-generator networks to produce highlights from research papers. It consists of a seq2seq model with a BiLSTM encoder and an LSTM decoder with attention [16]. However, while the original model proposed by See et al. [7] uses word-embeddings – they are learned from scratch during training – we use a pre-trained transformer to generate the contextual embeddings of the tokens in the input document. The architecture of our model is shown in Figure 1.

We perform experiments with 4 variants: (1) the original pointer-generator model proposed in [7], (2) pointer-generator model integrating coverage mechanism (proposed in [40]), described in the same work [7], (3) pointer-generator model with SciBERT, and (4) pointer-generator model with SciBERT and coverage mechanism.

A. BERT AND SciBERT

The pre-trained language model BERT stands for *bidirectional encoder representations from transformers*. BERT has been pre-trained on the tasks of masked language modelling (MLM) and next sentence prediction (NSP) [20]. Normally, standard conditional language models are trained on either left-to-right or right-to-left representations of the context, but MLM used both left-to-right and right-to-left representations of the context. The primary goal of the masked language model is to predict the actual vocabulary identifier of the input's randomly masked tokens. Next sentence prediction (NSP) aids the model in comprehending sentence relationships. This feature helps to improve the performance for the downstream tasks of question-answering (QA) and natural language inference (NLI). To encode the input, the input sentence is first tokenized, and then the tokens are combined with 3 new tokens, namely, CLS, SEP, and MASK. CLS is added at the start of sentence to represent sentence-level classification. To predict the next sentence, SEP is used. During the MLM task, MASK is used to represent masked tokens. English Wikipedia (2,500M words) and the BooksCorpus (800M words) are used for pre-training the BERT model. Summing the corresponding token, segment, and position embeddings yields the input representation for a given token. Primarily, BERT has two variants named as $BERT_{BASE}$ and $BERT_{LARGE}$. $BERT_{BASE}$ has 12 transformer layers, 768 hidden size, 12 attention heads, and 110M total parameters. $BERT_{LARGE}$ has 24 transformer layers, 1024 hidden size, 16 attention heads and 340M total parameters.

SciBERT is a BERT-based pre-trained language model that was trained on a large corpus of scientific text from

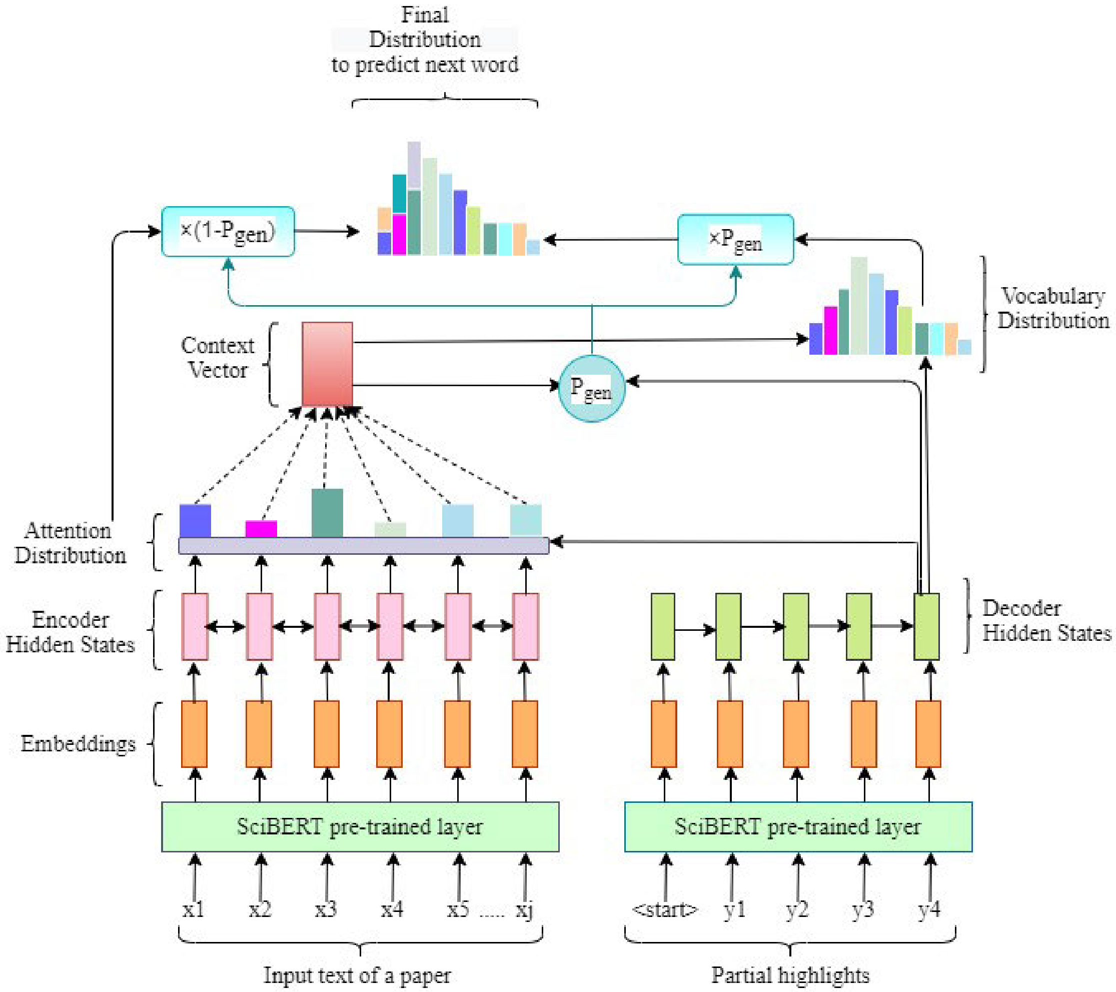


FIGURE 1. **Proposed model: Pointer-generator network with coverage mechanism and SciBERT word embeddings.**

Semantic Scholar [41]. The same size and configuration of *BERT$_{BASE}$* is used to train the SciBERT model and allowed 128 tokens of maximum sentence length. SciBERT has 4 variants: `cased`/`uncased` and `basevocab`/`scivocab`. The `basevocab` models are fine-tuned from the corresponding *BERT$_{BASE}$* models. The `scivocab` models have been trained from scratch.

B. POINTER-GENERATOR NETWORK WITH SciBERT

This model consists of a word-embedding layer and a pointer-generator network. The word-embedding layer converts the words in the input document to embeddings. We have used a pre-trained SciBERT model [8] to generate word embeddings. Using this mechanism, each word (x_t) in the encoder and the decoder part will be represented as an embedding vector $\mathbf{x}_t$ as:

$$\mathbf{x}_t = g(x_t) \tag{1}$$

where $g(.)$ is the embedding-generating function. The CLS token has been added to represent sentence-level classification. Here, the main use of SciBERT [8] is that instead of directly feeding the token ids of the input document into the encoder, we are passing the pre-trained SciBERT-generated word embeddings. In our experiments, the dimension of word embeddings is 768. A pointer-generator network [7] augments the sequence-to-sequence (seq2seq) model with attention [16] using a special copying mechanism. When generating words, the decoder probabilistically decides between generating new words from the vocabulary (i.e., from the training corpus) and copying words from the input document (by sampling from the attention distribution). While the generator helps in novel paraphrasing, copying helps to tackle OOV words. This improves the model's ability to calculate hidden states because the inputs at each time step have been accurately and completely represented, contributing to the improvement of the attention distribution. At each decoder time step t, the probability of generating a new word is

$$P_{gen} = \sigma(\mathbf{W}_{h^*}^{\top}\mathbf{h}_t^* + \mathbf{W}_s^{\top}\mathbf{s}_t + \mathbf{W}_x^{\top}\mathbf{x}_t + \mathbf{b}_{ptr}) \tag{2}$$

where $\mathbf{h}_t^*$ is the context vector, $\mathbf{s}_t$ is the decoder hidden state, $\mathbf{x}_t$ is the decoder input (which is the decoder output at time $t-1$ during test, and the correct word at time $t-1$ during training), σ is the *sigmoid function*, and $\mathbf{W}_{h^*}$, $\mathbf{W}_s$, $\mathbf{W}_x$ and $\mathbf{b}_{ptr}$ are the learnable parameters. Hence, for the SciBERT pre-trained embeddings layer the formula in (2) is modified as follows:

$$P_{gen} = \sigma(\mathbf{W}_{h^*}^\top \mathbf{h}_t^* + \mathbf{W}_s^\top \mathbf{s}_t + \mathbf{W}_x^\top g(x_t) + \mathbf{b}_{ptr}) \tag{3}$$

To predict the next word y_t, the probability distribution over the extended vocabulary (i.e., the fixed vocabulary of the training corpus and the present document) is calculated:

$$P(y_t) = P_{gen} P_{vocab}(y_t) + (1 - P_{gen}) \sum_{i:w_i=y_t} a_{t,i} \tag{4}$$

where $\mathbf{a}_t$ is the attention distribution over the fixed vocabulary at time t, $a_{t,i}$ is the attention over the word w_i at time t, and P_{vocab} is the probability distribution over the extended vocabulary generated by the softmax layer of the decoder. The loss for decoder time step t is:

$$loss_t = -\log P(y_t^*) \tag{5}$$

where y_t^* is the target word. The overall loss for the sequence is the average of the losses over all the decoder time steps for this sequence.

