Automating Scientific Highlight Generation with Transformer Models

Sangita Singh[1,*], Navya Sinha[2] and Jyoti Prakash Singh[1]

[1] *National Institute of Technology Patna, Patna, 800005, Bihar*
[2] *Sikkim Manipal Institute of Technology, Majitar, Sikkim, India*

Abstract

This work addresses the task of automatic highlight generation from scientific papers' abstracts, a challenging problem in research communication and summarization. To solve this issue, we used benchmark transformer-based models, including T5-small, PEGASUS, and LongT5, as well as an entity-aware variant of PEGASUS. Experimental results demonstrate that PEGASUS achieves the strongest overall performance in terms of ROUGE-1: 0.3272, ROUGE-2: 0.1166, ROUGE-L: 0.2345, and METEOR: 0.2841. These findings establish PEGASUS as the most effective approach for abstractive highlight generation, while also highlighting the limitations of entity-focused methods in this domain.

Keywords

highlights generation, Pegasus, T5 small, T5 long, and SciHigh-2025 Datasets

1. Introduction

The exponential growth of scientific publications has made it increasingly difficult for researchers to keep pace with new findings. According to recent estimates, millions of articles are published annually across various disciplines, creating an urgent need for automated tools that can distill essential information into concise and accessible forms. This raises a critical question: how can we enable researchers to quickly grasp the core contributions of a paper without reading lengthy abstracts or full texts? From a practical perspective, automatic highlight generation helps users quickly grasp the most important aspects of a scientific abstract without reading everything. This is particularly valuable in domains like news, legal documents, academic research, meetings, and customer service logs, where time and efficiency are critical. It reduces cognitive load and supports faster decision-making. Early work focused on selecting important sentences or phrases directly from the text. Techniques included statistical scoring (TF-IDF, frequency counts), graph-based ranking methods like TextRank and LexRank, and supervised classifiers trained to identify "highlight-worthy" sentences. These methods are simple and ensure factual consistency but often produce highlights that lack coherence or readability. With the advent of neural networks, especially sequence-to-sequence models with attention, headline generation shifted toward abstractive summarization. Transformer-based models (e.g., BERTSUM, GPT) can rephrase content to produce more human-like highlights. These methods generate fluent summaries but may suffer from hallucination (introducing unsupported details).

Despite progress, existing systems often face challenges such as:

- Overly generic summaries that miss domain-specific highlights.
- Lack of personalization or context-awareness.
- Hallucination issues in abstractive methods (introducing information not in the source).
- Poor alignment with human notions of "highlight-worthy" content (e.g., emphasizing novelty, emotion, or relevance depending on use-case).

Forum for Information Retrieval Evaluation, December 17th-20th, 2025, Vanarasi, India

[*] Corresponding author.

sangitas.ph21.cs@nitp.ac.in (S. Singh); navsinha1405@gmail.com (N. Sinha); jps@nitp.ac.in (J. P. Singh)
0000-0002-9318-2186 (S. Singh); 0009-0000-7406-8890 (N. Sinha); 0000-0002-3742-7484 (J. P. Singh)

To address these challenges, we propose a highlight generation approach designed to enhance both the accuracy and usefulness of generated highlights. The main contributions of our work are summarized as follows:

1. We benchmark transformer-based generation models, including T5-small, PEGASUS, and LongT5, and further experiment with an entity-aware variant of PEGASUS.
2. We provide a comparative evaluation using ROUGE and METEOR metrics.
3. Our method ensures that highlights are not only concise and accurate, but also tailored to what end-users actually value in practice.

1.1. Research Objectives

The main objective was to develop and evaluate methods for automatically generating highlights (concise summaries) from scientific papers. Highlights are short, reader-friendly descriptions that capture the key contributions of a paper. Since writing them manually is time-consuming and subjective, our goal was to create a system that can generate accurate, informative, and human-like highlights to assist researchers, publishers, and readers.

The work was guided by the following research questions:

RQ1: Can automatic text summarization methods generate highlights that are comparable in quality to human-written highlights?

