Generating Research Highlights from Scientific Literature: Findings from the FIRE 2025 SciHigh Track

Tohida Rehman[1,*], Debarshi Kumar Sanyal[2] and Samiran Chattopadhyay[1,3]

[1]Jadavpur University, Kolkata, India.
[2]Indian Association for the Cultivation of Science, Kolkata, India.
[3]Techno India University, Kolkata, India.

Abstract

Scientific papers normally contain an abstract which gives a summary of the paper, along with the full paper and a brief statement of sections such as the introduction, literature survey, methodology, results, and conclusion & future work. But recent trends provide bulleted points summarizing the paper, known as highlights, which give readers a quick overview of the core findings along with the abstract and the full paper with other sections. The "SciHigh" track at FIRE 2025 addresses a key challenge in scientific research: how to automatically generate concise, meaningful, and informative highlights from research papers. The goal is to accurately capture the main contributions of a paper, enabling readers to quickly grasp its essential findings, especially on handheld devices. The participants were provided with the MixSub dataset [1], which consists of abstracts paired with their original author-written highlights. This paper presents an overview of the SciHigh track, examining the methodologies used by participating teams, the evaluation metrics applied, and the major trends observed in the results.

Keywords

abstractive summarization, natural language generation, scientific data, pre-trained language models, highlight generation.

1. Introduction

The overwhelming rate of growth of scientific publications [2] necessitates tools that can extract the main research findings from papers and present them in an easily accessible manner to readers. According to the report [3], the number of scientific publications roughly doubles every nine years, resulting in a huge volume of papers across fields and sub-fields. Automatic text summarization can play a crucial role in addressing this challenge by generating condensed summaries of long documents. There are two primary approaches to text summarization: extractive and abstractive. Extractive document summarization systems generate a summary by directly selecting key phrases or sentences from a source document. In contrast, abstractive summarization systems first attempt to understand the whole document, paraphrase important sections, and generate new sentences that convey the main ideas [4]. Nowadays, many publishers require authors to provide bulleted points summarizing the main contributions of a research paper, in addition to the abstract and the full paper. In this context, the abstract and the author-written highlights may be regarded as summaries of the main paper. Highlights can also be viewed as a more compact version of the abstract. Compared to a continuous, long paragraph, they are easier to view and read on handheld devices. Research highlights from scientific papers can also be potentially utilized for a variety of applications, including automatic paper title generation [5], taxonomy construction for scholarly corpora [6], design of question-answering datasets and systems [7], and keyphrase indexing for academic search engines [8].

The FIRE 2025 SciHigh Shared Track focuses on the automatic generation of research highlights for scientific papers. In this track, participants were challenged to generate concise and informative bullet point-style summaries directly from paper abstracts. Twelve teams participated in the track and

Forum for Information Retrieval Evaluation, December 17-20, 2025, India
*Corresponding author.
tohidarehman.it@jadavpuruniversity.in (T. Rehman); debarshi.sanyal@iacs.res.in (D. K. Sanyal); samirancju@gmail.com (S. Chattopadhyay)
0000-0002-3578-1316 (T. Rehman); 0000-0001-8723-5002 (D. K. Sanyal); 0000-0002-8929-9605 (S. Chattopadhyay)

explored a variety of approaches to achieve this goal. Ultimately, ten teams submitted working notes. The dataset provided to them was MixSub [1], which is a multi-disciplinary corpus of research papers written in English. Overall, this track aims to achieve the following objectives:

1. To present innovative and effective methods for the automatic generation of research highlights that faithfully capture a paper's main contributions. Bullet-point highlights are easier to read and interpret than longer descriptive paragraphs, especially on mobile or handheld devices.
2. To reduce the time and effort required for readers to understand the key contributions of scientific articles.
3. To study the feasibility of generating concise, author-like research highlights directly from scientific abstracts.
4. To systematically evaluate and compare different approaches for scientific highlight generation using automatic evaluation metrics.

2. Literature Review

Automatic text summarization has been studied for decades, beginning with some of the earliest extractive approaches. Luhn et al. [9] pioneered a method in 1958 that selected sentences based on the frequency of significant words while discarding common terms. This early work established the foundation for extractive summarization, where the task is to select existing sentences from a document to represent its content. Over time, extractive methods evolved with more sophisticated heuristics and statistical models to better identify important sentences. Later, it was extended with the position of a sentence, cue words and many more [10, 11]. Later Sankarasubramaniam et al. [12] proposed an innovative summarization technique integrating Wikipedia with graph-based ranking. They construct a bipartite graph linking sentences and concepts, and iteratively rank sentences to generate summaries. All are extractive approaches.