C. POINTER-GENERATOR + COVERAGE MECHANISM WITH SciBERT

Sometimes the above pointer-generator network redundantly generates the same word multiple times during test. The coverage model of Tu et al. [40] aims to address this problem. This model essentially gives attention to the previous timesteps of the decoder. It computes a coverage vector $\mathbf{c}^t$ defined as the sum of the attention distributions $\mathbf{a}_t$ over all previous timesteps $\tau = 1$ to $\tau = t - 1$ of the decoder:

$$\mathbf{c}_t = \sum\nolimits_{\tau=0}^{t-1} \mathbf{a}_\tau \tag{6}$$

Note that $\mathbf{c}_0$ is a zero vector. The *coverage vector* will be taken as an extra input to the attention mechanism that is used by the decoder while generating the next word.

The *coverage loss* quantifies if the model is continuously giving more attention to the same words:

$$CoverageLoss_t = \sum_i \min(a_{t,i}, c_{t,i}) \tag{7}$$

Finally, the coverage loss is included in the primary loss function of the decoder. The revised loss for decoder time step t can be written using a hyperparameter λ as follows:

$$loss_t = -\log P(y_t^*) + \lambda \sum_i \min(a_{t,i}, c_{t,i}) \tag{8}$$

IV. EXPERIMENTAL SETUP

In this section, we discuss the datasets, the data preprocessing steps, and the experiments.

A. DATASETS

1) DATASET OF COMPUTER SCIENCE PAPERS

We use the dataset CSPubSum released by Collins et al. [36] containing URLs of 10142 computer science publications from ScienceDirect.[1] Every document contains the following fields: title, abstract, research highlights written by the authors, a list of keywords mentioned by the authors, and various sections such as introduction, related work, experiment, and conclusion, as typically found in a research paper. We organize each example in this dataset as *(abstract, author-written research highlights, introduction, conclusion)*. We have observed that, here, the average abstract size is 186 words while that of highlights is 52; and for 98% of the papers, highlights are 1.5 times or more shorter than the abstract. Thus, at least in terms of the word length, highlights can be considered as a summary not only of the paper but also of the abstract. For our experiments, we split the dataset into train/dev/test in the ratio 80 : 10 : 10, that is, reserve 8115 examples for training, 1014 examples for validation, and 1013 examples for testing. We use this holdout test set to measure the performance of our models on CSPubSum in all cases, except in Section V-A2 where we present some results obtained with K-fold cross-validation.

2) MixSub: A NEW DATASET OF PAPERS FROM MULTIPLE DOMAINS

We propose a new dataset called **MixSub** that contains research articles from multiple domains. Note that the CSPubSum corpus from [36] contains only computer science papers. To prepare MixSub, we crawled the ScienceDirect website and curated articles published in various journals in year 2020. We removed the articles that did not contain research highlights. Finally, we got 19785 articles with author-written research highlights as shown in Table 1. Each example in this dataset is organized as *(abstract, author-written research highlights)*. We have also segmented the dataset into training, validation and test subsets. In this corpus, the average abstract size is 148 words while that of highlights is 57. For 72% of the papers, highlights are 1.5 times or more shorter than the abstract. We split each category of documents into train/dev/test subsets in the ratio 80 : 10 : 10. We have grouped similar journal papers according to their domain as shown in Table 1 and also highlighted using a pie chart Figure 2. A summary of the above two datasets is shown in Table 2.

B. DATA PRE-PROCESSING

We have used the Stanford CoreNLP Tokenizer[2] for tokenizing the sentences. The whole corpus is first converted to lowercase. We have removed all unnecessary symbols, letters, and other elements from the text that do not affect the aim of our research. In particular, HTML tags, parentheses, and special characters have been removed.

Then we reorganized the dataset in several ways to perform various experiments. More specifically, for CSPubSum, we organize it as *(abstract, author-written research highlights)*, *(conclusion, author-written research*

[1] https://www.sciencedirect.com/

[2] https://stanfordnlp.github.io/CoreNLP/

TABLE 1. Subject-wise URL count in MixSub dataset.

Domain Name	Subject name	#Total	#Train	#Val	#Test
Biological	Agricultural and Biological Sciences	2156	1726	216	214
	Biochemistry, Genetics and Molecular Biology	976	806	71	99
	Immunology and Microbiology	233	195	24	14
	Neuroscience	962	771	96	95
Chemistry	Chemical Engineering	2140	1713	214	213
	Chemistry	2282	1919	240	123
	Materials Science	735	572	82	81
Energy	Energy	1313	1025	145	143
	Environmental Science	677	517	81	79
Management	Business, Management and Accounting	698	560	70	68
	Decision Sciences	947	759	95	93
	Economics, Econometrics and Finance	421	324	56	41
Nursing	Health Sciences	823	796	12	15
	Nursing and Health Professions	61	47	8	6
	Pharmacology,Toxicology,Pharmaceutical Science	1184	949	118	117
	Psychology	28	24	3	1
	Veterinary Science and Veterinary Medicine	186	156	19	11
Physics	Earth and Planetary Sciences	1354	1038	159	157
	Mathematics	288	232	29	27
	Physics and Astronomy	1469	1177	147	145
Social Science	Social Sciences	852	654	100	98

TABLE 2. Some statistics of CSPubSum and MixSub datasets.

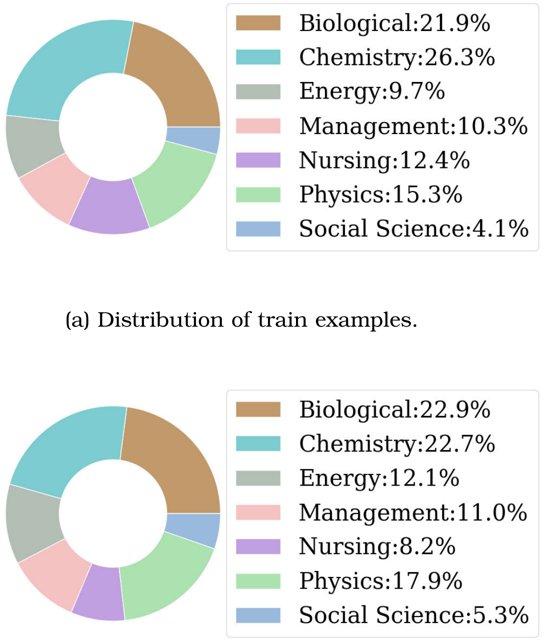
Dataset	Train	Val	Test	Average Words (abstract)	Average Words (highlight)	% of article-pairs where compression >= 1.5 times
CSPubSum	8115	1014	1013	186	52	98
MixSub	15960	1985	1840	148	57	72

highlights), *(introduction, author-written research highlights)*, *(abstract + conclusion, author-written research highlights)*, and *(introduction + conclusion, author-written research highlights)* where '+' denotes text concatenation. Since the background and a broad summary of the paper normally appear in the introduction, and the main findings of the paper are mentioned in the conclusion, we experiment taking these sections as inputs. Since an overview of the paper is present both in the introduction and the abstract, we do not use them together, rather we use the combinations (abstract + conclusion), and (introduction + conclusion). In case of MixSub, we only have *(abstract, author-written research highlights)* because we get the best results on CSPubSum using only abstracts. In future, we might explore the use of full-text or sections from the full-text for MixSub as well.

When the abstract is used as the input, we set the maximum number of input tokens to 400. When the conclusion is used as the input, the maximum number of input tokens allowed is 800. When the introduction is used as the input, the maximum number of input tokens allowed is set to 1200. For all other inputs, we have restricted the input size to 1500 tokens. In all cases, the maximum token count of the generated research highlights tokens is set to 100. The above figures are motivated by the observation that the average length of an abstract is 186, the average length of the author-written highlights in a paper is 52, and the average length of the conclusion is 425, that of the introduction is 837, the average length of (abstract + conclusion) is 643, and that of (introduction + conclusion) is 1230.

C. IMPLEMENTATION DETAILS

We trained four variants of the proposed model: pointer-generator network with word embeddings trained from scratch as part of the model training (**PGM**), pointer-generator network with coverage mechanism where word embeddings are trained from scratch as part of the model training (**PGM + Coverage**), pointer-generator network with SciBERT embeddings for the input tokens (**PGM + SciBERT**), and pointer-generator network with coverage mechanism and SciBERT embeddings for the input tokens (**PGM + Coverage + SciBERT**). For all variants of SciBERT models, during model training, the embeddings are fine-tuned. We trained all models on Tesla P100-PCIE-16GB `Colab Pro+` that supports GPU-based training. We used mini-batches of size 16. For all models, we used bidirectional LSTMs with cell size of 256. For models without SciBERT, word embeddings of dimension 128 are trained end-to-end with the model. For models with SciBERT, pre-trained word embeddings of dimension 768 are used. For all experiments, we constrained the vocabulary size to the most



(a) Distribution of train examples.

(b) Distribution of test examples.

FIGURE 2. Subject-wise distribution of papers in MixSub dataset.

frequent 50,000 tokens. We considered gradient clipping with a maximum gradient norm of 1.2. Out of the four variations of the SciBERT model, we use `SciVocab-uncased`.[3] We used other hyperparameters as suggested by [7]. We have used the validation set to determine the number of epochs for training.