RQ2: Which approaches (extractive vs. abstractive models, or hybrid methods) are more effective for highlight generation in scientific writing?

RQ3: What evaluation metrics best capture the quality of highlights (e.g., ROUGE, METEOR, BERTScore, human evaluation)?

RQ4: How well do models trained on general summarization datasets transfer to the scientific domain compared to models fine-tuned on domain-specific corpora?

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the dataset and proposed model. Section 4 describe evaluation metrics. Section 5 presents the results produced by the proposed model. Section 6 discussed the implications of recent pretraining strategies, such as PEGASUS, T5 small, and Long T5, for highlight generation tasks. Section 7 concludes the paper with potential directions for future research.

2. Literature

In this section, we have discussed Highlight generation, a sub-task of text summarization, focuses on producing concise and salient snippets that capture the core meaning of a longer document. Unlike traditional summarization, which may yield multi-sentence abstracts, highlight generation demands brevity—often one or two sentences—while preserving informativeness. This problem is particularly relevant in domains such as news (where highlights serve as teasers), scientific publishing (where highlights guide readers through dense papers), and meeting records (where highlights emphasize key decisions or action items).

Cho et al. [1] introduced a framework for sub-sentence-level highlight generation to ensure self-contained and meaningful segments. Their method combined XLNet for extracting candidate segments with Determinantal Point Processes (DPP) for selecting salient and non-redundant highlights. Models were trained and evaluated on multi-document summarization datasets, including DUC-03/04 and TAC-08/09/10/11, while a classifier for assessing segment quality was trained separately on the CNN/DM dataset. The proposed HL-XLNetSegs model achieved ROUGE-1 = 39.2, ROUGE-2 = 10.70, and ROUGE-SU4 = 14.47 on DUC-04. Human evaluations confirmed that the extracted highlights were highly self-contained.

Woodsend et al. [2] developed a phrase-based Integer Linear Programming (ILP) model for joint content selection and compression in news summarization, designed to generate story highlights.

The model integrated syntactic information from PCFG parse trees and dependency graphs, encoding grammatical constraints into the optimization process. They constructed a dataset of approximately 9,000 CNN.com article–highlight pairs (2007–2009), with 210 pairs manually annotated. Evaluations on the DUC-2002 benchmark showed that the ILP model achieved ROUGE-1 = 0.445 and ROUGE-2 = 0.200, outperforming the lead-3 baseline, while human judges found no significant difference in grammaticality compared to CNN highlights.

Gupta et al. [3] presented a comprehensive survey of abstractive summarization approaches, categorizing methods into three paradigms: structure-based (templates, trees, ontologies), semantic-based (semantic graphs, predicate–argument structures), and deep learning-based (neural encoder–decoder models). While no new experiments were introduced, they reported typical performance ranges across benchmarks such as DUC and TAC. Neural approaches typically achieved ROUGE-1 between 0.28 and 0.47, while semantic graph-based methods ranged between 0.30 and 0.40. The authors also highlighted the limitations of ROUGE for abstractive summaries and advocated for more semantic-aware evaluation metrics.

Rehman et al. [4] conducted one of the earliest studies on automatic highlight generation from abstracts using deep learning. They evaluated three models: (1) a seq2seq model with attention, (2) a pointer-generator network (PGN), and (3) a PGN with coverage. Using GloVe embeddings and the CSPubSum dataset (10,142 papers), the PGN+Coverage model achieved the best results with ROUGE-1 = 31.46, ROUGE-2 = 8.57, ROUGE-L = 29.14, and METEOR = 12.01, outperforming both the seq2seq and PGN baselines.

Building on this, Rehman et al. [5] proposed an abstractive highlight generation model by enhancing the pointer-generator network with Named Entity Recognition (NER). Named entities were treated as single tokens to avoid fragmentation and preserve meaning. Using the CSPubSum dataset, the best-performing model (NER+PGM+Cov) achieved ROUGE-1 = 38.13, ROUGE-2 = 13.68, ROUGE-L = 35.11, METEOR = 31.03, and BERTScore = 86.3, outperforming non-NER variants.

