Entity and Keyword Guided Highlight Generation from Scientific Abstracts Using Large Language Models

Nishalini K[1,†], Sreenath K A[1,†], Anand Kumar M[2] and Navya Binu[2]

[1]Shiv Nadar University Chennai, Tamil Nadu, India
[2]Department of Information Technology, National Institute of Technology Karnataka, Surathkal, India

Abstract

Recent years have seen exponential growth in scientific publications, making it increasingly challenging for researchers to keep pace with new findings across their fields. Abstracts, while informative, are often too lengthy for quick comprehension and discoverability enhancing the need for concise research highlights. We propose an automated pipeline for highlight generation that integrates entity extraction with SciBERT, keyword extraction with KeyBERT, sentence ranking with token budgeting, and supervised fine-tuning of LLaMA. Additionally, we explore a retrieval-augmented generation approach using BART, SciBART and T5 models with FAISS-indexed similar examples to provide contextual guidance during generation. Experiments on the MixSub dataset demonstrate that reference-aligned filtering enhances highlight quality in the constrained approach, achieving ROUGE-1, ROUGE-2 and ROUGE-L F1-scores of 32.03, 9.25 and 20.33, respectively, METEOR score of 27.31, while the retrieval-augmented method shows BART consistently outperforming SciBART and T5. Both approaches offer scalable solutions to information overload in scientific publishing.

Keywords

Highlight Generation, Generative AI, SciBERT, KeyBERT, LLaMA, BART, SciBART, T5

1. Introduction

Recent studies show that the total number of articles indexed in major databases have grown approximately 47% between 2016 and 2022 [1], far outpacing the growth of practicing scientists and dramatically increasing the publication workload per researcher. This volume makes comprehensive literature review increasingly challenging, particularly when the abstracts themselves can be lengthy and complex.

To address this, many journals have adopted research highlights. Research highlights are concise bullet-point summaries of key findings that emerge as a critical tool for rapid comprehension. However, creating such highlights manually for the vast and continuously growing body of literature is impractical, motivating the need for automated solutions.

Recent advances in natural language processing (NLP) and large language models (LLMs) present promising avenues for this challenge. Building on the foundational work by Rehman et al. [2] that established benchmark performance using pointer-generator networks with SciBERT embeddings, this work continues to explore automated highlight generation approaches. Yet, unconstrained generation from LLMs often risks factual inaccuracies, omissions, or hallucinations. A key insight driving this work is that research highlights can be made more accurate and focused when LLMs are guided explicitly by structured signals extracted from abstracts. Specifically, entities and domain-specific keywords can serve as constraints, anchoring the generation process to verifiable content and reducing the likelihood of irrelevant or fabricated information.

In this paper, we introduce an end-to-end pipeline for automated highlight generation from scientific abstracts using the MixSub-SciHigh dataset. Our approach integrates four components: (1) Named Entity Recognition, (2) Keyword Extraction, (3) Sentence Ranking with token budgeting, and (4) Supervised Fine-Tuning of Large Language Models [3] for guided highlight generation. Additionally, we explore an alternative prompt-driven retrieval-augmented generation approach that involves prompting and

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

† Both authors contributed equally to this research.

nishalinikarthik17@gmail.com (N. K); sreenathka2004@gmail.com (S. K. A)
0009-0006-4274-6996 (N. K); 0009-0007-7256-0858 (S. K. A)

Fine-tuning the BART, T5, and SciBART models to create highlights by obtaining similar abstract highlight pairs from Faiss vector stores. One variant of the vector store utilizes sentence-transformer embeddings for BART and T5, whereas another uses Specter2 embeddings for SciBART to facilitate similarity matching via Faiss.

We evaluate both approaches using standard summarization metrics ROUGE [4] and METEOR [5]. By combining domain-specific constraints with LLM-based summarization and exploring retrieval-augmented alternatives, this work contributes frameworks that balance accuracy, efficiency, and scalability in processing the ever-expanding corpus of scientific knowledge.

2. Literature Review

The field of automatic text summarization has undergone a significant transformation over the past decades, evolving from simple rule-based extraction to sophisticated neural generation systems. The earliest approaches, as illustrated by Kupiec et al. [6], focused on extractive summarization using statistical features to identify and rank important sentences within documents. However, these methods operated on surface-level features such as sentence position and word frequency, with designs that did not account for semantic relationships or domain-specific linguistic properties. Moreover, while computationally efficient, they were limited by their inability to generate novel text or adapt content to specific summarization requirements.

The researchers then began exploring abstractive summarization methods capable of generating novel text rather than merely extracting sentences. Nallapati et al. [7] applied attentional encoder-decoder Recurrent Neural Networks to abstractive summarization, proposing novel architectural variants to address critical challenges such as modeling keywords, capturing hierarchical sentence structure, and handling rare or unseen words. This foundational work established sequence-to-sequence models as a viable approach for abstractive text generation beyond simple extractive methods.

See et al. [8] proposed a hybrid pointer-generator network architecture that augments the standard sequence-to-sequence attentional model. This architecture addresses two shortcomings in neural abstractive summarization: (i) inaccurate reproduction of factual details and handling of out-of-vocabulary words through a hybrid mechanism that can copy words from the source text via pointing while retaining the ability to generate novel words from the vocabulary, and (ii) repetition in generated text through a coverage mechanism that maintains a coverage vector tracking the sum of attention distributions over previous decoder timesteps. This hybrid approach balances extractive and abstractive capabilities within a unified framework.