D. EVALUATION METRICS

To evaluate the performance of the models, we use ROUGH [9], METEOR [10], and BERTScore [11] metrics. These are the standard metrics used to measure the performance of summarization models [42]. When comparing the model-generated research highlights (`ModelHighlights`) with the author-written research highlights (`AuthorHighlights`) for assessment, ROUGE-n calculates the recall, precision, and F1-measure for each model using equations (9), (10) and (11). Note that an n-gram is a contiguous sequence of n words from a piece of text. Recall (R) is defined as:

$$R = \frac{\#matched\ n-grams}{\#n-grams\ in\ \texttt{AuthorHighlights}} \tag{9}$$

Precision (P) is defined as:

$$P = \frac{\#matched\ n-grams}{\#n-grams\ in\ \texttt{ModelHighlights}} \tag{10}$$

F1-measure ($F1$) is calculated using the formula:

$$F1 = 2 * \frac{R * P}{R + P} \tag{11}$$

We have used ROUGE-1, ROUGE-2 and ROUGH-L. In particular, ROUGE-L measures the longest matching sub-sequence of words between the two strings. All our ROUGE scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE script [9].

METEOR assigns a score to the match based on a combination of unigram precision, unigram recall, and a fragmentation measure that is intended to directly represent how well-ordered the matched words in the model-generated research highlights and author-written research highlights are. It calculates recall (R) and precision (P) of unigrams based on equations (9) and (10), respectively. Next, it computes F_{mean} score and chunk penalty using the formula:

$$F_{mean} = \frac{10(R * P)}{R + 9P} \tag{12}$$

$$Penalty = 0.5 * \left(\frac{\#chunks}{\#unigrams\_matched}\right)^3 \tag{13}$$

where *chunks* is defined as a set of unigrams that are adjacent in the `ModelHighlights` and in the `AuthorHighlights`. The final METEOR score is computed as follows:

$$Score = F_{mean} * (1 - Penalty) \tag{14}$$

For BERTScore computation, we consider the cosine similarity of contextual embeddings of each word from model-generated research highlights and author-written research highlights, instead of counting the exact words matched across them. Denoting the contextual embeddings of the author-written research highlights by $\mathbf{x} = \langle \mathbf{x}_1, \ldots, \mathbf{x}_n \rangle$ and those of the model-generated research highlights by $\hat{\mathbf{x}} = \langle \hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_m \rangle$, the recall ($R_{\text{BERT}}$), precision ($P_{\text{BERT}}$), and F1-scores ($F_{\text{BERT}}$) are calculated as follows:

$$R_{\text{BERT}} = \frac{1}{m} \sum_{\mathbf{x}_i \in \mathbf{x}} \max_{\hat{\mathbf{x}}_j \in \hat{\mathbf{x}}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \tag{15}$$

$$P_{\text{BERT}} = \frac{1}{n} \sum_{\hat{\mathbf{x}}_j \in \hat{\mathbf{x}}} \max_{\mathbf{x}_i \in \mathbf{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \tag{16}$$

$$F1_{\text{BERT}} = 2 * \frac{R_{\text{BERT}} * P_{\text{BERT}}}{R_{\text{BERT}} + P_{\text{BERT}}} \tag{17}$$

V. RESULTS

A. COMPARISON OF POINTER-GENERATOR TYPE MODELS

1) EVALUATION ON CSPubSum DATASET

In this sub-section, we report the results of experiments on the CSPubSum dataset for various input types.

a: INPUT: ABSTRACT

Results are shown in Table 3 for ROUGE-1, ROUGE-2, ROUGE-L, METEOR and BERTScore when the input is the

[3] https://huggingface.co/allenai/scibert_scivocab_uncased/

TABLE 3. Evaluation of pointer-generator type models: scores for ROUGE, METEOR and BERTScore on various inputs from CSPubSum dataset. All our ROUGE scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE script.

		ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BERTScore
Input	Model Name	F1	F1	F1	Final score	F1
abstract only	PGM	35.44	11.57	29.88	25.4	83.80
	PGM + Coverage	36.57	12.3	30.69	25.4	84.05
	PGM + SciBERT	36.55	13.44	33.57	30.34	86.34
	PGM + Coverage + SciBERT	**38.26**	**14.26**	**35.51**	**32.62**	**86.65**
conclusion only	PGM	32.11	9.32	29.62	24.04	85.72
	PGM + Coverage	34.33	9.73	31.71	24.99	86.07
	PGM + SciBERT	33.19	9.8	30.49	24.26	86.03
	PGM + Coverage + SciBERT	**34.81**	**10.02**	**32.31**	**25.21**	**86.52**
introduction only	PGM	30.85	7.92	28.55	19.76	85.25
	PGM + Coverage	32.46	8.18	30	20.50	85.48
	PGM + SciBERT	31.56	8.79	29.18	23.09	85.93
	PGM + Coverage + SciBERT	**33.33**	**9.7**	**30.86**	**24.10**	**86.17**
abstract + conclusion	PGM	29.85	8.16	25.80	19.38	83.19
	PGM + Coverage	31.70	8.31	26.72	20.92	83.49
	PGM + SciBERT	32.84	9.86	30.34	24.59	86.13
	PGM + Coverage + SciBERT	**35.09**	**10.94**	**32.69**	**27.31**	**86.52**
introduction + conclusion	PGM	29.78	7.47	25.15	19.25	83.05
	PGM + Coverage	31.63	7.65	26.25	20.24	83.32
	PGM + SciBERT	32.38	9.63	29.79	23.95	86.11
	PGM + Coverage + SciBERT	**35.32**	**10.93**	**32.76**	**26.57**	**86.59**

abstract of a research paper. We observe that among the four models, the pointer-generator network with coverage mechanism and SciBERT (**PGM + Coverage + SciBERT**) achieve the highest ROUGE, METEOR and BERTScore values.

b: INPUT: CONCLUSION

Results are shown in Table 3 for ROUGE-1, ROUGE-2, ROUGE-L, METEOR and BERTScore when the input is only the conclusion of a research paper. We observe that among the four models, the (PGM + Coverage + SciBERT) model achieves the highest ROUGE, METEOR and BERTScore values.

c: INPUT: INTRODUCTION

Results are shown in Table 3 for ROUGE-1, ROUGE-2, ROUGE-L, METEOR and BERTScore when the input is the introduction of a research paper. We observe that among the four models, the (PGM + Coverage + SciBERT) model achieves the highest ROUGE, METEOR and BERTScore values.

d: INPUT: ABSTRACT + CONCLUSION

Results are shown in Table 3 for ROUGE-1, ROUGE-2, ROUGE-L scores, METEOR and BERTScore when the input is the combination of the abstract and the conclusion of a paper. We again observe that the best performance is achieved by the (PGM + coverage + SciBERT) model.

e: INPUT: INTRODUCTION + CONCLUSION:

When the inputs is the combination of introduction and conclusion in the test dataset, we record ROUGE-1, ROUGE-2, ROUGE-L scores, METEOR and BERTScore as shown in Table 3. The best performing model is (PGM + coverage + SciBERT). Upon analysis of the dataset, we found that in many cases the highlights are largely included in the 'abstract'; therefore, using the 'abstract' as input to the model results in high performance. We have observed that the 'conclusion' typically presents a more detailed and technically dense description of the findings in contrast to the more overview-style summary included in the research highlights (see, for example, these papers[4] [5]). The 'conclusion' also includes future work, which does not form part of the highlights. So adding the 'conclusion' with the 'abstract' does not improve the performance. Although the 'introduction' of a paper often contains the main findings of the paper, it also contains a lot of other information (typically, to build the background and context to the current work) that is not included in the highlights and must be filtered away by the model when generating the output.

2) *K*-FOLD CROSS-VALIDATION

We also perform K-fold cross-validation (CV) of our model (**PGM + Coverage + SciBERT**) on the CSPubSum dataset. For this purpose, we set $K = 5$, that is, we split the whole dataset into five distinct parts. We trained using four parts (or folds) and tested the model using the remaining part. In each case, we trained the pointer-generator network with SciBERT for 20000 iterations, then added the coverage mechanism and continued training for another 1000 iterations. In all cases, we consider only the abstracts of the CSPubSum dataset as the input. Table 4 reports the ROUGE, METEOR and BERTScore for the model (PGM + Coverage + SciBERT) with 5-fold cross-validation and compares the performance with that of holdout validation. Since K-fold cross-validation is computationally quite expensive, we did not conduct it for

[4] https://www.sciencedirect.com/science/article/abs/pii/S0010448514001870

[5] https://www.sciencedirect.com/science/article/pii/S0010448514001638

TABLE 4. *K*-fold cross-validation of the proposed models on CSPubSum dataset. For comparison, the performance of the models with holdout validation are reproduced from Table 3.

Input	Model Name	ROUGE-1 F1	ROUGE-2 F1	ROUGE-L F1	METEOR Final score	BERTScore F1
abstract only (holdout validation)	PGM	35.44	11.57	29.88	25.4	83.80
	PGM + Coverage	36.57	12.3	30.69	25.4	84.05
	PGM + SciBERT	36.55	13.44	33.57	30.34	86.34
	PGM + Coverage + SciBERT	**38.26**	**14.26**	**35.51**	**32.62**	**86.65**
abstract only (5-fold CV)	PGM + SciBERT	37.79	12.77	34.78	29.92	86.72
	PGM + Coverage + SciBERT	**39.43**	**15.25**	**36.48**	**30.85**	**87.01**

the other input types. Note that the performance achieved by K-fold cross-validation is slightly higher than that reported by holdout validation. Since it is widely believed (see, for example, [43], [44]) that K-fold cross-validation results are a better indicator of the generalization performance, our model is likely to be better than that indicated by holdout testing.

