SCHOLARMATE: A Multi-Model Approach to Automated Literature Reviews using AI

1st Zaema Dar
Department of Computer Science
National University of Computing and Emerging Sciences
Islamabad, Pakistan
zaema.khurram@gmail.com

2nd Muhammad Raheel
Department of Computer Science
National University of Computing and Emerging Sciences
Islamabad, Pakistan
raheelmuhd3@gmail.com

3rd Usman Bokhari
Department of Computer Science
National University of Computing and Emerging Sciences
Islamabad, Pakistan
usman.ali.bokhari@outlook.com

4th Akhtar Jamil
Department of Computer Science
National University of Computing and Emerging Sciences
Islamabad, Pakistan
akhtar.jamil@nu.edu.pk

5th Alaa Ali Hameed
Department of Computer Engineering
Istinye University
Istanbul, Tureky
alaa.hameed@istinye.edu.tr

***Abstract*—This research presents a novel approach to summarizing complex AI and deep learning scientific papers by integrating and fine-tuning advanced models such as LED_Large, Pegasus variants, and BART. By experimenting with unique combinations like SciBERT x LED_Large, we aimed to capture essential details typically missed by conventional methods, particularly in methodologies and results sections. Our multi-model strategy showed significant improvements in summarization accuracy and depth, as evidenced by metrics like cosine similarity. Models like LED_Large exhibited rapid adaptation to training data with impressive semantic understanding, even with fewer epochs. Our findings highlight LED_Large's potential for further performance gains with more training. The results, underscored by metrics such as a FRES score of 28.5852 and ROUGE scores with a ROUGE-1 F1-Score of 0.4991, indicate substantial enhancements in both the accuracy and depth of summaries. In contrast, extensively trained models like BART_large and Pegasus showed strong semantic capabilities but revealed areas for refinement, particularly in enhancing readability and higher-order n-gram overlap in summaries. This research underscores the importance of model selection and training depth in AI-driven scientific summarization.**

***Index Terms*—Natural Language Processing, Scientific Summarization, Transformers, LED_Large, Pegasus-Large, BART, SciBERT, Literature Review Generation, Cosine Similarity, Deep Learning.**

I. Introduction

The rapid proliferation of academic research, especially in fields like artificial intelligence (AI) and deep learning, has created a substantial demand for effective summarization tools. These tools are vital for researchers to efficiently assimilate and categorize the vast influx of scholarly articles. Traditional summarization models, however, often fall short in capturing the intricate essence of scientific papers, especially those in complex domains. They tend to omit key elements such as detailed methodologies, metrics, mathematical computations, and comprehensive interpretations of results. This deficiency in existing models underscores the need for a more sophisticated summarization approach that can deliver both concise and content-rich summaries, encapsulating the full breadth and depth of scientific research.

In this paper, we address existing challenges. Our contributions are as follows:

1) We tested a range of models, including LED_Large [1], Pegasus-Large [2], BART [3], BERT [4], and SciBERT [5], and combinations of these models, such as SciBERT x Pegasus-Large and SciBERT x LED_Large, to leverage their complementary strengths.
2) We evaluated their effectiveness through metrics like cosine similarity.
3) For the generation of literature reviews, we employed models like facebook/bart-large-mnli [6], allenai/scibert_scivocab_uncased [7], and LED_Large, innovatively combining them to enhance the extraction and summarization of relevant content.

The rest of the paper is organised as follows: Section II presents an in-depth review of existing summarization models and their limitations. Section III delves into the extraction and collection of AI research papers. Section IV details the architecture and methodology of our proposed models, including

their unique combinations. Section V discusses the evaluation methods used to assess model performance, followed by, which analyzes the results and their implications for the field of scientific summarization. Finally, Section VI concludes the paper, summarizing our key findings and suggesting directions for future research.

II. Related Work

Initially, Sutskever et al. [8] , which laid the groundwork for modern NLP applications, we see the initial steps towards understanding complex document structures. The advent of BERT by Devlin et al. [4] marked a significant leap, introducing deep bidirectional transformers that greatly enhanced the ability to grasp the nuanced context of texts, setting a new standard for both extractive and abstractive summarization techniques.

Cohan et al. [9] further specialized this approach to long documents, introducing a discourse-aware attention model that better captured the intricate structure of academic texts. Following this, Beltagy et al. [5] tailored BERT's capabilities to the scientific domain with SciBERT, optimizing the model's performance for scientific literature.

The year 2019 also saw Liu fine-tune BERT [10] specifically for extractive summarization, demonstrating the model's adaptability to different summarization tasks. Zhang et al. [11] then introduced PEGASUS, which innovated with gap-sentence generation pre-training, significantly improving abstractive summarization's coherence and context sensitivity.

