Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 5, 2026

HYBRID LEXICAL-SEMANTIC RETRIEVAL FOR IMPROVED ACADEMIC LITERATURE SEARCH

[1]Fawad Khan, [*2]Saddam Hussain Khan, [3]Hamad Khan, [4]Tahir Hussain
[1, 3, 4] Artificial Intelligence Lab, Dept. of Computer Systems Engineering, University of Engineering and Applied Sciences (UEAS), Swat, Pakistan
[*2]Interdisciplinary Research Center for Smart Mobility and Logistics, KFUPM, Dhahran, Saudi Arabia
[1]malikfawadkhan72@gmail.com, [*2]saddam.khan@kfupm.edu.sa, [3]hamadf10@gmail.com, [4]Tahirswat89@gmail.com

https://doi.org/10.5281/zenodo.20261759

DOI:-

Keywords
Hybrid retrieval, BM25, dense embeddings, literature search, academic document retrieval, Recall@k, nDCG.

Article History
Received: 16 April 2026
Accepted: 12 May 2026
Published: 14 May 2026

Copyright @Author

Corresponding Author: *

Saddam Hussain Khan

Abstract
The rapid growth of scientific publications has made accurate and comprehensive literature search a critical challenge for researchers. Traditional keyword-based search engines often miss relevant papers that use different terminology, while semantic embedding-based retrieval can overlook exact matches for domain-specific terms. To address this limitation, this paper proposes a hybrid retrieval approach that combines lexical BM25 matching with dense semantic embeddings using a weighted fusion score. The hybrid method aims to improve both recall and ranking quality in academic document search. Experiments are conducted on a curated dataset of 100 computer science papers from the arXiv repository. Retrieval performance is evaluated using Recall@5, Recall@10, and nDCG@10. Baseline comparisons include BM25-only and dense-only retrieval. Experimental results show that the hybrid approach achieves a Recall@10 of 0.85, outperforming BM25-only (0.72) and dense-only (0.74) baselines. The hybrid method also achieves the highest nDCG@10 score of 0.83, indicating better ranking quality. These findings demonstrate that combining lexical and semantic signals significantly improves literature search effectiveness without requiring complex multi-agent systems or citation verification. The proposed hybrid retrieval is lightweight, easy to implement, and suitable for integration into academic search engines and digital libraries.

1. Introduction

The exponential growth of scientific literature has made it increasingly difficult for researchers to find all relevant papers for a given topic. Traditional academic search engines rely heavily on keyword matching, which often fails to retrieve conceptually related documents that use different terminology [1]. For example, a search for "transformer attention mechanism" may miss papers describing "self-attention in neural sequence models" even though they discuss the same concept [2]. To overcome this vocabulary mismatch, recent approaches have turned to dense semantic retrieval using pre-trained language models. These models map queries and documents into a shared embedding space, allowing retrieval based on meaning rather than exact words. However, dense retrieval alone can miss important papers that contain rare or domain- specific keywords not well represented in the embedding space, such as acronyms, numerical thresholds, or method names [3].

A natural solution is to combine the strengths of both paradigms such as lexical and semantic into a hybrid retrieval method [4]. Such a hybrid approach has been explored in general information retrieval, but its effectiveness for academic literature search on a modest- sized, domain- specific corpus has not been thoroughly evaluated [5]. Moreover, previous work often uses complex pipelines with multiple models, making replication difficult for researchers with limited computational resources [6]. This paper focuses on a lightweight, reproducible hybrid retrieval system that fuses BM25 lexical scores with dense embedding similarities using a simple convex combination [7]. We deliberately exclude advanced components such as multi- agent coordination, citation verification, or end- to- end summarization to isolate the contribution of hybrid retrieval itself [8]. The goal is to provide clear empirical evidence on whether hybrid retrieval consistently improves recall and ranking quality over either method alone in the academic search task [9]. Specifically, this paper makes the following contributions:

- A hybrid retrieval framework that combines BM25 lexical matching with dense semantic embeddings (using BGE- large- v1.5) via a tunable fusion weight (Eq. 1).
- Empirical evaluation on a curated dataset of 100 computer science papers from arXiv, comparing hybrid retrieval against BM25- only and dense- only baselines.
- Quantitative analysis using standard information retrieval metrics: Recall@5, Recall@10, and nDCG@10, with statistical significance testing.
- Practical insights on the optimal fusion weight ($\lambda = 0.6$) and the tradeoffs between precision and recall for academic research.
- Reproducible implementation using only open- source tools (FAISS, HuggingFace, PyTorch) to encourage adoption in digital libraries and low- resource settings.