3) COMPARISON WITH PREVIOUS WORKS

Table 5 compares the performance of our proposed approach (**PGM + Coverage + SciBERT**) with other competitive baselines in the literature, namely, an **LSTM-based** extractive summarization model [36], a **gradient boosting regression** extractive summarization model [37], and a **PGM model with GloVe embeddings** for abstractive summarization [38], on the CSPubSum dataset in terms of the ROUGE-1 (F1), ROUGE-2 (F1), ROUGE-L (F1), METEOR and BERTScore (F1) metrics.

ROUGE-2 F1-score and ROUGE-L F1-score of the LSTM-based model in [36] are 12.7 and 29.50, respectively while those in the gradient boosting regression model [37] are 13.9 and 31.6, respectively. Both the above methods use extractive summarization on the full text (sans abstract) of the paper, that is, they select a set of sentences from a given document for inclusion in the research highlights. Rehman et al. [38] use abstractive summarization to generate research highlights from abstracts only, and the best performing model in it is a pointer-generator network with coverage and GloVe embeddings that records ROUGE-1 F1, ROUGE-2 F1, ROUGE-L F1, METEOR score, and BERTScore F1 values as 31.46, 8.57, 29.14, 12.01 and 85.31, respectively. In a follow-up work, Rehman et al. [39] combined named-entity recognition (NER) with coverage-augmented pointer-generator network to generate research highlights from different parts of a paper. The best performing model in [39] is denoted as (NER + PGM + Coverage) in Table 5, and it uses only the abstract; it produces ROUGE-1 F1, ROUGE-2 F1, ROUGE-L F1, METEOR, and BERTScore F1 values as 38.13, 13.68, 35.11, 31.03 and 86.3, respectively. We clearly observe that the method proposed in this paper, i.e., a pointer-generator network with coverage and SciBERT word embeddings, achieves the best ROUGE-2 F1-score and ROUGE-L F1-score which are 14.26 and 35.51, respectively. The same model also achieves the highest METEOR score and BERTScore F1 of 32.62 and 86.65, respectively, among the three pointer-generator models compared in Table 5. (Note that here we have measured the performance on the holdout test set.) The above model (PGM + Coverage + SciBERT) uses only the abstracts as input unlike the methods in [36] and [37], that use the full text of the paper. Abstracts being much shorter than the main text of a paper, the computational overhead is significantly reduced. Our method establishes a new state-of-the-art for the CSPubSum dataset.

4) EVALUATION ON MixSub DATASET

In this sub-section, we report the results of experiments on the MixSub dataset. We trained the models in two ways:

- **Case 1:** We trained all the four models on each subject cluster separately and tested them on the corresponding test documents.
- **Case 2:** We did not distinguish between the subject categories of the papers but simply collected all the documents of the training corpus, and trained the models. Then we evaluated them on the test corpus and reported the results for each subject category.

Note that in each case, the input is only the abstract of a paper. Since MixSub currently does not contain the body of a paper, we cannot use other sections of a paper as the input. Results are reported in Table 6 for ROUGE-1, ROUGE-2, ROUGE-L, METEOR and BERTScore. The top row labeled 'Full MixSub' shows the results when the models are trained on the whole training corpus without regard to the specific subject category of the papers and tested on the test corpus, again without regard to the specific subject category of the papers. The remaining rows show the scores obtained on each category of papers when the models are trained either on the respective clusters (Case 1) or on the whole training corpus without regard to subject category (Case 2). We observe that among the four models, (PGM + coverage + SciBERT) achieves the highest ROUGE, METEOR and BERTScore values. We observe that sometimes training on subject-specific clusters leads to higher scores and at other times, training on the whole corpus produces better scores at the subject level. But (**PGM + Coverage + SciBERT**) outperforms all the other models in all cases.

B. COMPARISON WITH PRE-TRAINED MODELS

We have chosen the following pre-trained models from the Hugging Face website for the purpose of comparison:

TABLE 5. Comparison of the performance of the proposed model with that of other approaches for CSPubSum dataset.

Model Name	ROUGE-1 F1	ROUGE-2 F1	ROUGE-L F1	METEOR Final score	BERTScore F1
LSTM Classification [36]	—	12.7	29.50	—	—
Gradient Boosting Regressor [37]	—	13.9	31.60	—	—
PGM + Coverage + GloVe [38]	31.46	8.57	29.14	12.01	85.31
NER + PGM + Coverage [39]	38.13	13.68	35.11	31.03	86.3
PGM + Coverage + SciBERT (ours)	**38.26**	**14.26**	**35.51**	**32.62**	**86.65**

T5-base,[6] **Distilbart-CNN-12-6**,[7] **GPT-2**[8] and **ProphetNet-large-uncased-cnndm**.[9] We fine-tuned all four models to 15 epochs with CSPubSum where 8115 documents (each comprising an abstract and author written research highlights) are taken for training. We tested them on the test dataset of 1013 examples. We used a batch size of 4 for fine-tuning all four pre-trained models. Observations on the test set are shown in Table 7. The performance of ProphetNet-large-uncased-cnndm pretrained model is significantly worse than that of other models; the training duration and compute resources we used appeared to be inadequate for this model. We observe that T5-base performs better than the other models in terms of ROUGE and BERTScore metrics while Distilbart-CNN-12-6 gives the highest METEOR score. The slight performance gain of pre-trained models is not surprising at all given the number of parameters and the exhaustiveness of the training of such models. Rather the closeness of the proposed model, which does not require fine-tuning a large pre-trained transformer model, appears to demand more attention to strike the right trade-off between performance and the resources needed for training.

In the next sub-section, we will discuss an important aspect of these large models, which has received attention in the recent years. This aspect deals with the energy efficiency of algorithms that is also related to the consequent carbon footprint.

C. ANALYSIS OF ENERGY CONSUMPTION

Recently transformer architectures have significantly improved the performance of various natural language processing (NLP) tasks. Inspired by the original transformer [45], language models such as ELMo [46], BERT [20], GPT family [19] and BART [25] have emerged and produced state-of-the-art performance on various tasks. However, they require enormous amounts of data and compute resources for pre-training. This large computation consumes a lot of energy and has a high carbon footprint. It has an adverse financial and environmental impact [47], [48].

The expression to calculate carbon footprint C (in gram carbon dioxide equivalent or gCO_2e) as given in the equation 18 is taken from [48].

$$C = t \times (n_c \times P_c \times u_c + n_m \times P_m) \times PUE \times CI \times 0.001 \tag{18}$$

We modified Equation 18 to Equation 19:

$$C = t \times (n_c \times P_c \times u_c + n_{gpu} \times P_{gpu} \times u_{gpu} + n_m \times P_m) \times PUE \times CI \times 0.001 \tag{19}$$

where t is the running time (in hours), n_c is the number of cores, P_c is the power draw of a computing core, u_c is the core usage factor (between 0 to 1), n_{gpu} is the number of GPUs, P_{gpu} is the power drawn by the GPU, u_{gpu} is the GPU usage factor (between 0 to 1), P_m is the power draw of a memory unit (in watt). The power draw of memory is considered as 0.3725 W per GB [48], [49].

We trained all the models on Tesla P100-PCIE `Colab Pro+` that supports GPU. The efficiency coefficient of the data center is known as PUE (power usage effectiveness). Google uses ML to reduce its global yearly average PUE to 1.10 [50]. We use average worldwide value as carbon intensity (CI) of 475 gCO_2e KW/hour [51]. Gross CO_2 emission during training for T5 pre-trained model [24] was 46.7 tCO_2e [52], any transformer$_{\text{big}}$ model training required 192 $lbsCO_2e$ [47] and BERT base model with GPU required 1438 $lbsCO_2e$ [47]. We measure memory and compute power consumption and emission of CO_2 footprint using the WandB tool.[10] The quantitative results are shown in Table 8. In our proposed model, we require SciBERT embeddings of the input documents as input. So as a pre-processing step before model training, we encode the documents with SciBERT: this is a one-time operation and not repeated in every epoch. Table 8 clearly shows that our proposed model (third column) has fewer trainable parameters, and lower computational overhead and smaller carbon footprint per epoch than those of the other models. We have graphically compared the % of GPU utilization, % of CPU utilization, GPU Power usage, GPU memory allocated, memory used by process and required process CPU threads of the models over the training duration in Figure 3. The figure shows that GPU and CPU utilization, GPU power usage, and the process memory used by our proposed model are lower than those used in fine-tuning the large pre-trained summarization models. While our model

[6] https://huggingface.co/t5-base

[7] https://huggingface.co/sshleifer/distilbart-cnn-12-6

[8] https://huggingface.co/gpt2

[9] https://huggingface.co/microsoft/prophetnet-large-uncased-cnndm

[10] https://wandb.ai/site

TABLE 6. Evaluation of pointer-generator type models: scores for ROUGE, METEOR and BERTScore on MixSub dataset. The first row (where dataset is 'Full MixSub') indicates the performance when the models are trained on the whole MixSub training set and evaluated on the whole MixSub test set, without distinguishing between the subject categories of the papers. In the remaining part of the table, two cases are considered: Case 1: Trained on each subject-cluster of MixSub training set and evaluated on the corresponding test set; Case 2: Trained on the entire MixSub training set and evaluated on each subject-cluster of MixSub test set.

			ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BERTScore
Dataset	Input	Model Name	F1	F1	F1	Final score	F1
Full MixSub	abstract only	PGM	29.3	8.43	26.99	21.46	83.41
		PGM + Coverage	31.52	9.18	29.21	22.91	85.22
		PGM + SciBERT	30.44	9.68	27.81	23.38	84.83
		PGM + Coverage + SciBERT	**31.78**	**9.76**	**29.3**	**24**	**85.25**
Biological	Case 1	PGM	25.4	5.13	23.56	18.02	83.51
	Case 2	PGM	27.88	7.36	25.77	9.29	81.01
	Case 1	PGM + Coverage	28.23	6.18	25.96	19.60	83.99
	Case 2	PGM + Coverage	28.76	7.74	26.76	9.89	82.01
	Case 1	PGM + SciBERT	28.74	7.45	26.56	20.87	84.49
	Case 2	PGM + SciBERT	28.42	8.01	26.02	9.92	81.1
	Case 1	PGM + Coverage + SciBERT	**29.9**	7.6	**27.57**	**21.53**	**84.74**
	Case 2	PGM + Coverage + SciBERT	28.88	**8.03**	26.76	9.98	84.72
Chemistry	Case 1	PGM	27.83	7.39	26.1	13.58	82.99
	Case 2	PGM	27.44	7.15	25.55	9.63	81.83
	Case 1	PGM + Coverage	29.67	8.09	27.4	14.14	83.02
	Case 2	PGM + Coverage	29.68	7.73	27.57	9.88	82.11
	Case 1	PGM + SciBERT	30	8.58	27.98	22.38	84.93
	Case 2	PGM + SciBERT	29.16	8.47	26.96	9.85	81.92
	Case 1	PGM + Coverage + SciBERT	**31.4**	**8.9**	**29.33**	**24.37**	**85.11**
	Case 2	PGM + Coverage + SciBERT	30.41	8.58	28.15	10.19	82.19
Energy	Case 1	PGM	23.81	4.21	21.99	15.80	83.44
	Case 2	PGM	29.33	8.56	26.91	9.40	81.07
	Case 1	PGM + Coverage	27.12	4.71	24.98	18.11	84.05
	Case 2	PGM + Coverage	31.69	9.24	29.25	10.42	82.55
	Case 1	PGM + SciBERT	28.87	6.18	26.5	19.92	84.53
	Case 2	PGM + SciBERT	29.61	9.09	26.62	9.70	81.9
	Case 1	PGM + Coverage + SciBERT	30.04	6.84	27.4	**20.85**	**85.51**
	Case 2	PGM + Coverage + SciBERT	**32.15**	**9.66**	**29.7**	10.77	82.84
Management	Case 1	PGM	32.39	8.73	30.08	15.80	83.18
	Case 2	PGM	34.51	11.68	31.64	9.89	81.53
	Case 1	PGM + Coverage	34.47	9.54	31.77	18.23	83.67
	Case 2	PGM + Coverage	37.25	13.23	34.4	10.54	82.19
	Case 1	PGM + SciBERT	33.54	9.15	30.9	23.17	85.20
	Case 2	PGM + SciBERT	35.65	13.18	32.81	10.66	82.26
	Case 1	PGM + Coverage + SciBERT	36.05	11.02	33.27	**25.24**	**85.66**
	Case 2	PGM + Coverage + SciBERT	**38.39**	**13.62**	**35.64**	11.39	83.03
Nursing	Case 1	PGM	25.2	4.82	22.95	17.01	83.18
	Case 2	PGM	28.64	7.9	26.08	8.55	81.08
	Case 1	PGM + Coverage	28.2	5.46	25.5	18.79	83.71
	Case 2	PGM + Coverage	30.38	8.39	27.83	9.78	81.61
	Case 1	PGM + SciBERT	30.21	6.71	27.54	21.16	84.21
	Case 2	PGM + SciBERT	31.43	9.43	28.42	9.52	81.04
	Case 1	PGM + Coverage + SciBERT	**31.61**	8.09	28.7	**22.73**	**84.57**
	Case 2	PGM + Coverage + SciBERT	**31.61**	**10.28**	**28.83**	9.98	81.42
Physics	Case 1	PGM	29.97	7.98	27.44	21.19	84.19
	Case 2	PGM	30.41	9.07	28.07	10.21	81.05
	Case 1	PGM + Coverage	31.06	7.9	28.52	21.26	84.95
	Case 2	PGM + Coverage	32.05	10.26	30.27	10.51	82.4
	Case 1	PGM + SciBERT	30.67	8.3	28.45	21.76	85.02
	Case 2	PGM + SciBERT	31.31	10.67	28.6	10.38	81.93
	Case 1	PGM + Coverage + SciBERT	32.13	8.92	29.53	**22.83**	**85.25**
	Case 2	PGM + Coverage + SciBERT	**32.99**	**11.01**	**30.35**	11.16	82.45
Social Science	Case 1	PGM	22.64	4.36	20.51	13.94	81.86
	Case 2	PGM	30.23	10.39	27.29	9.23	81.64
	Case 1	PGM + Coverage	26.96	5.17	24.19	16.08	82.72
	Case 2	PGM + Coverage	31.99	11.2	28.82	9.61	81.69
	Case 1	PGM + SciBERT	30.11	8	26.87	19.02	83.35
	Case 2	PGM + SciBERT	31.21	10.97	27.75	9.63	81.65
	Case 1	PGM + Coverage + SciBERT	31.89	8.67	28.36	**19.99**	**83.81**
	Case 2	PGM + Coverage + SciBERT	**32.35**	**11.8**	**28.87**	9.79	81.75

consumes a large memory for a short time, the other models typically have a larger memory consumption that remains steady for a longer duration. Our model exploits more CPU threads than GPT-2 but fewer threads than other compared

TABLE 7. Performance of fine-tuned versions of pre-trained models on CSPubSum dataset using abstracts of the papers as the input. The highest performance scores are marked with bold.

Model Name	ROUGE-1 F1	ROUGE-2 F1	ROUGE-L F1	METEOR Final score	BERTScore F1
T5-base	**40.03**	**16.27**	**37.64**	36.33	**86.80**
Distilbart-CNN-12-6	39.95	16.13	37.16	**38.99**	86.69
GPT-2	33.12	11.76	30.64	33.14	85.30
ProphetNet-large-uncased-cnndm	23.95	0.96	20.38	15.3	81.41
PGM + Coverage + SciBERT (ours)	38.26	14.26	35.51	32.62	86.65

TABLE 8. Power consumption, compute expenditure, and CO_2 emission statistics for summarization models.

Factors	Sub-Factor	PGM + Coverage + SciBERT	ProphetNet-large-uncased-cnndm	GPT-2	Distilbart-CNN-12-6	T5-base
Total trainable parameters		21.5M	391M	117M	305M	220M
Colab Notebook	Avail. RAM: 51GB	2.86GB	4.07GB	3.20GB	6.47GB	3.89GB
	Avail. GPU: 16GB	1.14GB	15.57GB	13.51GB	14.64GB	2.92GB
	Avail. Disk: 166.83GB	48.30GB	41.53GB	40.64GB	41.21GB	44.57GB
Power consumed	Max. GPU power: 250W	116W	189W	187W	172W	206W
	Max. CPU power: 95W	95W	95W	95W	95W	95W
% of GPU utilization		75%	100%	97%	100%	97%
% of GPU memory allocated		55%	99%	89%	92%	60%
% of CPU utilization		35%	16%	27%	19%	17%
Used process CPU threads		54	57	38	55	66
Process memory in use (GB)		0.834	22.28	3.84	19.18	12.42
Time for one epoch (mins)		5.17	31	22	19	15
Epoch-wise carbon footprint (gms/epoch)		5.56	56.72	40.68	32.35	28.93

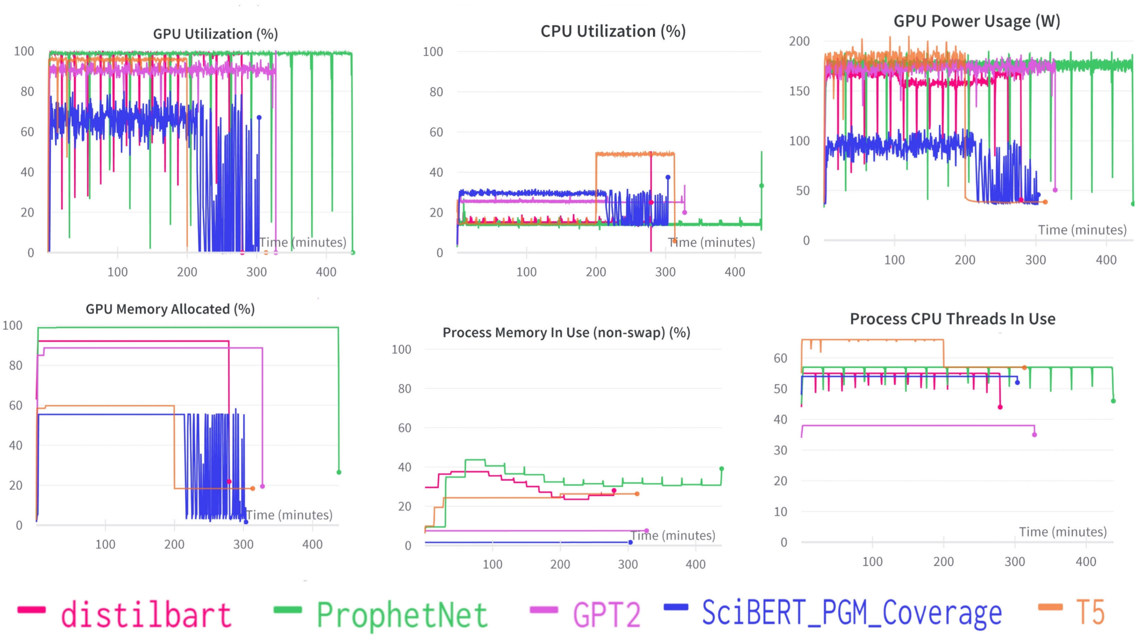


FIGURE 3. Comparison of compute resources used by summarization models.

models. We believe that researchers should give attention to energy-friendly models and algorithms rather than only to performance metrics. In this context, our model is a better alternative to large pre-trained transformers.