An abstractive summarization approach was used by Ganesan et al. [13], who developed the "Opinosis" method that builds a word-level graph and identifies high-scoring paths to select concise summaries. The sequence-to-sequence architecture introduced by Sutskever et al. [14] marked a major progress in abstractive summarization systems, substantially enhanced the performance of the systems. Bahdanau et al. [15] showed that combining an attention-based encoder with beam-search decoding improves abstractive summarization on datasets such as DUC 2004. Chopra et al. [16] later introduced the "conditional recurrent neural network" model, which they tested on the Gigaword Corpus and DUC 2004 to further enhance summarization quality. Building on this line of work, Nallapati et al. [17] and See et al. [18] proposed encoder–decoder systems that use attention, copying mechanisms, and coverage to handle context, rare words, and overcome the problem of repetition more effectively. Li et al. [19] presented a seq2seq model with a generative decoder to learn hidden structures in summaries. They used variational inference to deal with the latent variables. Wei et al. [20] proposed a regularization approach for seq2seq summarization that improves semantic consistency and leads to more accurate outputs. Transformers [21] models driven state-of-the-art progress in NLP by enabling large-scale pre-training that can be fine-tuned for many NLP tasks. Models like T5 [22], GPT [23], BART [24], and PEGASUS [25] set strong benchmarks in summarization. Their broad pre-training helps them perform well when adapted to specific domains fine-tuning. Rehman et al. [26] developed a GRU-based encoder-decoder with Bahdanau attention to generate concise english news summaries, achieving improved performance for headline style outputs. Rehman et al. [26] proposed a GRU-based encoder–decoder model for abstractive summarization, enhanced with Bahdanau attention to handle long input sequences in an efficient way. Rehman et al. [27] further evaluated pre-trained models such as facebook/bart-large-cnn, google/pegasus-cnndailymail,and T5-base models across multiple datasets, including CNN-DailyMail, SAMSum, and BillSum, to benchmark summarization performance. LLMs have boosted abstractive summarization showed that smaller models, trained with LLM-based contrastive learning, can approach LLM performance in automated metrics but still trail in human evaluations[28]. Sahba et al. [29] improved summarization using fuzzy features with an attention-based

seq2seq model, while Bhattacharya et al. [30] compared multiple neural and transformer models using standard evaluation metrics.

While text summarization has been studied extensively, generating highlights for research papers differs greatly from conventional document summarization, focusing on short bullet points that clearly showcase the paper's main contributions. Scientific papers have a structured, template-like format with predictable sections and cue words [31]. Early extractive summarization used small datasets, like 188 document-summary pairs from 21 publications [32], with the first trainable ML method. Paice et al. [33] proposed automatic abstract generation by extracting key phrases to capture important content. Contractor et al. [34] introduced an extractive summarization approach that leverages the argumentative zones framework in academic papers. Cohan et al. [35] created an abstractive system to generate summaries of scientific papers and compiled the arXiv and PubMed datasets for evaluation.

Collins et al. [36] used supervised classification to identify worthy sentences as highlights from the research paper. They also introduced the CSPubSum dataset, which contains around 10,000 URLs of scientific articles. Their approach helped automate the extraction of concise, informative highlights from papers. Cagliero et al. [37] introduced an extractive method that uses gradient boosting to pick the top-k sentences as research highlights. This approach ranks sentences by importance rather than labeling them simply as highlights or not. They tested their method on CSPubSum and two specialized datasets, AIPubSumm and BioPubSumm, gathered from ScienceDirect using AI and biomedical keywords.

Rehman et al. [38] first proposed an abstractive approach using pointer-generator networks with GloVe embeddings to generate highlights directly from research abstracts. This method is abstractive because it went beyond extraction and attempted to generate highlights that were concise, coherent, and aligned with the abstract. To further improve this approach, they incorporated Named Entity Recognition (NER) [39] into the highlight generation pipeline, showing that domain-specific knowledge could enhance the informativeness of generated highlights. Further studies [40] explored contextual embeddings like ELMo and different input combinations, including abstracts and other sections, to generate concise and coherent highlights. Their work advanced abstractive highlight generation other than simple extraction. In a recent development, they integrated SciBERT embeddings with a pointer-generator network enhanced by a coverage mechanism and introduced the MixSub dataset, which comprises research articles spanning multiple disciplines [1].