In a subsequent study, Rehman et al. [6] integrated pre-trained ELMo embeddings with the pointer-generator network and coverage mechanism. Four models (PGM, PGM+Cov, PGM+ELMo, and PGM+ELMo+Cov) were evaluated on CSPubSum using two input settings: (a) abstract only, and (b) abstract + introduction + conclusion. The best model, PGM+ELMo+Cov, achieved ROUGE-1 = 38.40, ROUGE-2 = 13.32, ROUGE-L = 35.45, METEOR = 30.61, and BERTScore = 86.6, demonstrating consistent improvements over the baselines.

Further extending this line of work, Rehman et al. [7] incorporated SciBERT embeddings into a PGN+Coverage architecture for domain-specific contextual understanding. Experiments were conducted on both the CSPubSum and a new MixSub corpus of 19,785 multidisciplinary articles with author-written highlights. On CSPubSum, the model achieved ROUGE-1 = 38.26, ROUGE-2 = 14.20, ROUGE-L = 35.51, METEOR = 32.62, and BERTScore = 86.65. On MixSub, it achieved ROUGE-1 = 31.78 and METEOR = 24.00, demonstrating strong cross-domain generalization.

Xiang et al. [8] investigated the use of highlights to improve unsupervised keyword extraction. They enriched abstracts with highlight sentences and tested four strategies for combining abstracts and highlights, including semantic filtering. Three unsupervised models—TextRank, MDERank, and PromptRank—were evaluated on a new dataset of 1,647 computer science papers derived from Elsevier and CSPubSum. Incorporating highlights consistently improved extraction performance; for instance, MDERank with Highlights+Filtered Abstract (H+FA) achieved F1@10 = 15.06, while PromptRank with Highlights+Abstract (H+A) reached F1@10 = 16.47.

Recent advances in pre-trained sequence-to-sequence models, including PEGASUS, T5, and LongT5, have further improved highlight generation, yielding more fluent and contextually accurate outputs.

Necva et al. [9] introduced MoDeST, a multi-domain and multilingual dataset for scientific title generation in English and Turkish, spanning disciplines such as social sciences, medical sciences, and engineering. MoDeST supports generation from multiple sources—keywords, abstracts, and full articles. Evaluations using LLMs (LLaMA-3.1, Aya-expanse) in zero-shot, few-shot, and fine-tuning setups revealed that fine-tuning yields the best performance. For Turkish, models achieved scores of 40.12–47.22, and for English, 40.02–49.10. Abstracts were identified as the most effective input source.

This dataset highlights domain-specific and cross-lingual challenges, making it a valuable resource for future research.

Finally, Rehman et al. [10] explored three deep learning models for research highlight generation: (1) a seq2seq model with attention using 128-dimensional GloVe embeddings, (2) a pointer-generator network (PGN), and (3) a PGN with a coverage mechanism. All models used a vocabulary of 50K tokens, beam search size 4, and input/output constraints of 400 and 100 tokens, respectively. In later work, Rehman et al. [11] proposed a research plan emphasizing evaluation techniques for scientific text summarization, underscoring the importance of reliable metrics and the need to address evaluation challenges effectively.

3. Methodology

To address our research objectives, we designed an abstractive summarization framework and evaluated it using the MixSub-SciHigh dataset as follows:

3.1. Dataset

We evaluted our models based on a cleaned and enriched version of the MixSub-SciHigh dataset. This dataset is given in track SciHigh at FIRE-2025 [1][7]. The dataset contains 10,000 training instances, 1,985 validation instances, and 1,840 test instances. Each sample consists of an abstract paired with corresponding gold-standard highlights.

3.2. Proposed Models

We developed three transformer-based text generation models, organized under two teams: Text_highlights_gen and NIT_PATNA_2025. The NIT_PATNA_2025 team built two models, T5-small and LongT5, while the Text_highlights_gen team developed Pegasus and Pegasus with NER.

3.3. Pre-processing

Before training the models, we applied several preprocessing steps, including tokenization, punctuation removal, and the identification and storage of named entities.