VI. CASE STUDIES

A. CASE STUDY ON CSPubSum DATASET

We now present a few examples demonstrating the outputs produced by the pointer-generator type models used in this

Author-written research highlights: Presenting a new two stage meta heuristic clustering algorithm based on general type 2 fuzzy sets. Incorporating a new similarity based objective function using alpha plane representation of general type 2 fuzzy sets. Implementing the proposed approach on real microarray gene expression datasets.

PGM: We propose a new two stage meta heuristic algorithm for expression data clustering. New objective function for general type 2 fuzzy means is proposed. Output is inserted to the first stage until no better solution is obtained. Proposed approach is evaluated using three microarray gene expression datasets. Experiments verify the performance of the proposed approach.

PGM + Coverage: A novel meta heuristic methodology for microarray gene expression data clustering is proposed. Two stage inspired the two stage meta heuristic optimization framework simulated annealing is proposed. Philosophy of the meta heuristic framework is devoted to the annealing process accompanied by its proposed perturbation mechanisms. Experiments were conducted using three microarray gene expression datasets. Experiments demonstrate the capabilities of the proposed approach.

PGM + SciBERT: A new two stage meta heuristic algorithm is proposed. A new objective function using planes for general type 2 fuzzy sets is presented. A new objective function using planes for general type 2 fuzzy C means clustering algorithm is proposed. The proposed approach has been evaluated using several synthesized datasets and three microarray gene expression datasets. The proposed approach is evaluated using several synthesized datasets and three microarray gene expression datasets .

PGM + Coverage + SciBERT: A powerful data clustering approach based on the concept of planes in general type 2 fuzzy sets is presented. A new objective function using planes for general type 2 fuzzy c means clustering algorithm is proposed. The proposed approach has been evaluated using three microarray gene expression datasets.

FIGURE 4. Input is only an abstract from CSPubSum dataset. Highlights produced by the four models are shown. Input and author-written research highlights taken from `https://www.sciencedirect.com/science/article/pii/S0010482514001565`.

Author-written research highlights: Presenting a new two stage meta heuristic clustering algorithm based on general type 2 fuzzy sets. Incorporating a new similarity based objective function using alpha plane representation of general type 2 fuzzy sets. Implementing the proposed approach on real microarray gene expression datasets.

PGM: we propose a two stage sa clustering algorithm based on the concept of type 2 fcm and fcm [UNK] utilizes a two stage sa clustering algorithm based on the new simulated annealing. Present the first hybrid meta heuristic algorithm to solve the problem. Present the first stage solution approach to solve the problem.

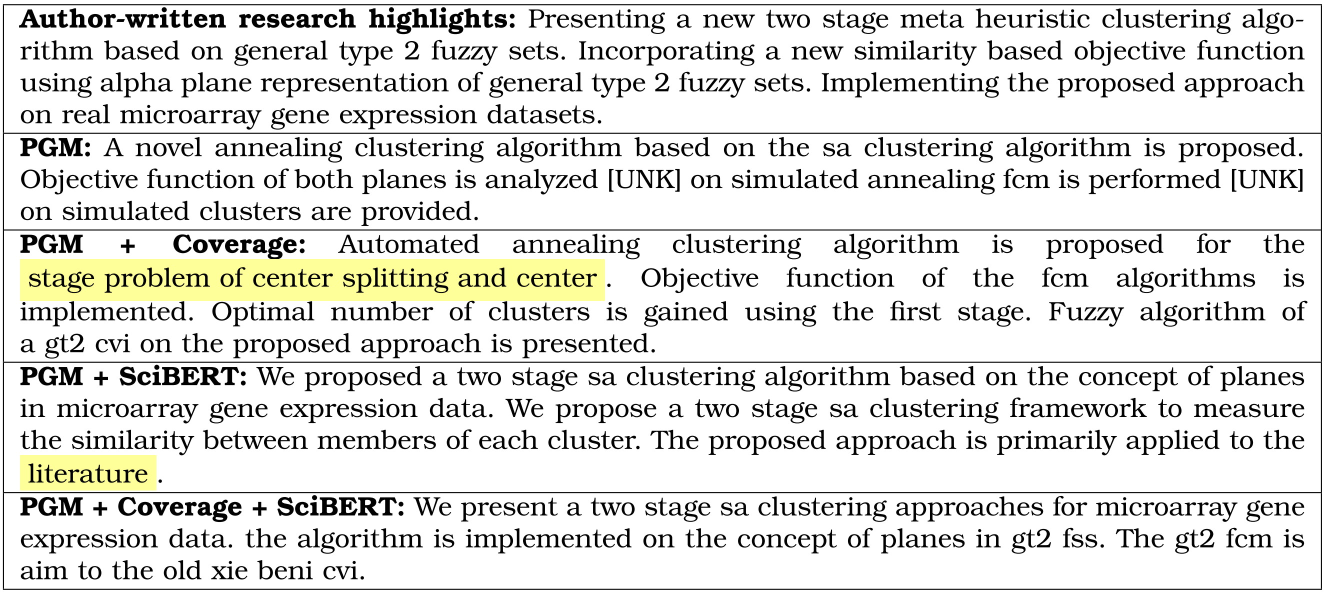
PGM + Coverage: We present a two stage sa clustering algorithm based on the concept of planes in gt2 fss. Utilizes a two stage solution based on the new simulated annealing technology. Mechanism performs very complex times using the proposed framework. Results have been shown to be suitable for other systems.

PGM + SciBERT: A two stage sa clustering algorithm is proposed based on the concept of planes in gt2 fss. The objective function of the gt2 fuzzy sa will be performed. Quality of the proposed approach is analyzed using a gt2 fuzzy cvi fuzzy cvi . The proposed approach is one of the first stage of the proposed approach .

PGM + Coverage + SciBERT: A two stage sa clustering algorithm based on planes in gt2 fss is proposed. Only the optimal number of clusters are obtained. The mechanism of gt2 fuzzy sa will be used to detect the optimal number of clusters. The objective function of simulated annealing and fcm is further analyzed.

FIGURE 5. Input is (abstract + conclusion) from CSPubSum dataset. Highlights produced by the four models are shown. Input and author-written research highlights taken from `https://www.sciencedirect.com/science/article/pii/S0010482514001565`.

paper. In all the case studies reported below, *yellow* color represents factual errors and *orange* shows repeating words . Figure 4 illustrates the highlights produced by the four models when the input is only the abstract. Note that the vanilla pointer-generator network misses or incorrectly uses some keywords while generating the highlights. For example,



Author-written research highlights: Presenting a new two stage meta heuristic clustering algorithm based on general type 2 fuzzy sets. Incorporating a new similarity based objective function using alpha plane representation of general type 2 fuzzy sets. Implementing the proposed approach on real microarray gene expression datasets.

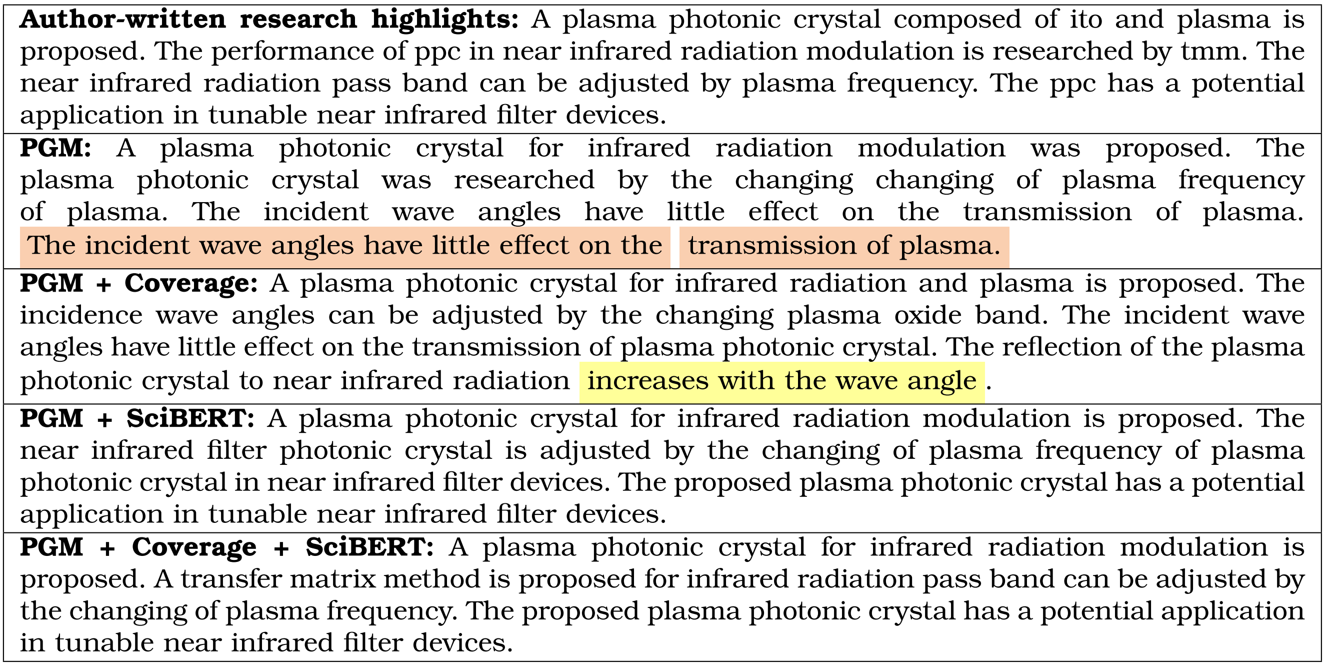
PGM: A novel annealing clustering algorithm based on the sa clustering algorithm is proposed. Objective function of both planes is analyzed [UNK] on simulated annealing fcm is performed [UNK] on simulated clusters are provided.