The integration of pre-trained language models brought significant advances to scientific text processing. Beltagy et al. [9] introduced SciBERT, a BERT-based language model trained on a large corpus of scientific publications from computer science and biomedical domains with a domain-specific vocabulary constructed from scientific text. This architecture design addresses the challenge of processing specialized scientific vocabulary and domain-specific language, which differ substantially from general domain corpora on which standard BERT is trained. SciBERT's design is particularly relevant for research highlight generation, where understanding specialized scientific vocabulary and terminology is necessary for accurate processing of domain-specific content.

Specialized research on highlights generation has emerged as researchers recognize the unique requirements of condensing scientific papers into key findings. Rehman et al. initiated systematic exploration of this task with foundational work [10] establishing pointer-generator networks with coverage mechanisms for automatic generation of research highlights from scientific abstracts, using general word embeddings (GloVe) and seq2seq architectures with BiLSTM encoders. Building on this architecture, subsequent work [11] explored the integration of Named Entity Recognition to treat multi-word entities as single tokens, addressing the challenge of preventing fragmentation of domain-relevant terms during generation. In parallel, analysis of pre-trained summarization models [12] compared the effectiveness of models like PEGASUS, T5, and BART across multiple datasets. The progression continued with [13] investigation of ELMo contextual embeddings as semantic representations, moving beyond general

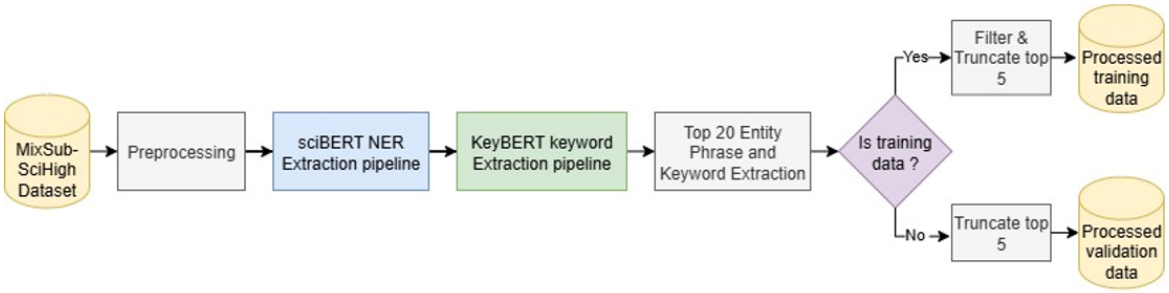

Figure 1: Overall preprocessing and entity-keyword extraction pipeline using SciBERT and KeyBERT.

embeddings to context-sensitive word meanings while maintaining the pointer-generator and coverage framework. This methodological progression led to [2] the combination of SciBERT domain-specific embeddings with pointer-generator networks and coverage mechanisms, enabling the model to probabilistically decide between generating novel words from the vocabulary and copying source terms via attention-based pointing. This architecture combines contextual embeddings from domain-specific language models with hybrid abstractive-extractive mechanisms. Most recently, comprehensive evaluation frameworks have been proposed [14] to assess highlight generation quality through multiple metrics and study the factual consistency of neural summaries. Collectively, these works demonstrate a systematic progression of architectural refinements from basic pointer-generator mechanisms with general embeddings toward hybrid architectures that integrate domain-specific embeddings (ELMo, then SciBERT) and explicit entity-aware tokenization.

Our work extends this progression by introducing two complementary approaches that address current limitations in factual grounding and domain-specific control. We propose explicit entity and keyword constraints derived from fine-tuned domain models, combined with reference-aligned filtering to ensure that generated highlights accurately reflect key scientific concepts. Additionally, we explore retrieval-augmented generation paradigms that leverage semantically similar examples to guide the generation process, offering an alternative pathway for improving highlight quality and relevance.

3. Entity and Keyword Guided Highlight Generation

This section describes how our model produces research highlights from scientific abstracts using four integrated components: Named Entity Recognition with a fine-tuned SciBERT, Keyword extraction using KeyBERT and sentence embeddings, Sentence ranking combined with token budgeting and Supervised fine-tuning of LLaMA. Together, these steps identify key concepts, select the most informative sentences within length limits, and train the model to generate concise, accurate summaries.

3.1. Dataset & Preprocessing

We use the MixSub dataset [2], a multi-disciplinary corpus containing 19,785 scientific research papers from ScienceDirect (published in 2020) with author-written highlights. The dataset spans diverse scientific disciplines, with each example consisting of an abstract paired with highlights. The average abstract length is 148 words, while highlights average 57 words, with 72% of papers having highlights at least 1.5 times shorter than their abstracts.

We apply a simple text cleaning process to prepare the abstracts. We begin by converting all text to lowercase for consistency, then removing standard punctuation and common English stop words using NLTK's stopword list. Short words with fewer than three characters are also filtered out, as these typically don't contribute meaningful information to our analysis. We also perform an additional cleaning step that removes unwanted characters while preserving essential punctuation like commas and periods, which helps maintain the text's readability and structure.