This track focuses on testing different methods for creating highlights from the MixSub dataset, offering a standard way to compare their performance across various domains.

3. Dataset

For the highlight generation task, we utilize the **MixSub** dataset [1], a multi-disciplinary corpus designed for automatic research highlight generation, which contains 19,785 research articles from multiple domains published in 2020 on ScienceDirect[1]. The dataset covers several academic fields, including Biological Sciences, Chemistry, Energy, Management, Nursing, Physics, and Social Sciences. Each article provides an *abstract* along with a set of author-written *highlights*. The dataset is divided into training, validation, and test sets in an 80:10:10 ratio. For the SciHigh track at FIRE 2025, we sampled 10,000 instances for training out of 15,960 available training instances in the original MixSub dataset. However, we retain the original 1,985 validation instances and 1,840 test instances. Figure 1 presents an example entry from the MixSub dataset.

4. Task Description

The SciHigh track addresses the problem of automatically generating research highlights from scientific paper abstracts using the MixSub dataset [1]. While abstracts provide a summary of the paper, research highlights offer a more concise and structured overview of its main contributions. The objective of this

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

Abstract: “ This paper presents the development and application of systematic framework to perform new and retrofit phenomena based synthesis intensification generating well known existing as well as novel intensified solutions . The fundamental pillars of this framework are generic definition expansion and use of Phenomena Building Blocks that include all possible phases identification of principle PBBs using physical property and thermodynamic insights and generation of phenomena based superstructure to systematically identify novel innovative and intensified flowsheet alternatives. The generated flowsheet options are ranked based on Enthalpy Index to identify promising candidates for detailed analysis. One of the key features of this framework that distinguishes it from preceding methods for phenomena based synthesis is that the flowsheet alternatives are synthesized from phenomena based superstructure rather than relying on a base case for phenomena and or hotspots identification and replacement of base case unit operations accordingly . That is a phenomena based superstructure is generated to directly identify intensified solutions . New phenomena their classes and systematic algorithms are developed in order to generate novel intensified solutions. These developments and systematic framework are illustrated through case study involving production of Dimethyl Ether . The results confirm that using this approach promising alternatives with novel unit operations are generated in systematic way.”

Author-written research highlights:

- “A framework for both new and retrofit phenomena based synthesis intensification. ”
- “ Phenomena based superstructure generation to identify novel feasible solutions.”
- “Reduction of alternatives using feasibility and logical rules.”
- “Ranking of feasible alternatives using enthalpy index EI to identify promising solutions.”
- “Applicability demonstrated for the production of DME as a case study.”

Figure 1: An example of an abstract and its corresponding author-written research highlights from the MixSub dataset. The colored boxes indicate the correspondence between portions of the abstract and the author-written highlights. While some highlights directly reflect text from the abstract, others integrate information from multiple sentences and rephrase it. This demonstrates the challenges of automatically generating accurate highlights from abstracts. The abstract and the author-written research highlights are taken from https://www.sciencedirect.com/science/article/pii/S0255270120305651.

task is to design machine learning models that can generate high-quality highlights closely matching those authored by researchers.

Participants were encouraged to experiment with a variety of techniques, such as basis machine learning techniques, retrieval-augmented models, transformer-based architectures, and fine-tuned large language models.

This task aims to improve both the efficiency and quality of automatic highlight generation, thereby helping researchers quickly identify the key contributions of papers and supporting enhanced academic search and indexing systems.

5. Performance Evaluation Metrics

To evaluate the quality of the generated research highlights, we employed standard automatic evaluation metrics commonly used in text summarization tasks. These metrics assess the similarity between the model-generated research highlights (`MGHighlights`) and the corresponding author-written highlights (`ARHighlights`). Although ROUGE-1, ROUGE-2, and ROUGE-L were computed, the final ranking of submissions was based on the ROUGE-L metric. The evaluation metrics are described below.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [41] is a widely used metric for summarization evaluation. ROUGE-n quantifies the lexical and structural similarity between the model-generated research highlights (`MGHighlights`) and the corresponding author-written highlights

(ARHighlights), where an n-gram is defined as a contiguous sequence of n words.