PGM + Coverage: Automated annealing clustering algorithm is proposed for the stage problem of center splitting and center . Objective function of the fcm algorithms is implemented. Optimal number of clusters is gained using the first stage. Fuzzy algorithm of a gt2 cvi on the proposed approach is presented.

PGM + SciBERT: We proposed a two stage sa clustering algorithm based on the concept of planes in microarray gene expression data. We propose a two stage sa clustering framework to measure the similarity between members of each cluster. The proposed approach is primarily applied to the literature .

PGM + Coverage + SciBERT: We present a two stage sa clustering approaches for microarray gene expression data. the algorithm is implemented on the concept of planes in gt2 fss. The gt2 fcm is aim to the old xie beni cvi.

FIGURE 6. Input is (introduction + conclusion) from CSPubSum dataset. Highlights produced by the four models are shown. Input and author-written research highlights taken from `https://www.sciencedirect.com/science/article/pii/S0010482514001565`.



Author-written research highlights: A plasma photonic crystal composed of ito and plasma is proposed. The performance of ppc in near infrared radiation modulation is researched by tmm. The near infrared radiation pass band can be adjusted by plasma frequency. The ppc has a potential application in tunable near infrared filter devices.

PGM: A plasma photonic crystal for infrared radiation modulation was proposed. The plasma photonic crystal was researched by the changing changing of plasma frequency of plasma. The incident wave angles have little effect on the transmission of plasma. The incident wave angles have little effect on the transmission of plasma.

PGM + Coverage: A plasma photonic crystal for infrared radiation and plasma is proposed. The incidence wave angles can be adjusted by the changing plasma oxide band. The incident wave angles have little effect on the transmission of plasma photonic crystal. The reflection of the plasma photonic crystal to near infrared radiation increases with the wave angle .

PGM + SciBERT: A plasma photonic crystal for infrared radiation modulation is proposed. The near infrared filter photonic crystal is adjusted by the changing of plasma frequency of plasma photonic crystal in near infrared filter devices. The proposed plasma photonic crystal has a potential application in tunable near infrared filter devices.

PGM + Coverage + SciBERT: A plasma photonic crystal for infrared radiation modulation is proposed. A transfer matrix method is proposed for infrared radiation pass band can be adjusted by the changing of plasma frequency. The proposed plasma photonic crystal has a potential application in tunable near infrared filter devices.

FIGURE 7. Input is only the abstract of an article from the MixSub dataset. Highlights produced by the four models are shown. Input and author-written research highlights taken from `https://www.sciencedirect.com/science/article/pii/S1567173920301292`.

it produces "…algorithm for expression data clustering." instead of "…algorithm for microarray gene expression data clustering.", and "…type 2 fuzzy means" instead of "…type 2 fuzzy c-means". While use of SciBERT corrects these issues, unnecessarily repeated words are seen when coverage mechanism is absent. The output produced by the (**PGM** + **Coverage** + **SciBERT**) model is closest to the author-written highlights.

Figure 5 and Figure 6 depict the highlights produced by the models when the input is (abstract + conclusion) and (introduction + conclusion), respectively. We observe that the highlights produced by all the models for the last four input types contain a number of acronyms like 'fcm' (fuzzy C-means), 'gt2' (general type 2), 'fss' (fuzzy sets), and 'cvi' (cluster validity index) which occur frequently in the introduction and conclusion of the paper. Since the abstract typically does not contain acronyms, highlights generated using it are also generally free of acronyms.

B. CASE STUDY ON MixSub DATASET

We now present an example demonstrating the output produced by the four variants of pointer-generator model for the MixSub dataset. Figure 7 displays the outputs when the models take only the abstract as the input. We observe that the last two sentences produced by the plain pointer-generator model are identical. This issue goes away when coverage is added but now the model's output is not entirely factually correct: it generates "The reflection of the plasma photonic crystal to near infrared radiation *increases* with the wave angle." while the abstract mentions "The reflection of the plasma photonic crystal to near infrared radiation *decreases* with increasing of the incident wave angle ...". This factual error disappears when SciBERT is added. The output quality improves further when coverage is added. While PGM with SciBERT (but without coverage) outputs "the near infrared filter photonic crystal is adjusted", the final model (PGM + coverage + SciBERT) is more precise: "*infrared radiation pass band* can be adjusted". Therefore, the highlights produced by **(PGM + Coverage + SciBERT)** seem to be most satisfactory though its last sentence has a syntax error due to a missing relative pronoun: "A transfer method is proposed for infrared radiation pass band [which] can be adjusted ...".

VII. CONCLUSION AND FUTURE WORK

We applied four different deep neural models to generate research highlights from a research paper. We experimented with different input types for each model for one of the datasets while we used only the abstract as input for the other dataset. The pointer-generator model with SciBERT and coverage mechanism achieved the best performance in each case. But the predicted research highlights are not yet perfect in terms of syntax and semantics. We are currently exploring other techniques to address these issues. A few other research directions would be to generate highlights that summarize a set of related papers, and to build a database containing research findings from different papers with links connecting semantically-related findings.

DECLARATIONS

AVAILABILITY OF SUPPORTING DATA

The dataset CSPubSum is publicly available.[11] The MixSub dataset used in the article has been publicly released by the authors on github.[12]

DECLARATION OF COMPETING INTERESTS

The authors declare that they have no competing interests.

[11] https://github.com/EdCo95/scientific-paper-summarisation/tree/master/DataDownloader

[12] https://github.com/tohidarehman/Highlights-Generation-ResearchPaper

REFERENCES

[1] L. Bornmann, R. Haunschild, and R. Mutz, "Growth rates of modern science: A latent piecewise growth curve approach to model publication numbers from established and new literature databases," *Humanities Social Sci. Commun.*, vol. 8, no. 1, pp. 1–15, Oct. 2021.

[2] R. Van Noorden, "Global scientific output doubles every nine years," Nature News Blog, 2014. [Online]. Available: https://blogs.nature.com/news/2014/05/global-scientific-output-doubles-every-nine-years.html

[3] H. P. Luhn, "The automatic creation of literature abstracts," *IBM J. Res. Develop.*, vol. 2, no. 2, pp. 159–165, Apr. 1958.

[4] M. F. Mridha, A. A. Lima, K. Nur, S. C. Das, M. Hasan, and M. M. Kabir, "A survey of automatic text summarization: Progress, process and challenges," *IEEE Access*, vol. 9, pp. 156043–156070, 2021.

[5] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proc. 18th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 1995, pp. 68–73.

[6] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113679.

[7] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1073–1083.

[8] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Hong Kong: Association for Computational Linguistics, 2019, pp. 3615–3620.

[9] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81.

[10] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," *Proc. ACL Workshop Intrinsic Extrinsic Eval. Measures Mach. Transl. Summarization*, 2005, pp. 65–72.

[11] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–43.

[12] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 3104–3112.

[13] A. A. Syed, F. L. Gaol, and T. Matsuo, "A survey of the state-of-the-art models in neural abstractive text summarization," *IEEE Access*, vol. 9, pp. 13248–13265, 2021.

[14] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

[15] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 93–98.

[16] R. Nallapati, B. Zhou, C. D. Santos, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.* Berlin, Germany: Association for Computational Linguistics, 2016, pp. 280–290.

[17] D. T. Anh and N. T. T. Trang, "Abstractive text summarization using pointer-generator networks with pre-trained word embedding," in *Proc. 10th Int. Symp. Inf. Commun. Technol.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 473–478.

[18] Y. Du and H. Huo, "News text summarization based on multi-feature and fuzzy logic," *IEEE Access*, vol. 8, pp. 140261–140272, 2020.

[19] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI Blog, 2018. [Online]. Available: https://openai.com/blog/language-unsupervised/

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.* Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[21] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics* Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 1441–1451.

[22] A. Sahu and S. G. Sanjeevi, "Better fine-tuning with extracted important sentences for abstractive summarization," in *Proc. Int. Conf. Commun., Control Inf. Sci. (ICCISc)*, Jun. 2021, pp. 11328–11339.