3.2. Named Entity Recognition using Fine-Tuned SciBERT

We have fine-tuned the SciBERT model (`scivocab-uncased`) on a combination of scientific NER datasets-including SciERC (SciIE) [15], BC5DR [16], JNLPBA, and NCBI-Disease [17], to extract domain-specific entities from scientific abstracts. This multi-dataset fine-tuning has improved the model's ability to recognize a wide spectrum of entity types commonly found in scientific literature, namely: [Method, Task, Entity, OtherScientificTerm, Material, Metric, Generic, protein, cell_type, Disease, cell_line, DNA, and RNA]. We have ranked these entity types according to a manually predefined priority schema tailored to the scientific domain. This ranking has reflected the characteristics of the MixSub-SciHigh dataset, which primarily consists of scientific abstracts.

Because SciBERT imposes a 512-token input limit, we have segmented each abstract into overlapping chunks of 500 tokens with a 50-token overlap to ensure that all relevant information has been captured while maintaining contextual continuity across chunks. From these chunks, we have extracted entities and removed duplicates.

Next, we have ranked the extracted entities based on the priority schema and initially selected 20 unique entities per abstract. To ensure that the entity constraints provided to the model have been grounded in the reference output, we have applied a filtering step that retains only those entities explicitly present in the corresponding author-written highlights. We have retained the top five-ranked entities and used them to train the model with the entity constraints. For the validation set, however, we have omitted this filtering step. Figure 2 illustrates this complete NER pipeline.

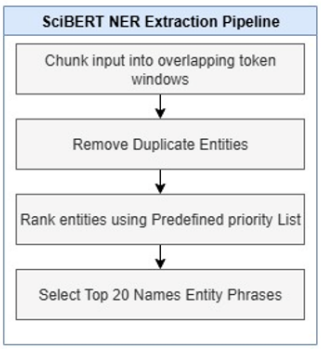

Figure 2: Named Entity Recognition pipeline using fine-tuned SciBERT.

3.3. Keyword Extraction with KeyBERT and Sentence Embeddings

To extract semantically rich bigram keyphrases from scientific abstracts, we employ KeyBERT, a lightweight yet effective keyphrase extraction framework [18]. KeyBERT begins by embedding both the full abstract and its candidate keyphrases into the same high-dimensional semantic space. In our implementation, we use the all-MiniLM-L6-v2 model from the SentenceTransformers library to generate these contextualized embeddings, as it produces compact vectors that accurately capture the meaning of short text spans. Figure 3 illustrates the overall keyword extraction pipeline. This process has three main stages as given below:

1. Candidate Generation: We apply a CountVectorizer over the abstract to form all possible bigram phrases. This yields a set of candidates $C = \{c_1, c_2, \ldots, c_n\}$.
2. Semantic Scoring: Each candidate c_i and the abstract D receive embeddings in the embedding space$\mathbb{R}^d$, denoted

$$D \in \mathbb{R}^d, \quad c_i \in \mathbb{R}^d.$$

We then compute cosine similarities to measure relevance to the document and redundancy

between candidates,

$$S(c_i, D) = \cos(c_i, D), \quad S(c_i, c_j) = \cos(c_i, c_j).$$

3. MaxSum Selection: The goal is to select a subset $K \subseteq C$, with $|K| = k$, that maximizes document relevance while minimizing redundancy:

$$K = \underset{K' \subseteq C,\, |K'|=k}{\arg\max} \left[\sum_{c \in K'} S(c, D) \; - \; \lambda \sum_{\substack{c_i, c_j \in K' \\ i<j}} S(c_i, c_j) \right], \tag{1}$$

where $\lambda \geq 0$ controls the trade-off between relevance and redundancy.

KeyBERT first ranks all bigram candidates by their similarity to the abstract and retains the top 20 as initial choices. It then applies the MaxSum strategy internally to select the final top five keyphrases. For training data, we apply the filtering strategy that retains only keyphrases appearing in the author-written highlights, and this filtering is not applied to validation data.

These keyphrases, each a concise, semantically aligned representation of the abstract's core concepts and entities, are passed on as constraints along with the prompt to guide the subsequent highlight generation stage.

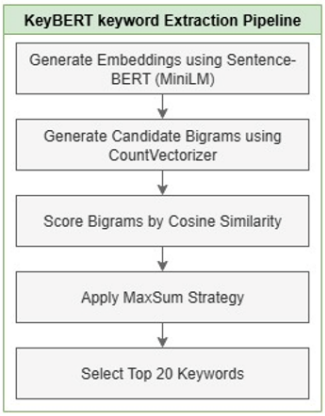

Figure 3: Keyword extraction pipeline using KeyBERT.

3.4. Sentence Ranking and Token Budgeting

LLMs generally have strict input length constraints, and we further face computational resource limitations. We train on Kaggle's T4 GPU, which imposes a 512-token sequence length limit per input to avoid CUDA out-of-memory errors. Consequently, we feed only the most relevant portions of each abstract into the model. To achieve this, we implement a token-budgeting strategy based on sentence ranking with SPECTER embeddings [19]. This approach ensures that we preserve contextually important sentences while adhering to strict input size constraints.