The Longformer, introduced by Beltagy, Peters, and Cohan [12], addressed the challenge of processing lengthy documents by employing a scalable attention mechanism, making it particularly suitable for summarizing extensive academic papers. This was complemented by the introduction of the LongT5 by Guo et al. [13], which further pushed the boundaries for handling long texts in NLP tasks.

Kryściński et al. [14]contributed the BookSum dataset, offering a unique challenge in summarizing long-form narratives, akin to the complexities found in detailed research papers. The same year, Liu et al. [15] introduced a hybrid model that skillfully combined extractive and abstractive summarization techniques, offering a more nuanced approach to summarization.

In 2022, advancements continued with Pang et al. introducing a novel method for long document summarization that integrates top-down and bottom-up inference, providing a comprehensive understanding of extended texts. Zhao et al. [16] presented SLiC, a method for calibrating sequence likelihood in language models, aiming to enhance the quality of generated summaries.

Gera et al. [17] explored zero-shot text classification with self-training, showcasing the potential of pre-trained models to adapt to specific summarization tasks without extensive domain-specific training. This adaptability is crucial for categorizing and summarizing research papers efficiently.

Yuan et al. [18] introduced BARTSCORE, offering a nuanced metric for evaluating text generation, focusing on aspects like informativeness and factuality, which are essential for assessing the quality of summaries and literature reviews.

In 2023, several significant contributions were made, including the "Web-based Pretrained Transformer Model for Scientific Paper Summarization (WPT-SPS)" by Girthana et al. [19], which applied transfer learning techniques to generate abstractive summaries, showing considerable improvements over existing models like BART and Longformers.

Badhe et al. in his paper, Synopsis Creation for Research Paper using Text Summarization Models [20] compared BERT, BART, and T5, highlighting T5's suitability for summarizing single research papers, addressing the need for more informative summaries than traditional abstracts provide.

Additionally, DeepSumm [21] by Joshi et al. proposed a novel extractive summarization method that leverages topic modeling and sequence networks to capture both the global and local semantic information of documents, showing superior performance on benchmark datasets.

Extractive Summarization via ChatGPT for Faithful Summary Generation" [22] by Zhang, Liu, and Zhang explored ChatGPT's capabilities in extractive summarization, revealing its potential when combined with an extract-then-generate pipeline to enhance summary faithfulness.

These developments, from early sequence-to-sequence models to the latest innovations in transformer-based models and large language models, illustrate the dynamic and rapidly evolving landscape of text summarization. Each contribution builds on the last, pushing the field toward more accurate, coherent, and context-aware summarization tools, crucial for synthesizing the ever-growing body of academic literature into concise, informative summaries and literature reviews.

III. Data Set Collection

In our study on literature review generation, we streamlined data preprocessing by querying the ArXiv API to fetch research paper abstracts using titles or IDs. This involved extracting paper IDs from PDF filenames, accommodating various naming conventions, and compiling abstracts into a structured dataset, which was then refined and saved in CSV format for accessibility.

Further, we extracted the main content from each PDF, excluding references, using a specialized library. The content was linked to filenames in a separate CSV, which was then merged with the abstract dataset, resulting in a comprehensive collection of research materials. This consolidated dataset, containing 787 AI-related papers and their abstracts, laid the groundwork for our literature review analysis.

IV. Methodology

Our proposed methodology for literature review generation, illustrated in the block diagram, aims to surmount the token limit constraints of models like GPT, ensuring richer, more comprehensive summaries.

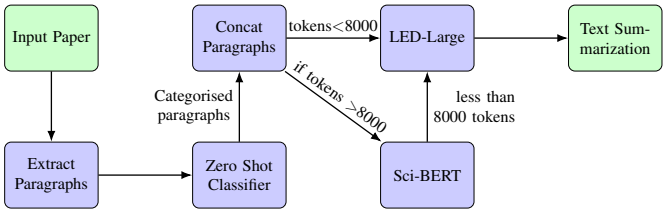

Fig. 1: Block diagram of the Proposed Approach

A. Research Paper Extraction

In the first step of our literature review generation process, we delve into the crucial task of obtaining and extracting research papers in PDF format from users. To acquire the source material, we have implemented a user-friendly mechanism that allows researchers and scholars to upload their PDF papers effortlessly. Once the PDF papers are submitted, we employ the pdfplumber library to extract the contents systematically. One key challenge in this step is to ensure that the extracted content is focused solely on the core substance of the paper, excluding any references or bibliographic information. This precision is vital as it enables us to work with the most relevant content for generating literature reviews, eliminating extraneous data that might dilute the final summaries.

B. Splitting into Paragraphs

In the second stage, we focus on the crucial task of formatting the research paper into a structured list of paragraphs. To achieve this, we employ a series of formatting techniques and utilize specialized libraries tailored for efficient text processing. The initial step involves parsing the extracted content into coherent blocks of text, ensuring that each paragraph is logically segmented based on the content's thematic structure.