[23] D. Aksenov, J. Moreno-Schneider, P. Bourgonje, R. Schwarzenberg, L. Hennig, and G. Rehm, "Abstractive text summarization based on language model conditioning and locality modeling," in *Proc. 12th Lang. Resour. Eval. Conf.* Marseille, France: European Language Resources Association, May 2020, pp. 6680–6689.

[24] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.

[25] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 7871–7880.

[26] S. Teufel and M. Moens, "Summarizing scientific articles: Experiments with relevance and rhetorical status," *Comput. Linguistics*, vol. 28, no. 4, pp. 409–445, Dec. 2002.

[27] W. T. Visser and M. B. Wieling. (2007). *Sentence-Based Summarization of Scientific Documents: The Design and Implementation of an Automatic Summarizer*. [Online]. Available: http://www.martijnwieling.nl/files/wielingvisser05automaticsummarization.pdf

[28] E. Lloret, M. T. Romá-Ferri, and M. Palomar, "COMPENDIUM: A text summarization system for generating abstracts of research papers," *Data Knowl. Eng.*, vol. 88, pp. 164–175, Nov. 2013.

[29] M. Kim, D. S. Moirangthem, and M. Lee, "Towards abstraction from extraction: Multiple timescale gated recurrent unit for summarization," in *Proc. 1st Workshop Represent. Learn. NLP*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 70–77.

[30] C. M. Souza, M. R. G. Meireles, and R. Vimieiro, "A multi-view extractive text summarization approach for long scientific articles," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–8.

[31] N. I. Nikolov, M. Pfeiffer, and R. H. R. Hahnloser, "Data-driven summarization of scientific articles," *Proc. 11th Int. Conf. Lang. Resour. Eval. (LREC)*. Paris, France: European Language Resources Association (ELRA), May 2018, pp. 1–8.

[32] T. Y. S. S. Santosh, D. K. Sanyal, P. K. Bhowmick, and P. P. Das, "DAKE: Document-level attention for keyphrase extraction," in *Proc. Eur. Conf. Inf. Retr. (ECIR)*. Cham, Switzerland: Springer, 2020, pp. 392–401.

[33] T. Y. S. S. Santosh, D. K. Sanyal, P. K. Bhowmick, and P. P. Das, "SaSAKE: Syntax and semantics aware keyphrase extraction from research papers," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 5372–5383.

[34] T. Y. S. S. Santosh, D. K. Sanyal, P. K. Bhowmick, and P. P. Das, "Gazetteer-guided keyphrase generation from research papers," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining (PAKDD)*. Cham, Switzerland: Springer, 2021, pp. 655–667.

[35] T. Y. S. S. Santosh, N. R. Varimalla, A. Vallabhajosyula, D. K. Sanyal, and P. P. Das, "HiCoVA: Hierarchical conditional variational autoencoder for keyphrase generation," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 3448–3452.

[36] E. Collins, I. Augenstein, and S. Riedel, "A supervised approach to extractive summarisation of scientific papers," in *Proc. 21st Conf. Comput. Natural Lang. Learn. (CoNLL)*. Vancouver, BC, Canada: Association for Computational Linguistics, 2017, pp. 195–205.

[37] L. Cagliero and M. La Quatra, "Extracting highlights of scientific articles: A supervised summarization approach," *Expert Syst. Appl.*, vol. 160, Dec. 2020, Art. no. 113659.

[38] T. Rehman, D. K. Sanyal, S. Chattopadhyay, P. K. Bhowmick, and P. P. Das, "Automatic generation of research highlights from scientific abstracts," in *Proc. 2nd Workshop Extraction Eval. Knowl. Entities Sci. Documents (EEKE), JCDL, CEUR, Workshop*, 2021, pp. 69–70.

[39] T. Rehman, D. K. Sanyal, P. Majumder, and S. Chattopadhyay, "Named entity recognition based automatic generation of research highlights," in *Proc. 3rd Workshop Scholarly Document Process. (SDP), COLING*. Gyeongju, South Korea: Association for Computational Linguistics, Oct. 2022, pp. 163–169.

[40] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 76–85.

[41] W. Ammar et al., "Construction of the literature graph in semantic scholar," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.* New Orleans, LA, USA: Association for Computational Linguistics, 2018, pp. 84–91.

[42] A. R. Fabbri, W. Kryściński, B. McCann, C. Xiong, R. Socher, and D. Radev, "SummEval: Re-evaluating summarization evaluation," *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 391–409, Apr. 2021.

[43] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. 14th Int. Joint Conf. Artif. Intell. (IJCAI)*. San Francisco, CA, USA: Morgan Kaufmann, 1995, pp. 1137–1143.

[44] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning," 2018, *arXiv:1811.12808*.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 6000–6010.

[46] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.* New Orleans, LA, USA: Association for Computational Linguistics, 2018, pp. 2227–2237.

[47] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3645–3650.

[48] L. Lannelongue, J. Grealey, and M. Inouye, "Green algorithms: Quantifying the carbon footprint of computation," *Adv. Sci.*, vol. 8, no. 12, Jun. 2021, Art. no. 2100707.

[49] A. Karyakin and K. Salem, "An analysis of memory power consumption in database systems," in *Proc. 13th Int. Workshop Data Manage. New Hardw.*, May 2017, pp. 1–9.

[50] Google. (2021). *Efficiency—Data Centers*. [Online]. Available: https://www.google.com/about/datacenters/efficiency

[51] IEA. (2019). *Global Energy & CO$_2$ Status Report 2019*. [Online]. Available: https://www.iea.org/reports/global-energy-co2-status-report-2019

[52] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, "Carbon emissions and large neural network training," 2021, *arXiv:2104.10350*.

TOHIDA REHMAN is an Assistant Professor with the Department of Information Technology, Jadavpur University, India. From 2014 to 2018, she was an Assistant Professor with the Department of Computer Science, Surendranath College, Calcutta University, India. She has more than nine years of teaching experience. Her current research interests include machine learning and natural language processing. Her current work focuses on improving text summarization.

DEBARSHI KUMAR SANYAL received the B.E. degree in information technology and the Ph.D. degree in engineering from Jadavpur University, Kolkata, in 2005 and 2012, respectively. He is an Assistant Professor with the School of Mathematical and Computational Sciences, Indian Association for the Cultivation of Science, Kolkata, India. He was with IIT Kharagpur, KIIT Deemed University, Xilinx India Pvt. Ltd., Interra Systems India Pvt. Ltd., and Infosys Ltd. His current research interests include natural language processing, digital library technologies, information retrieval, and machine learning.

SAMIRAN CHATTOPADHYAY is a Pro Vice Chancellor in Techno India University, West Bengal, and a former professor with the Department of Information Technology, Jadavpur University. He works in the areas of machine intelligence and its applications, wireless networks, and network security. His current work mostly centers around the design and creation of computer models for decision analysis and optimization, particularly in the fields of HCI, high throughput wireless networks, healthcare, power engineering, and technology-enabled learning. He has more than three decades of experience in teaching and research in the broad domain of computer science and engineering (CSE). He is the author of about 180 articles in reputed journals and conference proceedings, including more than 60 high impact journal publications.

PLABAN KUMAR BHOWMICK received the M.S. and Ph.D. degrees from the Department of Computer Science and Engineering, IIT Kharagpur. He is an Assistant Professor with the G. S. Sanyal School of Telecommunications, Centre of Excellence in Artificial Intelligence, Indian Institute of Technology Kharagpur. His research interests include automated answer grading, augmenting learner experience, and graph machine learning.

PARTHA PRATIM DAS (Member, IEEE) received the B.Tech., M.Tech., and Ph.D. degrees from the Department of Electronics and Electrical Communication, IIT Kharagpur, in 1984, 1985, and, 1988 respectively.

He is a Visiting Professor with the Department of Computer Science, Ashoka University. He is on leave from IIT Kharagpur, where he was a Professor with the Department of Computer Science and Engineering. He has over 22 years of experience in teaching and research with IIT Kharagpur, and about 13 years of experience in software industry, including start-ups. Over the past ten years, he has led the Development of National Digital Library of India (NDLI) Project, MoE, GoI, as a Joint Principal Investigator. He has also developed a unique vertical DEEPAK: Disability Education and Engagement Portal for Access to Knowledge. He has also led engineering entrepreneurship education, research, facilitation, and deployment with IIT Kharagpur, from 2013 to 2020. He is a strong proponent of online education. He has been a Key Instructor of three courses with SWAYAM-NPTEL, since 2016. During pandemic, he has instrumental in making learning material available to the students through NDLI. He currently works on the following problems: hands-free control and immersive navigation of Chandrayaan and Mangalyaan images on large displays (with ISRO), smart knowledge transfer for legacy software projects, automated interpretation of Bharatanatyam dance, and the development of Indian food atlas and food knowledge graph.

Dr. Das received the Young Scientist/Engineer Award from the Indian National Science Academy, in 1990, the Indian National Academy of Engineering, in 1996, and the Indian Academy of Sciences, in 1992. Being in the leadership team of NDLI, he was recognized for his contributions toward online education during pandemic through several awards, including the OE Awards for Excellence: Open Resilience, in 2020; the SM4E Award: Innovation@COVID-19, in 2021; and the World Summit Award: Learning and Education, in 2021.