3.4.1. Token Budget Allocation

Let the maximum allowed token length be denoted as $T_{\text{max}} = 512$. This budget is divided among the prompt, the abstract content, and the reserved space for generated highlights. The prompt and constraints together have a length of approximately $T_{\text{prompt}} \approx 120$ tokens. The generated highlight, also referred to as the completion, has a length of $T_{\text{completion}} = 128$ tokens. Consequently, the number of tokens available for the abstract can be calculated as

$$T_{\text{abstract}} = T_{\text{max}} - T_{\text{prompt}} - T_{\text{completion}} \approx 264$$

This means that, after accounting for the prompt and the generated highlight, about 264 tokens remain for composing the abstract.

3.4.2. Sentence Embedding and Similarity Scoring

Each abstract is split into individual sentences using the NLTK sentence tokenizer function. Let the set of abstract sentences be denoted as:

$$\mathcal{A} = \{s_1, s_2, \ldots, s_n\}$$

Using the SPECTER model, each sentence s_i is embedded into a vector $\mathbf{e}_i \in \mathbb{R}^d$. The overall abstract embedding $\bar{\mathbf{e}}$ is computed as the mean of all sentence embeddings:

$$\bar{e} = \frac{1}{n} \sum_{i=1}^{n} e_i \tag{2}$$

We then compute the cosine similarity between each sentence and the abstract mean:

$$\text{similarity}(s_i) = \cos(\bar{e}, e_i) = \frac{\bar{e} \cdot e_i}{\|\bar{e}\| \cdot \|e_i\|} \tag{3}$$

3.4.3. Keyword and Entity Boosting

To improve factual alignment, we further boost sentence scores based on the presence of extracted keywords or named entities. The final sentence importance score is calculated as:

$$\text{score}(s_i) = \text{sim}(s_i) + \alpha \cdot I_{\text{keyword}}(s_i) + \beta \cdot I_{\text{entity}}(s_i) \tag{4}$$

where $I_{\text{keyword}}(s_i)$ equals 1 if s_i contains a keyword and 0 otherwise, and $I_{\text{entity}}(s_i)$ equals 1 if s_i contains a named entity and 0 otherwise, with $\alpha = \beta = 1.5$.

3.4.4. Token-Constrained Sentence Selection

Sentences are sorted in descending order of their final score. Let $\tau(s_i)$ denote the token length of sentence s_i. The sentence subset $\mathcal{S}^* \subseteq \mathcal{A}$ is selected greedily such that:

$$\sum_{s_i \in S^*} \tau(s_i) \leq T_{\text{abstract}} \tag{5}$$

Selected sentences are then reordered to match their original sequence in the abstract to preserve narrative flow and coherence.

This approach allows us to compress abstracts effectively while maintaining coverage of important ideas and terminology.

Boosting based on keywords and named entities ensures that the model's inputs align with the focus of the target highlights. The use of a hard token budget avoids truncation at the tokenizer level, which might otherwise cut off relevant context mid-sentence. By combining semantic importance (via SPECTER) and factual constraints (via entity/keywords), our token budgeting mechanism contributes to both the factuality and efficiency of model training under limited GPU resources.

3.5. Supervised Fine-Tuning

We generate research highlights from scientific abstracts using supervised fine-tuning of the LLaMA-2-7B-chat model [20]. The objective is to train the model to produce concise, factually grounded highlights similar to author-written highlights based on instruction-following prompts.

Each training sample consists of an input prompt containing the abstract (trimmed using token budgeting), along with a set of constraints(top-ranked keywords and named entity phrases). The

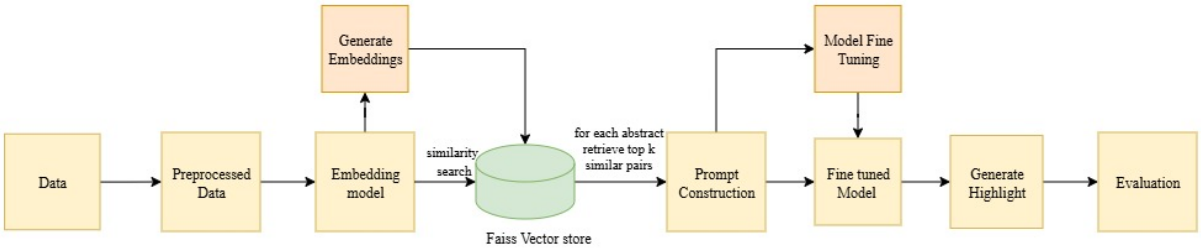

Figure 4: Overview of a Retrieval augmented generation based pipeline for prompt- driven highlight generation

target output (completion) is the human-written research highlight. We concatenate the prompt and completion into a single training instance as required by the LLaMA instruction format.

We apply QLoRA [21] to enable memory-efficient fine-tuning of the model. QLoRA reduces model size through 4-bit quantization and updates only a small set of parameters via LoRA adapters, allowing efficient training [22] without requiring high-end GPUs. The base model is loaded in 4-bit precision using the bitsandbytes library, while computations are performed in float16.

For training, we use a LoRA rank of 16, a LoRA alpha of 32, and a dropout rate of 0.1. The model is trained for one epoch with 1 batch per GPU and an accumulation step size of 8 for gradients. We optimize using the PagedAdamW optimizer with a learning rate of 2×10^{-4}, a cosine scheduler, 3% warm-up, and gradient clipping with a max norm of 0.3. The maximum sequence length is fixed at 512 tokens. Mixed-precision training is disabled due to hardware limitations on the Kaggle T4 GPU.