- **T5-small** [12]: A compact variant of the Text-to-Text Transfer Transformer (T5), where all NLP tasks are reframed as text-to-text problems. Despite its smaller size compared to larger T5 versions, it is efficient and suitable for resource-constrained environments. For highlight generation, the task is formulated as: “document → highlights”.
- **LongT5** [13]: An extension of T5 for handling longer input sequences. It introduces efficient attention mechanisms such as local-global attention to reduce the quadratic cost of standard transformers, enabling it to process full-length scientific articles rather than just abstracts. This makes it particularly suitable for highlight generation when important information is spread across lengthy papers.
- **PEGASUS** [14], which is shown in Figure 1, is pre-trained with a gap-sentence generation objective optimized for abstractive summarization. Its novelty lies in the *gap-sentence generation* (GSG) pre-training, where entire sentences are masked and the model learns to reconstruct them from the remaining context. This objective closely mimics summarization, enabling PEGASUS to generate coherent, compressed highlights. During fine-tuning, the encoder processes the scientific abstract and the decoder generates highlights autoregressively, with beam search improving output quality. This alignment between pre-training and summarization tasks makes PEGASUS especially effective for highlight generation.
  Additionally, we developed an **entity-aware PEGASUS** variant, where named entity information from abstracts was incorporated into the input representation to assess its effect on summarization.

[1] https://sites.google.com/jadavpuruniversity.in/scihigh2025/home

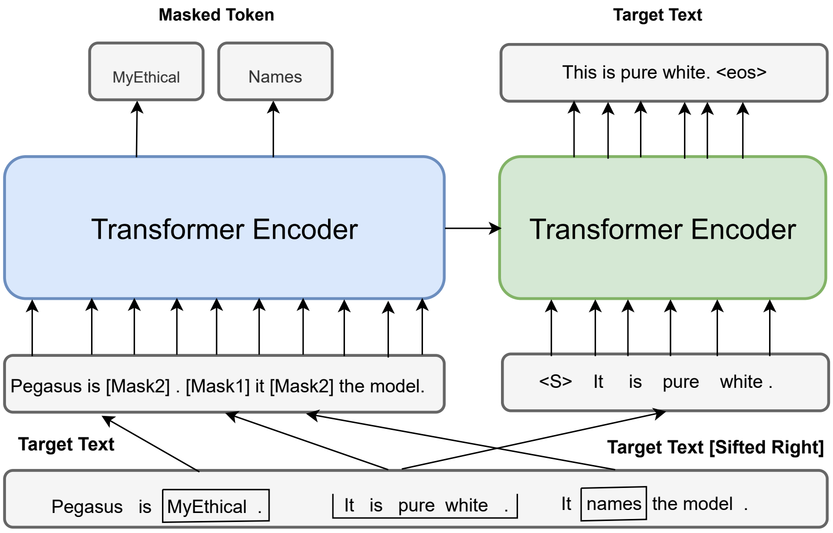

Figure 1: Architecture of the PEGASUS model to generate Highlights.

3.3.1. Hyperparameters

These models were fine-tuned using the Hugging Face Transformers framework. Training was performed with the following hyperparameters:

- **Learning rate**: The model is trained using the Adam optimizer with a learning rate of $2e{-}5$, selected to ensure stable convergence during training.
- **Batch size**: The model is trained with a batch size of 2 using the Adam optimizer and a learning rate of $2e{-}5$, chosen to ensure stable convergence during training.
- **Epochs**: The model is trained for 10 epochs with a batch size of 2 using the Adam optimizer and a learning rate of $2e{-}5$, selected to ensure stable convergence during training.
- **Beam search**: The model is trained for 10 epochs with a batch size of 2 using the Adam optimizer and a learning rate of $2e{-}5$, and decoding is performed with a beam search width of 4 to improve sequence generation quality.
- **Maximum input length**: The model is trained for 10 epochs with a batch size of 2 using the Adam optimizer and a learning rate of $2e{-}5$. Inputs are truncated to a maximum length of 512 tokens, and decoding is performed with a beam search width of 4 to improve sequence generation quality.
- **Maximum output length**: The model is trained for 10 epochs with a batch size of 2 using the Adam optimizer and a learning rate of $2e{-}5$. Inputs are truncated to a maximum length of 512 tokens, outputs are limited to 100 tokens, and decoding is performed with a beam search width of 4 to improve sequence generation quality
- **Weight_decay**: The model is trained for 10 epochs with a batch size of 2 using the Adam optimizer with a learning rate of $2e{-}5$ and a weight decay of 0.01. Inputs are truncated to a maximum length of 512 tokens, outputs are limited to 100 tokens, and decoding is performed with a beam search width of 4 to improve sequence generation quality