1. **ROUGE-1** measures the unigram overlap between the model-generated research highlights (MGHighlights) and the corresponding author-written highlights (ARHighlights), indicating how effectively the generated highlights capture essential keywords and core concepts from the reference text.
2. **ROUGE-2** computes the bigram overlap between the model-generated research highlights (MGHighlights) and the corresponding author-written highlights (ARHighlights). It finds insights into the preservation of local word order, contextual consistency, and linguistic coherence.
3. **ROUGE-L** is based on the longest common subsequence between the model-generated research highlights (MGHighlights) and the corresponding author-written highlights (ARHighlights). It evaluates similarity in sentence structure, information ordering, and overall discourse flow.

For each ROUGE-n variant, recall, precision, and F1-score are computed as defined in Equations (1), (2), and (3).

Recall (R) is defined as:

$$R = \frac{\text{Number of matched } n\text{-grams}}{\text{Total } n\text{-grams in ARHighlights}} \tag{1}$$

Precision (P) is defined as:

$$P = \frac{\text{Number of matched } n\text{-grams}}{\text{Total } n\text{-grams in MGHighlights}} \tag{2}$$

The F1-score ($F1$), which provides a harmonic balance between recall and precision, is computed as:

$$F1 = 2 \times \frac{R \times P}{R + P} \tag{3}$$

6. Participation and Evaluation

Initially, the training and validation datasets were released to the track participants. Subsequently, the test set, consisting of 1,840 instances of abstracts, was released with author-written highlights masked out. After submissions were received, the complete test set containing the author-written highlights was released for evaluation. All submissions were assessed using the ROUGE-1, ROUGE-2, and ROUGE-L metrics. To maintain consistency, the evaluation code was shared with all teams, allowing them to verify their model performance. The final ranking of the submissions was determined based on the ROUGE-L F1 scores.

Fourteen teams from various universities, colleges, and research institutions registered for the SciHigh track. However, twelve teams submitted their runs along with trained models, providing up to two runs each in CSV format containing the predicted highlights. Eventually, ten teams registered for the conference and submitted their working notes.

The participating teams utilized diverse strategies, such as extractive techniques, abstractive approaches, hybrid of extractive and abstractive approaches, and fine-tuning of pre-trained language models. Table 1 presents a summary of the results, highlighting the performance of all systems based on ROUGE-L F1 scores.

Two approaches were explored by the **Text_highlights_gen** team, namely the standard Pegasus model and a version incorporating NER features. Fine-tuning was performed over 10 epochs with batch size 2, employing Adam at a learning rate of $2e{-}5$ for both models. Beam search with a width of 4 was employed to improve sequence generation. The input length was limited to 512 tokens, and the output length was limited to 100 tokens. The fine-tuned Pegasus model achieved a ROUGE-L F1 score of 23.45%, ranking first in the SciHigh track at FIRE 2025.

The **AiNauts** team explored two strategies for highlight generation. The first method combined extractive and abstractive techniques: sentences were ranked using TF–IDF, Sentence-BERT embeddings, cosine similarity, and MMR, followed by abstractive rewriting with a fine-tuned

Table 1
Performance of participating teams in the SciHigh track based on ROUGE-L F1 scores. The best run for each team is reported.

Group Name	Run Submission (Best Run)	ROUGE-L F1	Rank
Text_highlights_gen	run1	23.45	1
AiNauts	run1	23.24	2
SVNIT_CSE	run1	23.02	3
NLPFusion	run2	22.96	4
The NLP Explorers	run2	22.94	5
NIT_PATNA_2025	run1	22.42	6
MUCS	run1	22.08	7
JU_CSE_PR_KS	run1	22.06	8
SCaLAR	run1	20.33	9
Ayanika	run1	17.91	10

`Facebook/bart-large-cnn` model. The second method `DistilBERT-base-uncased` model used for binary sentence classification, selecting sentences with probabilities above 0.5. Both methods were fine-tuned for three epochs; Method 1 used a batch size of 8, whereas Method 2 used 16. The first method yielded the best performance, achieving a ROUGE-L F1 score of 23.24% and securing second place in the SciHigh track at FIRE 2025.

Team **SVNIT_CSE** employed an ensemble of transformer-based summarization models, including `facebook/bart-large-cnn`, `t5-base`, `google/long-t5-tglobal-base`, `allenai/led-base-16384`, and `google/pegasus-pubmed`. Models BART, T5, and Long-T5 were fine-tuned with a batch size of 8, but batch size as 4 used for LED and Pegasus models. Maximum input lengths were set as 512 for T5, 516 for Long-T5, 384 for BART, 516 for Pegasus, 2048 for LED, and with predicted highlights limited to 64 tokens. The `bart-large-cnn` model achieved the best performance, attaining a ROUGE-L F1 score of 23.02% and securing third place among 12 participants in the SciHigh track.