C. Zero-Shot Classifier - facebook/bart_large_mnli

In the third phase, we employ a zero-shot text classifier, specifically the "facebook/bart_large_mnli" model, to categorize each of the previously structured paragraphs. This zero-shot classifier utilizes a novel technique based on Natural Language Inference (NLI) to perform text classification without the need for explicit training on labeled data. The underlying principle, as proposed by Yin et al. , involves treating the sequence to be classified as the NLI premise and generating a hypothesis from each candidate label. For instance, if we aim to determine whether a paragraph belongs to the category "Methodology and Dataset," the classifier constructs a hypothesis like "This text is about methodology and dataset." The probabilities of entailment and contradiction between the premise and hypothesis are then converted into label probabilities, allowing us to assign a label to each paragraph based on the highest probability.

This approach has proven to be remarkably effective, particularly when employed with larger pre-trained models like BART. It enables us to categorize paragraphs into meaningful labels, such as "Methodology and Dataset," "Evaluation Metrics and Accuracy Obtained," "Critical Analysis," "Relation to Proposed Work," or "None." To ensure comprehensive coverage, we assign a paragraph to one or more labels depending on the classification scores it receives. This adaptive approach minimizes the likelihood of empty labels, ensuring that valuable information is captured even when a paragraph pertains to multiple aspects of the literature review.

D. Concatenation within Each Label

In the fourth stage, we focus on the concatenation of paragraphs within each label. After categorizing paragraphs into meaningful labels using the zero-shot classifier, we assemble the content within each category, ensuring that each label contains substantial amount of text.

E. Checking Token Limit- SciBERT

In the subsequent phase, we encounter the crucial task of managing token limits for each label's content. Given the inherent constraints imposed by token limits, we ensure that the accumulated text within each category does not exceed 8000 tokens, a limit commonly encountered in various text processing tasks. To achieve this, we leverage the power of SciBERT, a specialized language model designed to cater to the unique challenges posed by scientific texts.

When confronted with label content that exceeds the 8000-token limit, SciBERT steps in to curate the information. It accomplishes this by sorting sentences within the label's content based on their relevance, ensuring that the most significant and informative sentences are retained. This selective approach not only condenses the text but also guarantees that the retained content is meaningful and captures the essence of the label's category.

F. Summarization with LED_Large

we then use LED_Large model to summarize the content within each label. Traditional transformers like BERT and GPT face limitations with lengthy sequences due to the quadratic scaling of self-attention. LED_Large, with its linear-scaling attention mechanism, efficiently processes and understands comprehensive documents, ensuring the generation of concise summaries that encapsulate the depth and critical insights of the original papers.

G. Literature Review Generation

Finally, we merge the individually summarized content from different labels to construct a comprehensive and cohesive literature review.

V. EXPERIMENTAL RESULTS

In this section, we detail the comprehensive suite of evaluation metrics employed to assess the effectiveness of our summarization models, alongside the computational implementation utilized in this process. These metrics are crucial for understanding the performance of our models in various

aspects, including semantic accuracy, readability, and overall summary quality.

A. Experimental Setup

Our research utilized Google Colab's GPUs: NVIDIA Tesla A100 with 80GB RAM for demanding tasks, V100 with 12GB for moderate tasks, and T4 with 12GB for lighter tasks, balancing performance and cost, which totaled approximately 20 USD for over 100 compute units. This strategic resource allocation was pivotal in our computational studies, underscoring the importance of hardware selection in machine learning research.

B. Evaluation Metrics

To rigorously evaluate our models, we employed the following metrics, each with its own unique formula and purpose:

$$\text{BARTScore} = \sum_{t=1}^{m} \omega_t \log p(y_t | y_{<t}, x, \theta) \tag{1}$$

The BARTScore integrates weights ω_t for each token t, considering the sequence $y_{<t}$, context x, and model parameters θ to evaluate summary quality.

$$\begin{aligned} \text{BERTScore} = \frac{1}{N} \sum_{i=1}^{N} \max_j \text{Cos}(\text{BERT}_\text{g}[i], \text{BERT}_\text{r}[j]) \\ + \frac{1}{M} \sum_{j=1}^{M} \max_i \text{Cos}(\text{BERT}_\text{g}[i], \text{BERT}_\text{r}[j]) \end{aligned} \tag{2}$$

The BERTScore employs cosine similarity between BERT embeddings of tokens in the generated summary (BERT_g) and the reference text (BERT_r), averaged over the number of tokens in both texts, N and M, to assess similarity.

$$\text{Cosine Similarity} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3}$$

where:

- A_i and B_i are the components of vectors A and B in the i-th dimension,
- n is the dimensionality of the vectors, representing the number of features or attributes in each vector.