4. Prompt-Driven Retrieval-Augmented Generation Pipeline

As an alternative method, we implement a retrieval-augmented generation (RAG) approach [23] that makes use of similar abstract-highlight pairs stored in Faiss vector stores, one utilizing sentence-transformer embeddings for BART and T5, and another using Specter2 embeddings for SciBART to facilitate the generation process. This method integrates semantic retrieval of the top-k nearest examples with fine-tuned sequence-to-sequence models (BART, T5, and SciBART), creating prompts that incorporate the new abstract along with the retrieved examples to produce research highlights.

Prior to building the retrieval pipeline, we performed basic preprocessing of the dataset using the Stanza NLP package. This phase involves tokenization, splitting sentences, and normalization to achieve a uniform text representation throughout the abstracts and highlights.

4.1. Semantic Embedding and FAISS Indexing

To retrieve semantically similar samples, we use Sentence-BERT [24] and Specter2 to convert abstracts and highlights in the training set into dense vector embeddings. These embeddings are indexed using Facebook AI Similarity Search (FAISS) [25], an efficient search library for high-dimensional data, and separate FAISS indices are built for the Sentence-BERT and Specter2 embeddings. These indices are used during both training and inference to retrieve the top-k similar abstract-highlight pairs.

Let an abstract a_i be a sequence of tokens:

$$a_i = \{w_1, w_2, \ldots, w_n\}$$

The Sentence-BERT encoder $\text{SBERT}(\cdot)$ generates a dense embedding:

$$\mathbf{v}_i = \text{SBERT}(a_i) \in \mathbb{R}^d$$

where d is the embedding dimension.

Let $\mathcal{D} = \{(\mathbf{v}_1, h_1), (\mathbf{v}_2, h_2), \ldots, (\mathbf{v}_N, h_N)\}$ be the set of embedded abstract-highlight pairs. For a new query abstract embedding $\mathbf{v}_q$, we retrieve the top-k similar samples by maximizing cosine

similarity:

$$\text{TopK}(\mathbf{v}_q) = \arg\underset{j \in \mathcal{D}}{\text{top-}k} \left(\frac{\mathbf{v}_q \cdot \mathbf{v}_j}{\|\mathbf{v}_q\| \|\mathbf{v}_j\|} \right) \tag{6}$$

4.2. Retrieval-Augmented Prompt Construction

For each new abstract, whether in training or validation, we retrieve the top-k most similar examples from the FAISS index. These retrieved pairs are formatted into a template-style prompt inspired by in-context learning approaches [26]. The prompt presents three example abstract-summary pairs, followed by the new abstract to summarize. This strategy helps the model understand the summarization context by observing relevant examples before generating its output.

4.3. Model Fine-Tuning with Teacher Forcing

We fine-tune the BART model [27], which is a transformer-based sequence-to-sequence framework that has been pre-trained using a denoising autoencoding approach, for the purpose of generating highlights from scientific abstracts. Additionally, we utilize the same retrieval-augmented method for the domain-specialized SciBART and the general-purpose T5 model [28]. To provide contextual examples to the models, we create a prompt-based, retrieval-augmented training dataset; in this dataset, each abstract from the original training set is utilized as a new abstract, with its corresponding highlight acting as the target summary.

To enhance the input, we retrieve the top-k semantically similar abstract–summary pairs using FAISS indices built on Sentence-BERT embeddings for BART and T5, as well as Specter2 embeddings for SciBART, adjusting k to evaluate how the quantity of examples influences the quality of generation. From these, we exclude the current abstract if it is included in the retrieved set and employ the resulting pairs as contextual examples. The chosen examples are then organized into a structured input prompt using a consistent natural language template as follows:

```
Abstract1: ... Summary1: ... Abstract2: ... Summary2: ...
Abstract3: ... Summary3: ... New Abstract: <abstract>
```

This prompt serves as the model's input, while the original highlight becomes the target output. All input-target pairs are stored in a structured CSV file and used for supervised fine-tuning of the model. The training employs teacher forcing, wherein the ground truth summary tokens are supplied during decoding to stabilize learning and accelerate convergence. We utilize the sentence-transformers/all-MiniLM-L6-v2 model and tokenizer from the HuggingFace Transformers library for BART and T5, and the allenai/ specter2-base model with AutoTokenizer for SciBART. This strategy enables the model to leverage semantically relevant examples while learning to generate coherent and context-aware summaries tailored to scientific texts.

5. Results

In the Entity and Keyword guided Highlight Generation approach, to understand the impact of different input configurations on highlight generation quality, we evaluate six variants (V1–V6) of our model. These configurations vary by abstract length (complete vs. trimmed), inclusion of named entity phrases and keywords, and whether a filtering step is applied to retain only those entity phrases and keywords present in the reference highlights. Table 1 summarizes the ROUGE and METEOR scores for each variant.

Our baseline, V1, uses the complete abstract without any entity or keyword guidance. While this provides the model with full contextual information, it also introduces practical limitations. Since the maximum input length during training is capped at 512 tokens, the absence of trimming means that the output space reserved for generating highlights is sometimes pushed beyond the token limit, leading

Table 1
Evaluation of approach 1 models using Llama across different input configurations.