4. Evaluation Metrics

We evaluated the performance of our models through ROUGE-N, ROUGE-L [15], BERTScore [16] and METEOR [17] metrics. These metrics are employed to evaluate the similarity between the generated text and the reference text.

ROUGE-N: It measures the overlap of n-grams between a system-generated text and a reference text. Here, we took ROUGE-1, and ROUGE-2 metrics to evaluate our proposed models. It is defined as:

$$\text{Precision} = \frac{\text{Overlap}(n\text{-grams})}{\text{Total } n\text{-grams in system summary}} \tag{1}$$

$$\text{Recall} = \frac{\text{Overlap}(n\text{-grams})}{\text{Total } n\text{-grams in reference summary}} \tag{2}$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

For ROUGE-1, $n = 1$ (unigrams). For ROUGE-2, $n = 2$ (bigrams).

ROUGE-L: It is based on the Longest Common Subsequence (LCS) between the system-generated text and a reference text. Unlike ROUGE-N, it does not require consecutive matches, but it preserves sentence-level word order. It is defined using recall, precision, and F-score.

$$\text{Recall}_{LCS} = \frac{LCS(X, Y)}{|Y|} \tag{4}$$

$$\text{Precision}_{LCS} = \frac{LCS(X, Y)}{|X|} \tag{5}$$

$$F_{LCS} = \frac{(1 + \beta^2) \cdot \text{Recall}_{LCS} \cdot \text{Precision}_{LCS}}{\text{Recall}_{LCS} + \beta^2 \cdot \text{Precision}_{LCS}} \tag{6}$$

where $LCS(X, Y)$ is the length of the longest common subsequence between system-generated text X and reference text Y, and $|X|$, $|Y|$ are their lengths.

METEOR: It aligns candidate and reference texts using exact matches, stemming, synonyms, and paraphrases. It combines precision and recall into a harmonic mean and applies a fragmentation penalty to account for word order. By incorporating linguistic variations and semantic similarity, METEOR is better correlated with human judgment than simple overlap-based metrics.

$$F_{\text{mean}} = \frac{10 \cdot P \cdot R}{R + 9P} \tag{7}$$

$$\text{Penalty} = 0.5 \left(\frac{\#\text{chunks}}{\#\text{matches}} \right)^3 \tag{8}$$

$$\text{METEOR} = (1 - \text{Penalty}) \cdot F_{\text{mean}} \tag{9}$$

where P is Precision, R is Recall, “matches” is the number of unigram matches (exact, stem, synonym), and “chunks” is the number of contiguous matched word sequences.

5. Results

Table 1 reports the evaluation results on the MixSub-SciHigh validation dataset using F1-scores for ROUGE-1, ROUGE-2, ROUGE-L, and BERTScore. Among all models, PEGASUS achieved the best performance, obtaining the highest scores across every metric (ROUGE-1: 35.72, ROUGE-2: 15.66, ROUGE-L: 25.45, BERTScore: 89.41). Table 2 reports the evaluation results on the MixSub-SciHigh test dataset using F1-scores for ROUGE-1, ROUGE-2, ROUGE-L, and METEOR. Among all models, PEGASUS achieved the best performance, obtaining the highest scores across every metric (ROUGE-1: 32.72, ROUGE-2: 11.66, ROUGE-L: 23.45, METEOR: 28.41). Compared to T5-small, PEGASUS provided consistent improvements of around 1–1.2 points on ROUGE-1 and ROUGE-L and a modest gain on METEOR.