In this context, A and B represent the vectorized forms of the generated summary and the reference text, respectively. These vectors are typically derived from text embeddings, like BERT embeddings, which capture semantic and syntactic features of the texts.

$$\text{ROUGE-N} = \frac{\sum_{s \in \{\text{Reference Summaries}\}} \sum_{gram_n \in s} \text{Count}_\text{match}(gram_n)}{\sum_{s \in \{\text{Reference Summaries}\}} \sum_{gram_n \in s} \text{Count}(gram_n)} \tag{4}$$

ROUGE-N calculates the ratio of matching n-grams in the generated summary to those in the reference summaries, reflecting the summary's quality by measuring its overlap with reference texts.

ROUGE-N is a recall-oriented metric used in evaluating text summarization and machine translation. It measures the overlap of n-grams between the generated summary and the reference summaries.

$$\begin{aligned} \text{FRES} = 206.835 - 1.015 \times \left(\frac{\text{total words}}{\text{total sentences}} \right) \\ - 84.6 \times \left(\frac{\text{total syllables}}{\text{total words}} \right) \end{aligned} \tag{5}$$

The FRES formula incorporates the total number of words, sentences, and syllables within a text, offering insights into its readability. These elements collectively determine the text's complexity and ease of understanding.

The Flesch Reading Ease Score is a widely recognized tool for assessing the readability of English text. It quantifies how easy it is to comprehend a piece of writing.

C. Fine Tuning

Our study tackled token limitations in summarization models by initially processing scientific texts with SciBERT to extract key sentences, reducing token counts for compatibility with LED-Large, BART, and Pegasus models. Due to token constraints—8000 for LED-Large and 1024 for others—we fine-tuned each model accordingly, with BART and Pegasus undergoing 30 epochs and LED-Large 5 epochs, to compare their summarization efficacy on condensed scientific content.

D. Model Evaluation

1) BART: In our comprehensive evaluation of the BART_large model, we utilized a range of metrics to ascertain the quality of the generated summaries. Figure 2 shows the training loss over steps, which is a critical indicator of the model's learning performance.

As observed in Figure 2, the training loss starts at a high value and exhibits a steep decline within the initial training steps, indicating a rapid learning phase as the model begins to fit the training data. Following the sharp initial decrease, the loss curve flattens, suggesting that the model's ability to learn from the training data diminishes as it converges towards an optimal set of parameters. This typical loss trajectory is indicative of a well-tuned learning rate where the model is capable of making significant improvements in the early stages, followed by fine-tuning as it approaches peak performance.

Table III provides an overview of the additional results obtained from the evaluation metrics applied to the generated summaries.

TABLE I: Summary of ROUGE Scores (P = Precision, R = Recall)

Model	*ROUGE-1 P*	*ROUGE-1 R*	*ROUGE-1 F1*	*ROUGE-2 P*	*ROUGE-2 R*	*ROUGE-2 F1*	*ROUGE-L P*	*ROUGE-L F1*
LED_Large	0.4462	0.5904	0.4991	0.3166	0.4314	0.3496	0.4336	0.4770
BART_Large	0.2483	0.2171	0.2231	0.4486	0.3614	0.3832	0.2226	0.2004
PEGASUS-arXiv	0.3192	0.1853	0.2259	0.8670	0.8371	0.8456	0.2888	0.2041
PEGASUS-Large	0.3034	0.1941	0.2267	0.8663	0.8489	0.8477	0.2785	0.2077
PEGASUS-xsum	0.3310	0.1586	0.2589	0.6671	0.8268	0.3649	0.3024	0.1875

TABLE II: Summary of Cosine Similarity, Flesch Readability, BERTScore, and BARTScore

Model	*Cosine Similarity*	*Flesch Readability*	*BERT P*	*BERT R*	*BERT F1*	*BARTScore*
LED_Large	0.7344	28.506	0.8612	0.8866	0.8734	-3.3096
BART_Large	0.5027	30.450	0.8404	0.8350	0.8376	-4.6432
PEGASUS-arXiv	0.4572	36.480	0.8361	0.8207	0.8282	-4.8164
PEGASUS-Large	0.4716	35.536	0.8384	0.8227	0.8304	-4.7994
PEGASUS-xsum	0.4344	39.557	0.8479	0.8146	0.8328	-4.8567