Model	Abstract	Entities	Keywords	Filtered	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
V1	Complete	No	No	No	30.17	07.28	18.35	25.33
V2	Trimmed	No	No	No	30.53	07.55	18.83	25.71
V3	Trimmed	Yes	No	No	30.33	07.42	18.59	25.60
V4	Trimmed	No	Yes	No	30.92	08.05	19.22	26.08
V5	Trimmed	Yes	Yes	No	31.53	08.65	19.83	26.61
V6	Trimmed	Yes	Yes	Yes	**32.03**	**09.25**	**20.33**	**27.31**

Table 2
Evaluation of BART and T5 models for retrieval-augmented generation with different values of k.

Model	Top-k	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
BART	3	22.21	05.21	15.74	13.82
SciBART	3	17.44	01.61	11.52	12.82
BART	5	16.65	01.47	11.61	09.48
BART	7	15.87	01.19	11.20	08.97
T5	3	08.83	00.19	07.03	04.30
T5	5	13.69	00.94	10.07	08.19

to partial or truncated highlight generations. This may impact learning, especially in cases where the target output is not fully visible. Despite this, V1 achieves a ROUGE-1 score of 30.17, ROUGE-2 score of 7.28, ROUGE-L score of 18.35 and a METEOR score of 25.33, making it a good starting point for comparison.

V2 uses a trimmed abstract but no entities, keywords, or filtering. It slightly outperforms V1 across all metrics, achievinging ROUGE-1 score of 30.53, ROUGE-2 score of 07.55, ROUGE-L score of 18.83 and METEOR score of 25.71, likely because trimming helps prioritize key content while ensuring enough space for highlight generation. However, with no additional guidance, the model relies entirely on the trimmed abstract, and the improvement can be attributed to better token management rather than semantic direction.

The V6 configuration demonstrates consistent improvement with ROUGE-1 increasing to 32.03, ROUGE-2 to 09.25, ROUGE-L to 20.33, and METEOR to 27.31. Overall, V6 emerges as the most performant and reliable configuration in terms of factual consistency and generation quality.

In the Prompt-driven Retrieval Augmented approach, we evaluate the effectiveness of the retrieval-augmented pipeline by comparing the performance of BART, SciBART, and T5 models under different retrieval configurations. Specifically, we examine how varying the number of retrieved examples ($k = 3$, $k = 5$, and $k = 7$) influences the quality of the generated highlights. Table 2 reports the ROUGE-1, ROUGE-2, ROUGE-L, and METEOR scores for all settings.

In this setup, all models are fine-tuned to generate highlights for scientific abstracts using the same retrieval-augmented prompting strategy. For each input abstract, we retrieve the top-k most similar abstract-highlight pairs using FAISS indices built on all-MiniLM-L6-v2 embeddings for BART and T5, and on specter2-base embeddings for SciBART, ensuring a fair comparison between models.

BART achieves the strongest overall performance, with its best results at top-3 retrievals (ROUGE-1: 22.21, ROUGE-2: 5.21, ROUGE-L: 15.74, METEOR: 13.82). SciBART, though competitive, lags behind BART under the same setting (ROUGE-1: 17.44, ROUGE-2: 1.61, ROUGE-L: 11.52, METEOR: 12.82). For BART, increasing k from 3 to 5 or 7 leads to reduced performance, due to longer, less focused prompts. In contrast, T5 shows improvement when moving from $k = 3$ (ROUGE-L: 07.03) to $k = 5$ (ROUGE-L:10.07), indicating that it gains from having more context, although its results are still considerably lower than those of BART.

6. Discussion

6.1. Qualitative Error Analysis

Qualitative analysis of model-generated highlights reveals systematic error patterns that provide deeper insights into the strengths and limitations of our approach. We analyzed a representative sample to identify recurring failure modes and understand their underlying causes.

Our analysis reveals numerical discrepancies in model-generated highlights. For example, while the source abstract reports "36.4% experienced stress," the corresponding generated highlight exhibits precision degradation, producing "36.5% experienced stress." According to Shah et al. [29], such numerical inconsistencies arise from fundamental limitations in how LLMs encode numerical information, particularly when numbers are tokenized and processed through attention mechanisms that may lose precise numerical relationships.

Beyond numerical errors, we observed over-abstraction in cases where model-generated highlights replace domain-specific terminology with more generic language, despite our entity and keyword-guided constraints. For instance, in a caregiver mental health study, the source abstract emphasizes specific intervention-related barriers: "negative perception of home care therapy is associated with higher strain" and "not using tele-rehabilitation and perception of it being a poor medium." However, the generated highlights abstract these to generalized pandemic-related barriers-"caregivers reported an increase in strain, particularly in the areas of social isolation, financial strain, and childcare responsibilities." While the reference highlights preserve intervention-specific context (home care therapy, tele-rehabilitation adoption), the generated highlights disconnect from the study's focus on rehabilitation service barriers. This over-abstraction reflects the model's tendency to prioritize high-frequency generic language patterns over domain-specific terminology during generation, even when such terms are successfully extracted through our entity and keyword guidance pipeline.

6.2. Performance Analysis

The performance comparison across six variants in the entity and keyword guided highlight generation approach demonstrates that simply providing more information or constraints does not guarantee better outcomes. Rather, the quality and relevance of guidance play a very important role.