Interestingly, while incorporating NER into PEGASUS was expected to boost performance, the results instead showed a significant decline, with ROUGE-2 dropping from 11.66 to 6.89 and METEOR from 28.41 to 19.46. This suggests that the NER-based preprocessing may have introduced noise or disrupted contextual coherence in summaries for this dataset.

LongT5, despite being designed to handle longer contexts, performed poorly in this task, with ROUGE-2 as low as 2.08 and METEOR at only 12.78, highlighting its limitations when applied to relatively short scientific abstracts in the MixSub-SciHigh dataset.

Overall, PEGASUS emerged as the most effective model for scientific highlight generation in this setting, demonstrating that its pre-training objectives are well-suited for abstractive summarization of domain-specific text.

Table 3 presents the ranks of all participating teams based on ROUGE-L scores in the SciHigh shared task. We have two teams named (1) Text_highlights_gen and (2) NIT_PATNA_2025. The team Text_highlights_gen achieved the top performance with a ROUGE-L score of 23.45, followed closely by AiNauts (23.24) and SVNIT_CSE (23.02). The differences among the top three submissions are relatively small (less than 0.5 points), indicating strong competition at the upper end of the leaderboard.

Our second team, NIT_PATNA_2025, secured the 6th position with a ROUGE-L score of 22.42, placing us within the top half of all participating teams. The results demonstrate that our approach performs competitively, outperforming several strong baselines such as MUCS (22.08) and JU_CSE_PR_KS (22.06), while remaining close to higher-ranked systems like The NLP Explorers (22.94).

Overall, the leaderboard highlights that while the leading systems achieve similar ROUGE-L scores, even minor improvements can significantly influence rank positions. Our system's placement within the top tier validates the effectiveness of our summarization strategy.

Table 1
Evaluation results on the MixSub-SciHigh validation dataset using F1-scores for ROUGE-1, ROUGE-2, and ROUGE-L metrics. The best scores are shown in bold.

Model	ROUGE-1	ROUGE-2	ROUGE-L
T5-small	28.67	9.05	20.42
PEGASUS	**35.72**	**15.66**	**25.45**
PEGASUS + NER	23.48	04.89	18.11
LongT5	12.54	00.08	8.38

Table 2
Evaluation results on the MixSub-SciHigh test dataset using F1-scores for ROUGE-1, ROUGE-2, ROUGE-L, and METEOR metrics. The best scores are shown in bold.

Model	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
T5-small	31.67	11.05	22.42	28.04
PEGASUS	**32.72**	**11.66**	**23.45**	**28.41**
PEGASUS + NER	25.48	06.89	20.11	19.46
LongT5	16.54	02.08	14.38	12.78

6. Discussion

The evaluation results in Table 2 highlight several important insights into the performance of different transformer-based summarization models on the MixSub-SciHigh test dataset. Among all the models, PEGASUS demonstrated the strongest performance, achieving the highest scores across all evaluation metrics. Its superior ROUGE-1 (32.72), ROUGE-2 (11.66), ROUGE-L (23.45), and METEOR (28.41) indicate that the pre-training objectives of PEGASUS, which focus on gap-sentence generation, are well aligned with the requirements of abstractive scientific summarization.

Table 3
Ranks of all teams based on ROUGE-L performance.

Group Name	Run Submission	ROUGE-L	Rank
Text_highlights_gen	**run 1**	**23.45**	**1**
AiNauts	run 1	23.24	2
SVNIT_CSE	run 1	23.02	3
NLPFusion	run 2	22.96	4
The NLP Explorers	run 2	22.94	5
NIT_PATNA_2025	**run 1**	**22.42**	**6**
MUCS	run 1	22.08	7
JU_CSE_PR_KS	run 1	22.06	8
SCaLAR	run 1	20.33	9
Ayanika 7	run 1	17.91	10
Shilpo	run 1	16.75	11
TJP	run 1	06.73	12

While T5-small performed slightly below PEGASUS, its results were still competitive, confirming its capability as a baseline model for scientific text summarization. The relative gap between T5-small and PEGASUS (e.g., +1.05 on ROUGE-1, +0.61 on ROUGE-2) suggests that PEGASUS has an advantage in capturing context and generating more coherent summaries, albeit with only modest improvements.