TABLE III: Comprehensive Evaluation Metrics for Various Models

Metric	BART_large	Pegasus_arxiv	Pegasus-large	Pegasus-xsum	LED_Large
FRES (Readability)	30.4503	36.4792	35.5386	39.5575	28.5852
ROUGE-1 F1-Score	0.2231	0.2259	0.2267	0.2589	0.4991
ROUGE-1 Precision	0.2483	0.3192	0.3034	0.3310	0.4462
ROUGE-1 Recall	0.2171	0.1854	0.1945	0.1586	0.5900
ROUGE-2 F1-Score	0.0832	0.2457	0.2477	0.3649	0.3496
ROUGE-2 Precision	0.0446	0.6749	0.6663	0.6741	0.3166
ROUGE-2 Recall	0.0361	0.8377	0.4899	0.2679	0.4314
ROUGE-L F1-Score	0.2084	0.2047	0.2077	0.1875	0.4772
ROUGE-L Precision	0.2226	0.2889	0.2785	0.3024	0.4354
ROUGE-L Recall	0.1952	0.1677	0.1777	0.1442	0.5728
BERTScore Precision	0.8404	0.8361	0.8384	0.8479	0.8612
BERTScore Recall	0.8350	0.8267	0.8227	0.8147	0.8867
BERTScore F1	0.8376	0.8282	0.8304	0.8328	0.8736
BARTScore	-4.6432	-4.8164	-4.7994	-4.8567	-3.3962
Cosine Similarity	0.5030	0.4571	0.4716	0.4344	0.7340

Fig. 2: Training Loss Over Steps for the BART_large Model

The Flesch Reading Ease Score (FRES), which is indicative of how comprehensible the summaries are to an average reader, stood at 30.4503. This score suggests that the text may pose comprehension challenges for readers without a college-level education.

ROUGE metrics, which assess the overlap of n-grams between the generated and reference summaries, provided a nuanced view of the model's performance. The ROUGE-1 F1-score was 0.2231, with a precision of 0.2483 and a recall of 0.2171, indicating a moderate level of word overlap. For ROUGE-2, which considers two-word sequences, the scores were significantly lower, with an F1-score of 0.0832, precision of 0.0446, and recall of 0.0361, implying that the model's ability to replicate bi-gram sequences from the reference summaries could be enhanced. The ROUGE-L scores, evaluating the longest matching sequence of words, were relatively higher, with an F1-score of 0.2084, precision of 0.2226, and recall of 0.1952, suggesting better performance in capturing longer sequences.

BERTScore was employed to measure the semantic equivalence between the generated and reference summaries. The high scores in precision (0.8404), recall (0.8350), and F1 (0.8376) indicate that the model's summaries are semantically similar to the expected outcomes and maintain a strong contextual alignment with the reference texts.

The BARTScore, which evaluates text generation quality, was -4.6432. Although negative, this score is typical within the context of machine-generated text evaluation, where lower scores can often reflect higher levels of fluency and factual alignment.

The average Cosine Similarity of 0.5030 is indicative of a moderate degree of semantic similarity when considering the vector space representations of the generated and reference summaries. This metric confirms that the semantic content of the model's output is directionally aligned with the reference data, although there is room for improvement.

2) Pegasus_arxiv: This section details the evaluation of the Pegasus_arxiv model, focusing on its training loss trajectory and performance metrics.

Figure 3 presents the training loss of the Pegasus_arxiv model over numerous training steps.

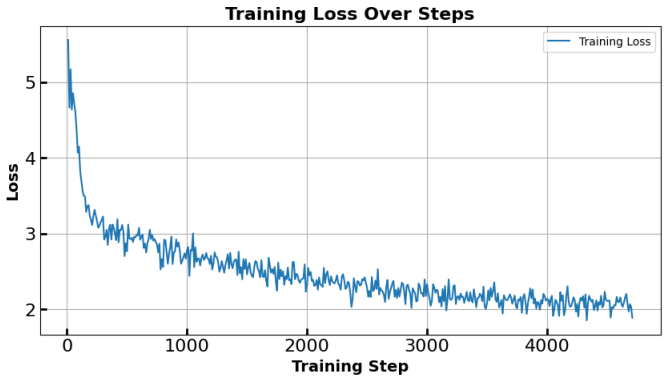

Fig. 3: Training Loss Over Steps for the Pegasus_arxiv Model

The training loss graph illustrates a rapid decrease in loss at the commencement of training, suggesting an effective initial learning phase. As the steps progress, the loss demonstrates a pattern of minor fluctuations while generally decreasing, which indicates the model's ongoing adjustments and learning from the training data. The absence of a plateau at the final steps suggests potential for further training, where the model might continue to improve with additional epochs.

The Pegasus_arxiv model's performance is quantified using several established metrics, as summarized in Table III.

The Flesch Reading Ease Score (FRES) for generated summaries is 36.4792, indicating moderate complexity and accessibility for readers with some college education. The ROUGE-1 scores reflect basic unigram matching between the generated and reference texts, with a precision of 0.3192 suggesting the model's strength in identifying key terms. However, the recall of 0.1854 indicates room for improvement in coverage. The ROUGE-2 and ROUGE-L metrics, which assess longer sequences of text, show a similar trend in precision over recall, pointing to the model's capacity to generate relevant phrases but not always capturing the full scope of the reference summaries.

BERTScore results with precision at 0.8361, recall at 0.8267, and an F1 score of 0.8282, underscore the model's semantic accuracy. The BARTScore of -4.8164, while negative, is within expected ranges for machine learning models, indicating the model's fluency and coherence. Finally, the Cosine Similarity score of 0.4571 reflects a moderate semantic alignment with the target summaries.