The Pegasus model was fine-tuned for abstractive summarization by the **NLPFusion** team using Low-Rank Adaptation (LoRA).Evaluation was performed on the `Pegasus-PubMed` and `Pegasus-ArXiv` models using 256-token inputs and 64-token outputs, with fine-tuning carried out under the same conditions for comparability. Among all submissions, `Pegasus-PubMed` enhanced with LoRA performed best, attaining a ROUGE-L F1 score of 22.96% and securing fourth place in the SciHigh track.

Team **The NLP Explorers** fine-tuned the `T5-base` and `BART-base` models over 5 epochs, using a learning rate of $2e{-}5$ and batch size of 8. Input abstracts were limited to 512 tokens, and generated highlights to 100 tokens. Under these settings, the fine-tuned `T5-base` model performed best, achieved a ROUGE-L F1 score of 22.94% and and securing fifth place in the SciHigh track.

Team **NIT_PATNA_2025** initially operated as two sub-teams but later combined their results into a single submission. They evaluated `T5-small` and a `LongT5-based` extended model, both fine-tuned for 10 epochs following the same configuration as the **Text_highlights_gen** team. The `T5-small` model achieved the best performance, obtaining a ROUGE-L F1 score of 22.42% and securing sixth place in the SciHigh track at FIRE 2025.

Team **MUCS** fine-tuned a `T5-base` model for 2 epochs, using a maximum input length of 512 tokens, a learning rate of $3e{-}4$, an output limit of 128 tokens and a batch size of 4 for both fine-tuning and evaluation. Their model achieved a ROUGE-L F1 score of 22.08%, securing seventh place in the SciHigh track at FIRE 2025.

For the SciHigh track, team **JU_CSE_PR_KS** applied two approaches: a binary XGBoost classifier and a regression-based XGBoost model. The classifier labeled sentences as relevant or not based on overlap with reference highlights, while the regressor provided graded similarity scores to rank sentences more precisely. The top-ranked sentences were chosen as highlights. The regression model achieved a ROUGE-L F1 score of 22.06%, placing the team eighth in the track at FIRE 2025.

Team **SCaLAR** developed an automated highlight-generation pipeline that combines SciBERT based entity extraction method, keyword extraction with KeyBERT, sentence ranking through token budgeting, and supervised fine-tuning of LLaMA. A retrieval-augmented strategy was also tested with BART, SciBART, and T5, and employing Facebook AI Similarity Search (FAISS) to find similar examples that helps the generation process. The team's best setup (V6), which included trimmed abstracts, guided constraints, and reference-aligned filtering, achieved a ROUGE-L F1 score of 20.33% and ranked ninth in the SciHigh track.

Team **Ayanika** fine-tuned the pre-trained `T5-small` model for highlight generation. This model achieved a ROUGE-L F1 score of 17.91%, placing the team tenth in the SciHigh track at FIRE 2025.

6.1. Model Usage and Trends

Table 2 summarizes the frequency of model families used by participating teams. The distribution of model choices suggests a strong leveraging on encoder–decoder transformer architectures for highlight generation. The `T5` family was the most widely adopted, used by eight teams, followed by `BART` and `Pegasus` variants, each appearing in five submissions. This reflects a preference for models that can be easily fine-tuned for abstractive generation with limited architectural modification. In contrast, comparatively fewer teams explored long-context models such as LED or instruction-tuned large language models like LLaMA-2. Traditional machine learning approaches, including XGBoost-based classifiers and regressors, were used by a small number of teams, primarily for extractive sentence selection. Overall, the trend indicates that while transformer-based abstractive models dominate the task, hybrid and non-neural approaches remain relevant alternatives.

Table 2
Frequency of model families used by participating teams in the SciHigh track.

Model Family	Number of Teams
T5 variants	8
BART variants	5
Pegasus variants	5
LED / Longformer-based	2
LLaMA-2	1
BERT variants (DistilBERT)	1
SciBERT	1
XGBoost (Classifier)	1
XGBoost (Regressor)	1

7. Conclusion and Future Directions

In this work, we addressed the problem of automatically generating research highlights from abstracts, using the MixSub dataset introduced in the SciHigh track at FIRE 2025. This track provided a standardized dataset, a uniform evaluation framework, and benchmark results for systematically exploring this task.