Surprisingly, the PEGASUS + NER variant underperformed significantly compared to its vanilla counterpart. Despite the expectation that entity-focused preprocessing would enhance content selection and factual consistency, the results show a sharp decline across all metrics (e.g., a 4.77-point drop in ROUGE-1 and nearly 9 points in METEOR). This suggests that the integration of NER-based features may have disrupted the natural flow of contextual information, leading to less fluent and incomplete summaries. It also highlights that naive incorporation of linguistic features does not always guarantee improvements and requires careful integration strategies.

The performance of LongT5 was the weakest among all models. Its notably low scores (ROUGE-2: 2.08, METEOR: 12.78) suggest that the model struggled with the relatively short and structured nature of the MixSub-SciHigh dataset. Although LongT5 is designed to handle long input contexts, this characteristic may not provide an advantage when processing shorter scientific abstracts, where concise content selection is more critical than handling extended contexts.

Overall, these findings emphasize that model pre-training objectives and dataset characteristics strongly influence summarization performance. PEGASUS emerges as the most effective choice for highlight generation in this domain, while approaches relying on NER or long-context handling require more tailored adaptation to achieve competitive results.

7. Conclusion and Future work

In this paper, we investigate the task of automatic highlight generation for scientific abstracts using transformer-based models. We introduce a cleaned and enriched version of the MixSub-SciHigh dataset, incorporating preprocessing steps such as tokenization, stopword removal, punctuation removal, and named entity recognition. On this dataset, we use multiple transformer models, including T5-small, LongT5, PEGASUS, and an entity-aware variant of PEGASUS to train these models. Our experiments demonstrate that PEGASUS achieved the best overall performance across ROUGE and METEOR metrics, making it the most suitable model for scientific highlight generation.

In the future, research will focus on incorporating advanced entity-aware highlights generation techniques that integrate knowledge graphs, ontology alignment, and domain-specific entity linking to enhance factual accuracy and semantic coherence. Controlled text generation with constraints on readability, factual grounding, and redundancy reduction will be further investigated to ensure high-quality highlights generation. Moreover, integrating reinforcement learning with fact-consistency objectives and human-in-the-loop feedback can help optimize highlights for both precision and interpretability. Expanding the dataset across multiple disciplines, including low-resource scientific domains, will support better generalization of long-sequence models. Finally, hybrid approaches that combine symbolic reasoning with large language models may open new directions for producing reliable, explainable, and domain-adaptive scientific highlights.

Acknowledgments

This first author would want to acknowledge the Ministry of Education (MOE), Government of India for financial support during the research work through the Rajiv Gandhi fellowship Ph.D scheme (UGC) for computer science & engineering.

Declaration on Generative AI

This paper includes no content generated by artificial intelligence tools beyond language editing and formatting assistance. All intellectual contributions, including the conception, analysis, and interpretation of results, were made by the authors.

References

[1] S. Cho, K. Song, C. Li, D. Yu, H. Foroosh, F. Liu, Better highlighting: Creating sub-sentence summary highlights, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 6282–6300. URL: https://doi.org/10.18653/v1/2020.emnlp-main.509. doi:10.18653/V1/2020.EMNLP-MAIN.509.

[2] K. Woodsend, M. Lapata, Automatic generation of story highlights, in: J. Hajic, S. Carberry, S. Clark (Eds.), ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden, The Association for Computer Linguistics, 2010, pp. 565–574. URL: https://aclanthology.org/P10-1058/.

[3] S. Gupta, S. K. Gupta, Abstractive summarization: An overview of the state of the art, Expert Systems with Applications 121 (2019) 49–65. URL: https://doi.org/10.1016/j.eswa.2018.12.011. doi:10.1016/j.eswa.2018.12.011.