In conclusion, the Pegasus_arxiv model exhibits a promising learning curve and achieves respectable scores across several evaluation metrics. While it shows proficiency in generating semantically relevant text, the results suggest potential areas for improvement in maximizing recall and extending training to refine performance further.

3) Pegasus_large: We present an in-depth analysis of the Pegasus-large model, examining the training loss graph and various performance metrics that reflect the model's capabilities in generating text summaries.

The training loss graph, depicted in Figure 4, provides insight into the model's learning efficiency throughout the training steps.

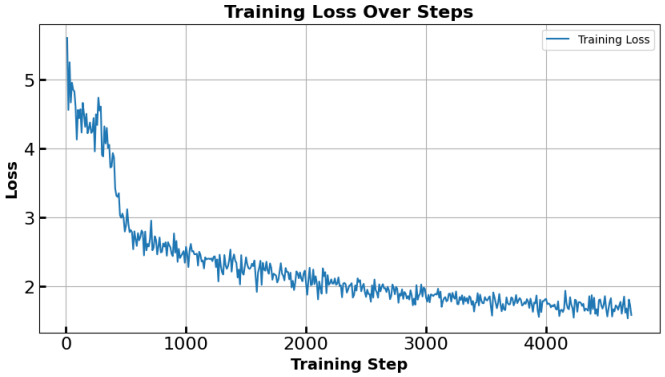

Fig. 4: Training Loss Over Steps for the Pegasus-large Model

As shown in Figure 4, the training loss rapidly decreases in the initial phase, suggesting a fast adaptation of the model to the training data. This is followed by a gradual decline and stabilization, indicating the model's convergence. Notably, the loss demonstrates occasional spikes, which may be attributed to the model's adjustments to new patterns within the data or learning rate adaptations. The overall downward trend, without a clear plateau, suggests that further gains could be achieved by continuing the training process.

Table III summarizes the performance metrics for the Pegasus-large model, highlighting its effectiveness in various aspects of text summarization.

The Flesch Reading Ease Score (FRES) of 35.5386 indicates that the summaries generated by Pegasus-large have a moderate level of readability. The ROUGE-1 metrics show fair unigram overlap with a better precision than recall, suggesting the model's relative proficiency in capturing key terms but less effectiveness in covering the entirety of the content. The ROUGE-2 and ROUGE-L scores follow a similar pattern, with higher precision indicating the model's ability to replicate important bigrams and longer subsequences from the reference summaries.

The BERTScore provides an assessment of semantic similarity, with scores of 0.8384 for precision, 0.8227 for recall, and 0.8304 for the F1 score, demonstrating that the model produces summaries that are semantically close to the reference texts. The BARTScore, at -4.7994, is within the range typically observed for this type of model, indicating satisfactory fluency and coherence. Lastly, the Cosine Similarity score of 0.4716 reflects a moderate degree of semantic alignment with the reference summaries.

The analysis indicates that while the Pegasus-large model performs well in terms of semantic relevance and precision,

there is potential for improvement in terms of recall and readability. Future iterations of the model could benefit from extended training and optimization to enhance these aspects.

4) Pegasus-xsum: The Pegasus-xsum model's performance was evaluated using a series of metrics, each providing insights into different facets of the model's text summarization capabilities.

A visual representation of the model's training loss is shown in Figure 5, depicting the loss magnitude across training steps.

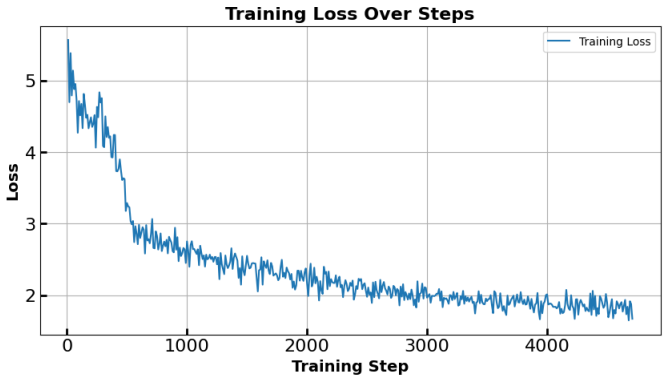

Fig. 5: Training Loss Over Steps for the Pegasus-xsum Model

The graph in Figure 5 illustrates the model's training loss decrease, starting from a high level and showing a steep decline, indicating a significant learning progression initially. As the training proceeds, the loss levels off, with some fluctuations that suggest the model's continuous learning and adaptation. The absence of a plateau towards the end of training may imply that the model has not yet fully converged and might benefit from additional training iterations.

The table below Table III presents the metrics used to evaluate the Pegasus-xsum model's performance.