Our analysis of the submissions received from the ten participating teams revealed interesting insights into the various solutions. In particular, we observed that fine-tuned transformer-based models, particularly Pegasus, BART, and T5 variants, are highly effective on domain-specific data such as MixSub for generating research highlights. Hybrid approaches that combine extractive sentence selection with abstractive rewriting also showed competitive performance, suggesting that integrating content selection and generation can be beneficial. At the same time, the modest and relatively close ROUGE-L scores among the top-performing systems indicate that the task remains challenging, especially given the need to generate short, accurate, and non-redundant bullet points that align closely with author-written highlights.

For future research, several directions are promising. These include developing cross-domain and multilingual highlight generation models, integrating retrieval-augmented generation to incorporate external scientific knowledge, and exploring reinforcement learning or contrastive learning to optimize highlight informativeness and coherence. Additionally, expanding evaluation beyond standard ROUGE metrics to include semantic similarity and human-centered assessments could provide a more comprehensive understanding of model performance.

Overall, this work contributes to establishing benchmarks and best practices for automatic research highlight generation, aiming to support researchers in efficiently navigating scientific literature and improving the accessibility of knowledge across academic platforms. We hope that the SciHigh track will continue to be organized in the coming years, fostering progress in this area and contributing toward more accessible, efficient, and user-friendly scientific communication.

Declaration on Generative AI

Generative AI tools were employed solely to aid in language polishing and formatting for specific sections of this manuscript. All scientific content, experimental design, data collection, analysis, and interpretation were independently developed and verified by the author(s). The AI tools did not participate in experiment planning, coding, data processing, or drawing conclusions.

References

[1] 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.

[2] L. Bornmann, R. Haunschild, R. Mutz, Growth rates of modern science: A latent piecewise growth curve approach to model publication numbers from established and new literature databases, Humanities and Social Sciences Communications 8 (2021) 1–15.

[3] R. Van Noorden, Global scientific output doubles every nine years, Nature news blog (2014). URL: https://blogs.nature.com/news/2014/05/global-scientific-output-doubles-every-nine-years.html.

[4] W. S. El-Kassas, C. R. Salama, A. A. Rafea, H. K. Mohamed, Automatic text summarization: A comprehensive survey, Expert Systems with Applications 165 (2021) 113679.

[5] T. Rehman, D. K. Sanyal, S. Chattopadhyay, Can pre-trained language models generate titles for research papers?, in: Proceedings of the 26th International Conference on Asian Digital Libraries (ICADL), Springer, 2024, pp. 154–170.

[6] A. Lahiri, Y. Hou, D. K. Sanyal, TaxoAlign: Scholarly taxonomy generation using language models, in: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2025, pp. 30191–30211.

[7] A. Lahiri, D. K. Sanyal, I. Mukherjee, NLP-QA: A large-scale benchmark for informative question answering over natural language processing documents, in: Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM), 2025, pp. 6444–6450.

[8] A. Lahiri, D. K. Sanyal, I. Mukherjee, A keyphrase-centric search engine for scientific papers, in: Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation, 2023, pp. 125–128.

[9] H. P. Luhn, The Automatic Creation of Literature Abstracts, IBM Journal of Research and Development 2 (1958) 159–165.

[10] H. P. Edmundson, New methods in automatic extracting, Journal of the ACM (JACM) 16 (1969) 264–285.

[11] P. B. Baxendale, Machine-made index for technical literature—an experiment, IBM Journal of Research and Development 2 (1958) 354–361.