[4] T. Rehman, D. K. Sanyal, S. Chattopadhyay, P. K. Bhowmick, P. P. Das, Automatic generation of research highlights from scientific articles, in: Proceedings of the 2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE 2021), co-located with JCDL 2021, CEUR Workshop Proceedings, 2021, pp. 1–8. URL: http://ceur-ws.org/Vol-2936/paper3.pdf.

[5] T. Rehman, D. K. Sanyal, P. Majumder, S. Chattopadhyay, Named entity recognition based automatic generation of research highlights, CoRR abs/2303.12795 (2023). URL: https://doi.org/10.48550/arXiv.2303.12795. doi:10.48550/ARXIV.2303.12795. arXiv:2303.12795.

[6] T. Rehman, D. K. Sanyal, S. Chattopadhyay, Research highlight generation with elmo contextual embeddings, Scalable Comput. Pract. Exp. 24 (2023) 181–190. URL: https://doi.org/10.12694/scpe.v24i2.2238. doi:10.12694/SCPE.V24I2.2238.

[7] T. Rehman, D. K. Sanyal, S. Chattopadhyay, P. K. Bhowmick, P. P. Das, Generation of highlights from research papers using pointer-generator networks and scibert embeddings, IEEE Access 11 (2023) 91358–91374. URL: https://doi.org/10.1109/ACCESS.2023.3292300. doi:10.1109/ACCESS.2023.3292300.

[8] X. Yi, Y. Xinyi, C. Zhang, Enhancing keyword extraction from academic articles using highlights, Proceedings of the Association for Information Science and Technology 61 (2024) 1147–1149.

[9] N. Bölücü, Y. C. Bilge, D. Çetintas, Z. Yücel, Modest: A dataset for multi domain scientific title generation, Knowl. Based Syst. 321 (2025) 113557. URL: https://doi.org/10.1016/j.knosys.2025.113557. doi:10.1016/J.KNOSYS.2025.113557.

[10] T. Rehman, D. K. Sanyal, S. Chattopadhyay, P. K. Bhowmick, P. P. Das, Automatic generation of research highlights from scientific abstracts, in: C. Zhang, P. Mayr, W. Lu, Y. Zhang (Eds.), Proceedings of the 2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE 2021) co-located with JCDL 2021, Virtual Event, September 30th, 2021, volume 3004 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 69–70. URL: https://ceur-ws.org/Vol-3004/paper10.pdf.

[11] T. Rehman, S. Chattopadhyay, D. K. Sanyal, Abstractive summarization of scientific documents: Models and evaluation techniques, in: D. Ganguly, S. Majumdar, B. Mitra, P. Gupta, S. Gangopadhyay, P. Majumder (Eds.), Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE 2023, Panjim, India, December 15-18, 2023, ACM, 2023, pp. 121–124. URL: https://doi.org/10.1145/3632754.3632771. doi:10.1145/3632754.3632771.

[12] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020) 140:1–140:67. URL: https://jmlr.org/papers/v21/20-074.html.

[13] M. Guo, J. Ainslie, D. C. Uthus, S. Ontañón, J. Ni, Y. Sung, Y. Yang, Longt5: Efficient text-to-text transformer for long sequences, CoRR abs/2112.07916 (2021). URL: https://arxiv.org/abs/2112.07916. arXiv:2112.07916.

[14] J. Zhang, Y. Zhao, M. Saleh, P. J. Liu, PEGASUS: pre-training with extracted gap-sentences for abstractive summarization, in: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 11328–11339. URL: http://proceedings.mlr.press/v119/zhang20ae.html.

[15] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text summarization branches out, 2004, pp. 74–81.

[16] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, Bertscore: Evaluating text generation with BERT, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020. URL: https://openreview.net/forum?id=SkeHuCVFDr.

[17] S. Banerjee, A. Lavie, METEOR: an automatic metric for MT evaluation with improved correlation with human judgments, in: J. Goldstein, A. Lavie, C. Lin, C. R. Voss (Eds.), Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005, Association for Computational Linguistics, 2005, pp. 65–72. URL: https://aclanthology.org/W05-0909/.