The readability of the summaries, as measured by the Flesch Reading Ease Score (FRES), is relatively high at 39.5575, indicating that the summaries are fairly easy to understand for the general population. The ROUGE scores demonstrate the model's varying performance in matching n-grams with the reference summaries, with ROUGE-1 and ROUGE-2 showing stronger precision than recall. This suggests that while the model is quite good at identifying the most relevant terms and bigrams, it may miss some content from the references.

There was an error in processing the BARTScore image; therefore, the score is not available. The BERTScores, with precision at 0.8479 and an F1 score of 0.8328, reflect the model's strong capability in capturing the semantic essence of the reference summaries. The Cosine Similarity score of 0.4344 further corroborates this, though it also indicates room for improvement in semantic alignment.

The evaluation demonstrates that the Pegasus-xsum model is capable of generating summaries that are relatively easy to read and semantically aligned with reference texts. However, the lower recall scores across ROUGE metrics suggest that the model could be improved to capture a broader range of information present in the source texts.

5) LED_Large: In this section, we evaluate the LED_Large (Longformer Encoder-Decoder) model, which is particularly designed for summarizing extended documents. The model's training behavior over a limited number of epochs and its comparative performance are explored.

The LED_Large model's training trajectory over the course of three epochs is visualized in Figure 6.

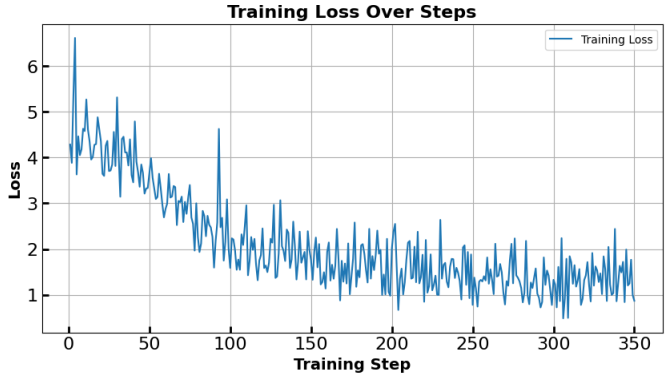

Fig. 6: Training Loss Over Steps for the LED_Large Model

Figure 6 shows the training loss decreasing significantly within the initial epochs. Although the graph exhibits variability, it suggests a trend toward stabilization. Given that the model was only trained for three epochs, as opposed to thirty for other models, we posit that the loss would likely continue to diminish and level out if given more training time. This trend indicates the model's capacity for further improvement and efficiency in learning from the training data.

The LED_Large model's performance is quantified by the metrics presented in Table III.

Despite the limited number of training epochs, the LED_Large model outperforms the others in several key metrics. Its FRES score suggests that the generated summaries are on the more complex side, yet the high ROUGE and BERTScore values indicate a strong ability to capture and reflect the content of the source documents both in terms of n-gram overlap and semantic quality. Notably, the high Cosine Similarity score demonstrates the model's capability to produce semantically rich summaries closely aligned with the reference texts.

E. Comparative Analysis

In this research, we undertook a comparative analysis of several state-of-the-art summarization models: BART_large, Pegasus_arxiv, Pegasus_large, Pegasus_xsum, and LED_Large. The evaluation was based on various metrics, including the Flesch Reading Ease Score (FRES), ROUGE scores, BERTScore, BARTScore, and Cosine Similarity, as well as the training loss over steps, which serves as an indicator of each model's learning performance.

1) Training Efficiency: An initial observation across models is the variance in training efficiency. The BART_large model (Figure 2) and Pegasus_large model (Figure 4) showed a typical loss trajectory, with rapid decreases in the early stages followed by a plateau, indicative of a well-tuned learning rate. In contrast, the Pegasus_arxiv model (Figure 3) and Pegasus_xsum model (Figure 5) did not display a clear plateau, suggesting potential for further performance improvements with additional training epochs. Notably, the LED_Large model (Figure 6), even after only three epochs of training, showed a promising decline in loss, implying an exceptional learning capability that outpaced the others.

2) Readability and Comprehensibility: Analyzing readability via FRES, the Pegasus_xsum model achieved the highest score (39.5575), indicating summaries that are easier for the general population to understand. The LED_Large model, despite its brevity of training, performed well but generated summaries that were more complex, as reflected by a lower FRES score of 28.5852. This complexity may be attributed to the model's design for extended document summarization, which tends to be inherently more intricate.

3) Semantic and N-gram Overlap: The ROUGE scores highlighted each model's ability to match n-grams with reference summaries. LED_Large excelled in ROUGE-1 recall, suggesting its strength in capturing the majority of content. However, its precision scores point towards a potential for generating more concise summaries. BART_large showed room for improvement, particularly in ROUGE-2 and ROUGE-L scores, to enhance the overlap of higher-order n-grams with reference texts.

Semantic similarity, as measured by BERTScore and Cosine Similarity, was a strong point for LED_Large, with high scores indicating a strong alignment with reference summaries in terms of context and meaning. BART_large and Pegasus models also performed well, confirming their capability to maintain semantic content directionally aligned with reference data.