[12] Y. Sankarasubramaniam, K. Ramanathan, S. Ghosh, Text summarization using wikipedia, Information Processing & Management 50 (2014) 443–461.
[13] K. Ganesan, C. Zhai, J. Han, Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions, in: Proceedings of the 23rd International Conference on Computational Linguistics (ACL), Association for Computational Linguistics, 2010, pp. 340–348.
[14] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NeurIPS), Vol. 2, MIT Press, Cambridge, MA, USA, 2014, p. 3104–3112.
[15] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: Proceedings of the International Conference on Learning Representations (ICLR), 2015.
[16] S. Chopra, M. Auli, A. M. Rush, Abstractive sentence summarization with attentive recurrent neural networks, in: Proceedings of the 2016 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2016, pp. 93–98.
[17] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, B. Xiang, Abstractive text summarization using sequence-to-sequence RNNs and beyond, in: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 280–290.
[18] A. See, P. J. Liu, C. D. Manning, Get to the point: Summarization with pointer-generator networks, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1073–1083.
[19] P. Li, W. Lam, L. Bing, Z. Wang, Deep recurrent generative decoder for abstractive text summarization, in: M. Palmer, R. Hwa, S. Riedel (Eds.), Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 2091–2100.
[20] B. Wei, X. Ren, Y. Zhang, X. Cai, Q. Su, X. Sun, Regularizing output distribution of abstractive chinese social media text summarization for improved semantic consistency, ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) 18 (2019) 1–15.
[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in Neural Information Processing Systems (NeurIPS) 30 (2017).
[22] 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, Journal of Machine Learning Research (JMLR) 21 (2020) 1–67.
[23] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training, OpenAI Blog (2018).
[24] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), Association for Computational Linguistics, Online, 2020, pp. 7871–7880.
[25] J. Zhang, Y. Zhao, M. Saleh, P. Liu, PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization, in: Proceedings of the International Conference on Machine Learning (ICLR), PMLR, 2020, pp. 11328–11339.
[26] T. Rehman, S. Das, D. K. Sanyal, S. Chattopadhyay, Abstractive text summarization using attentive gru based encoder-decoder, in: Applications of Artificial Intelligence and Machine Learning: Select Proceedings of ICAAAIML 2021, Springer, 2022, pp. 687–695.
[27] T. Rehman, S. Das, D. K. Sanyal, S. Chattopadhyay, An analysis of abstractive text summarization using pre-trained models, in: Proceedings of the International Conference on Computational Intelligence, Data Science and Cloud Computing: IEM-ICDC 2021, Springer, 2022, pp. 253–264.
[28] T. Goyal, J. J. Li, G. Durrett, News summarization and evaluation in the era of GPT-3, arXiv preprint arXiv:2209.12356 (2022).

[29] R. Sahba, N. Ebadi, M. Jamshidi, P. Rad, Automatic text summarization using customizable fuzzy features and attention on the context and vocabulary, in: Proceedings of the 2018 World Automation Congress (WAC), IEEE, 2018, pp. 1–5.
[30] A. Bhattacharya, T. Rehman, D. K. Sanyal, S. Chattopadhyay, Comparative analysis of abstractive summarization models for clinical radiology reports, arXiv preprint arXiv:2506.16247 (2025).
[31] A. Kazantseva, S. Szpakowicz, Summarizing short stories, Computational Linguistics 36 (2010) 71–109.
[32] J. Kupiec, J. Pedersen, F. Chen, A trainable document summarizer, in: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1995, pp. 68–73.
[33] C. D. Paice, The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases, in: Proceedings of the 3rd Annual ACM Conference on Research and Development in Information Retrieval, SIGIR '80, Butterworth & Co., GBR, 1980, p. 172–191.
[34] D. Contractor, Y. Guo, A. Korhonen, Using argumentative zones for extractive summarization of scientific articles, in: Proceedings of COLING 2012, 2012, pp. 663–678.
[35] A. Cohan, F. Dernoncourt, D. S. Kim, T. Bui, S. Kim, W. Chang, N. Goharian, A discourse-aware attention model for abstractive summarization of long documents, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 615–621.
[36] E. Collins, I. Augenstein, S. Riedel, A supervised approach to extractive summarisation of scientific papers, in: Proc. 21st Conf. on Computational Natural Language Learning (CoNLL), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 195–205.
[37] L. Cagliero, M. La Quatra, Extracting highlights of scientific articles: A supervised summarization approach, Expert Systems with Applications 160 (2020) 113659.
[38] T. Rehman, D. K. Sanyal, S. Chattopadhyay, P. K. Bhowmick, P. P. Das, Automatic generation of research highlights from scientific abstracts, in: 2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE 2021), collocated with JCDL 2021, 2021.
[39] T. Rehman, D. K. Sanyal, P. Majumder, S. Chattopadhyay, Named entity recognition based automatic generation of research highlights, in: Proceedings of the Third Workshop on Scholarly Document Processing (SDP 2022) collocated with COLING 2022, Association for Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 163–169.
[40] T. Rehman, D. K. Sanyal, S. Chattopadhyay, Research highlight generation with ELMo contextual embeddings, Scalable Computing: Practice and Experience 24 (2023) 181–190.
[41] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81.