Three critical design choices contribute to V6's superior performance: (1) Trimmed Abstracts - using sentence ranking and token budgeting, we select the most relevant 260–264 tokens from the abstract as input. This preserves the full 128-token space for highlight generation, ensuring that the completion is not cut off midway and stabilizing training. (2) Guided Constraints - V6 includes both named entity phrases and keywords, which help the model focus on core concepts and terminology relevant to the paper's contributions. (3) Reference-Aligned Filtering - The most significant gain comes from a filtering step that retains only those entity phrases and keywords explicitly present in the author-written highlights during training. This is in contrast to V5, which includes the same types of guidance but without filtering. V5 underperforms, suggesting that unfiltered guidance may introduce noise by highlighting irrelevant or overly specific content from the abstract.

The filtering mechanism in V6 plays a crucial role in reducing hallucinations, a well-known challenge in abstractive summarization. By grounding the model with only validated entity phrases and keywords, V6 avoids fabricating unsupported content and stays focused on what actually matters. Overall, the improvements observed in V6 show that effective highlight generation is not just about adding more guidance, but about adding the right, reference-aligned guidance.

The results from prompt-driven retrieval augmented approach indicate that retrieval-augmented generation can enhance highlight generation when combined with strong sequence-to-sequence models such as BART. Among the configurations evaluated in this methodology, BART with top-3 retrieval achieves the best overall performance, suggesting that a concise and relevant context benefits its generation abilities the most.

SciBART performs reasonably well under top-3 retrievals but still does not match BART's performance, suggesting that while domain-specific pretraining is useful, the model benefits less from added retrieval

context. In contrast, T5 shows better outcomes with top-5 retrievals compared to top-3, suggesting that it requires a larger number of examples to effectively utilize the retrieved context. Still, even with this improvement, T5 outputs remain less accurate and fluent than those of BART. For both BART and SciBART, increasing the number of retrieved examples beyond 3 tends to reduce performance, as excess context dilutes prompt relevance and coherence.

Overall, these findings highlight that both model selection and retrieval context size play a critical role in designing effective retrieval-augmented summarization pipelines. Optimizing these factors is essential to balance informativeness and focus, thereby producing high-quality, concise research highlights.

7. Conclusion

We present a structured approach for automatically generating research highlights from scientific abstracts using fine-tuned LLaMA. Our method integrates named entity recognition, keyword extraction, and sentence-level importance ranking to guide highlight generation under strict computational constraints.

Our evaluation across six input configurations demonstrates that optimal results are achieved when inputs are carefully curated through reference-aligned filtering. The V6 configuration, combining trimmed abstracts, guided constraints, and reference-aligned filtering, achieves the highest performance, confirming that effective highlight generation requires contextually relevant guidance rather than simply more constraints.

Additionally, we explore a retrieval-augmented generation approach that employs BART, SciBART, and T5 models with FAISS-indexed abstract-highlight pairs as contextual examples. Although BART with top-3 retrieval achieves the best performance in this configuration, and SciBART performs competitively under similar conditions, the constrained fine-tuning approach proves more effective overall for scientific highlight generation.

One key limitation in our current setup was the 512-token input constraint, which limited how much of the abstract and constraints could be passed to the model. With access to higher-end GPUs, future experiments can extend the input length to 1024 tokens or more, allowing for richer context and potentially more coherent outputs. Additionally, the entity extraction component can be improved by training our SciBERT model on a broader set of scientific NER datasets. This could enhance the quality and coverage of extracted entities, leading to even better guidance during generation.

Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly in order to: Grammar and spelling check, Plagiarism detection. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

Acknowledgments

The work presented in this report is carried out as part of our internship under the ANRF-SERB-CRG Project titled *"A Deep Explainable Framework for Semantically Similar Document Retrieval and Summarization of Legal Text"* (CRG/2023/007688) at the Department of Information Technology, National Institute of Technology Karnataka (NITK), Surathkal.

References

[1] M. A. Hanson, P. G. Barreiro, P. Crosetto, D. Brockington, The strain on scientific publishing, Quantitative Science Studies 5 (2024) 823–843. URL: https://doi.org/10.1162/qss_a_00327. doi:10.1162/qss_a_00327.

[2] 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. doi:10.1109/ACCESS.2023.3292300.

[3] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, LLaMA: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971, 2023. URL: https://arxiv.org/abs/2302.13971.

[4] 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. URL: https://aclanthology.org/W04-1013/.

[5] A. Lavie, A. Agarwal, METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments, in: Proceedings of the Second Workshop on Statistical Machine Translation, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 228–231. URL: https://aclanthology.org/W07-0734/.

[6] J. Kupiec, J. O. Pedersen, F. Chen, A trainable document summarizer, in: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, Seattle, Washington, USA, 1995, pp. 68–73. URL: https://doi.org/10.1145/215206.215333. doi:10.1145/215206.215333.

[7] R. Nallapati, B. Zhou, C. N. dos Santos, Ç. Gülçehre, 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. URL: https://aclanthology.org/K16-1028/. doi:10.18653/v1/K16-1028.

[8] 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 (ACL), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1073–1083. URL: https://aclanthology.org/P17-1099/. doi:10.18653/v1/P17-1099.