4) Quality of Text Generation: BARTScore evaluations revealed a range within expected levels for machine-generated text. While negative, lower BARTScores are often reflective of higher fluency and factual consistency. The LED_Large model had a BARTScore indicating satisfactory fluency, although it was slightly lower than that of the Pegasus and BART models.

VI. Conclusion and Future Work

This paper presented SCHOLARMATE, a system designed for summarizing scientific papers, particularly focusing on the fields of artificial intelligence and deep learning. By integrating and fine-tuning state-of-the-art NLP models such as LED_Large, Pegasus-Large, BART, and SciBERT, etc., we aimed to overcome the limitations of traditional summarization models, which often fail to capture the complex nuances of academic papers. Our approach, characterized by a multi-model strategy, demonstrated significant improvements in accuracy and depth of summaries over existing methods.

The evaluation of our models using rigorous metrics such as cosine similarity, ROUGE scores, BERTScore, BARTScore, and the Flesch Reading Ease Score, confirmed the efficacy of our system. The LED_Large model, in particular, showcased exceptional performance with its ability to generate high-quality summaries even with fewer training epochs compared to its counterparts. In conclusion, SCHOLARMATE represents a significant step forward in the automatic summarization of scientific literature. The promising results obtained lay the groundwork for further development, with the objective of creating a system that is not only accurate and thorough but also efficient and user-friendly.

References

[1] AllenAI. allenai/led-large-16384-arxiv. [Online]. Available: https://huggingface.co/allenai/led-large-16384-arxiv

[2] J. Zhang. Pegasus-large. [Online]. Available: https://huggingface.co/google/pegasus-large

[3] 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," *arXiv preprint arXiv:1910.13461*, 2019.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[5] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," *arXiv preprint arXiv:1903.10676*, 2019.

[6] facebook/bart-large-cnn. [Online]. Available: https://huggingface.co/facebook/bart-large-cnn

[7] scibert_scivocab_uncased. [Online]. Available: https://huggingface.co/allenai/scibert_scivocab_uncased

[8] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[9] A. Cohan, F. Dernoncourt, D. S. Kim, T. Bui, S. Kim, W. Chang, and N. Goharian, "A discourse-aware attention model for abstractive summarization of long documents," *arXiv preprint arXiv:1804.05685*, 2018.

[10] Y. Liu, "Fine-tune bert for extractive summarization," *arXiv preprint arXiv:1903.10318*, 2019.

[11] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," in *International Conference on Machine Learning*. PMLR, 2020, pp. 11 328–11 339.

[12] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[13] M. Guo, J. Ainslie, D. Uthus, S. Ontanon, J. Ni, Y.-H. Sung, and Y. Yang, "Longt5: Efficient text-to-text transformer for long sequences," *arXiv preprint arXiv:2112.07916*, 2021.

[14] W. Kryściński, N. Rajani, D. Agarwal, C. Xiong, and D. Radev, "Booksum: A collection of datasets for long-form narrative summarization," *arXiv preprint arXiv:2105.08209*, 2021.

[15] W. Liu, Y. Gao, J. Li, and Y. Yang, "A combined extractive with abstractive model for summarization," *IEEE Access*, vol. 9, pp. 43 970–43 980, 2021.

[16] Y. Zhao, M. Khalman, R. Joshi, S. Narayan, M. Saleh, and P. J. Liu, "Calibrating sequence likelihood improves conditional language generation," *arXiv preprint arXiv:2210.00045*, 2022.

[17] A. Gera, A. Halfon, E. Shnarch, Y. Perlitz, L. Ein-Dor, and N. Slonim, "Zero-shot text classification with self-training," *arXiv preprint arXiv:2210.17541*, 2022.

[18] W. Yuan, G. Neubig, and P. Liu, "Bartscore: Evaluating generated text as text generation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 263–27 277, 2021.

[19] K. Girthana, S. Swamynathan, A. Nirupama, S. S. Akshya, and S. Adhithyan, "Web-based pretrained transformer model for scientific paper summarization (wpt-sps)," in *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*. IEEE, 2023, pp. 298–302.

[20] S. Badhe, M. Hasan, V. Rughwani, and R. Koshy, "Synopsis creation for research paper using text summarization models," in *2023 4th International Conference for Emerging Technology (INCET)*, 2023, pp. 1–5.

[21] A. Joshi, E. Fidalgo, E. Alegre, and L. Fernández-Robles, "Deepsumm: Exploiting topic models and sequence to sequence networks for extractive text summarization," *Expert Systems with Applications*, vol. 211, p. 118442, 2023.

[22] H. Zhang, X. Liu, and J. Zhang, "Extractive summarization via chatgpt for faithful summary generation," *arXiv preprint arXiv:2304.04193*, 2023.