[9] I. Beltagy, K. Lo, A. Cohan, SciBERT: A pretrained language model for scientific text, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3613–3618. URL: https://aclanthology.org/D19-1371/. doi:10.18653/v1/D19-1371.

[10] T. Rehman, D. K. Sanyal, S. Chattopadhyay, P. K. Bhowmick, P. P. Das, Automatic generation of research highlights from scientific abstracts, in: Proceedings of the 2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE 2021), CEUR-WS.org, Online, 2021, pp. 69–70. URL: https://ceur-ws.org/Vol-3004/paper10.pdf.

[11] 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@COLING 2022), Association for Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 163–169. URL: https://aclanthology.org/2022.sdp-1.18.

[12] T. Rehman, S. Das, D. K. Sanyal, S. Chattopadhyay, An analysis of abstractive text summarization using pre-trained models, arXiv preprint arXiv:2303.12796 (2023). URL: https://doi.org/10.48550/arXiv.2303.12796. doi:10.48550/arXiv.2303.12796.

[13] T. Rehman, D. K. Sanyal, S. Chattopadhyay, Research highlight generation with ELMo contextual embeddings, Scalable Computing: Practice and Experience 24 (2023) 181–190. URL: https://doi.org/10.12694/scpe.v24i2.2238. doi:10.12694/scpe.v24i2.2238.

[14] T. Rehman, S. Chattopadhyay, D. K. Sanyal, Abstractive summarization of scientific documents:

Models and evaluation techniques, in: Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation (FIRE 2023), ACM, Panjim, India, 2023, pp. 121–124. URL: https://doi.org/10.1145/3632754.3632771. doi:10.1145/3632754.3632771.

[15] Q. Zhang, Z. Chen, H. Pan, C. Caragea, L. J. Latecki, E. C. Dragut, SciER: An entity and relation extraction dataset for datasets, methods, and tasks in scientific documents, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Miami, FL, USA, 2024, pp. 13083–13100. URL: https://aclanthology.org/2024.emnlp-main.726/. doi:10.18653/v1/2024.emnlp-main.726.

[16] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C.-H. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, Z. Lu, BioCreative V CDR task corpus: A resource for chemical disease relation extraction, Database 2016 (2016) baw068. URL: https://doi.org/10.1093/database/baw068. doi:10.1093/database/baw068.

[17] S. Zhao, T. Liu, S. Zhao, F. Wang, A neural multi-task learning framework to jointly model medical named entity recognition and normalization, in: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, AAAI Press, Honolulu, Hawaii, USA, 2019, pp. 817–824. doi:10.1609/aaai.v33i01.3301817.

[18] B. Issa, M. B. Jasser, H. N. Chua, M. Hamzah, A comparative study on embedding models for keyword extraction using KeyBERT method, in: Proceedings of the 13th IEEE International Conference on System Engineering and Technology (ICSET), IEEE, Shah Alam, Malaysia, 2023, pp. 40–45. doi:10.1109/ICSET59111.2023.10295108.

[19] A. Cohan, S. Feldman, I. Beltagy, D. Downey, D. S. Weld, SPECTER: Document-level representation learning using citation-informed transformers, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), Association for Computational Linguistics, Online, 2020, pp. 2270–2282. URL: https://aclanthology.org/2020.acl-main.207/. doi:10.18653/v1/2020.acl-main.207.

[20] V. Nivethitha, N. Verma, R. Bokadia, R. Jaju, Fine-tuning LLaMA-2-7B model for conversation summarization, in: Proceedings of the 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), IEEE, Kamand, India, 2024, pp. 1–7. doi:10.1109/ICCCNT61001.2024.10724909.

[21] Y. Xu, L. Xie, X. Gu, X. Chen, H. Chang, H. Zhang, Z. Chen, X. Zhang, Q. Tian, QA-LoRA: Quantization-aware low-rank adaptation of large language models, in: Proceedings of the Twelfth International Conference on Learning Representations (ICLR), OpenReview.net, Vienna, Austria, 2024. URL: https://openreview.net/forum?id=WvFoJccpo8.

[22] X. Liu, K. Ji, Y. Fu, Z. Du, Z. Yang, J. Tang, P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks, arXiv preprint arXiv:2110.07602 (2021).

[23] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. K"uttler, M. Lewis, W.-t. Yih, T. Rockt"aschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: Advances in Neural Information Processing Systems 33, Curran Associates Inc., Red Hook, NY, USA, 2020.

[24] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using siamese BERT-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3980–3990. doi:10.18653/v1/D19-1410.

[25] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazar'e, M. Lomeli, L. Hosseini, H. J'egou, The faiss library, arXiv preprint arXiv:2401.08281 (2024). doi:10.48550/arXiv.2401.08281.

[26] A. Voronov, L. Wolf, M. Ryabinin, Mind your format: Towards consistent evaluation of in-context learning improvements, Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024 (2024) 6287–6310. doi:10.18653/V1/2024.FINDINGS-ACL.375.

[27] 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: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. doi:10.18653/v1/2020.acl-main.703.

[28] 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 21 (2020) 1–67.

[29] R. S. Shah, V. Marupudi, R. Koenen, K. Bhardwaj, S. Varma, Numeric magnitude comparison effects in large language models, in: Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 6147–6161. URL: https://doi.org/10.18653/v1/2023.findings-acl.383. doi:10.18653/v1/2023.findings-acl.383.