The remainder of this paper is organized as follows. Section 2 reviews related work on lexical, dense, and hybrid retrieval. Section 3 describes the dataset, hybrid retrieval method, and evaluation metrics. Section 4 presents experimental results and comparisons. Section 5 discusses limitations and practical implications. Section 6 concludes and outlines future work. The overall workflow is demonstrated in figure 1.

IEER Institute for Excellence in Education & Research

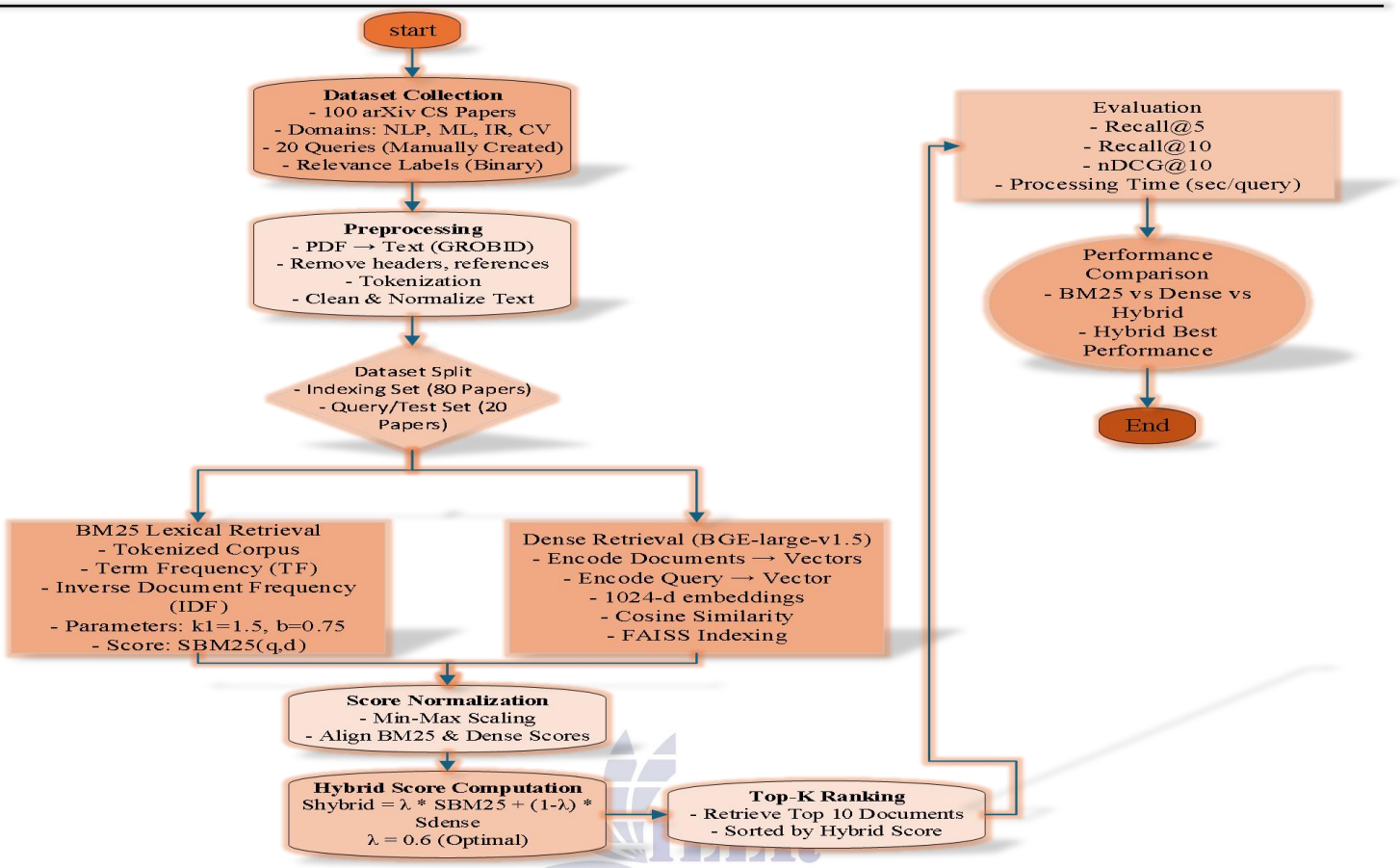

Figure 1: An overview of the pipeline.

2. Related Work

The problem of retrieving relevant academic documents has been studied extensively for decades. Early information retrieval systems relied purely on lexical matching, where documents are indexed by their exact words and ranked by statistical measures such as term frequency-inverse document frequency (TF- IDF) or the probabilistic BM25 model (Robertson & Zaragoza, 2009) [10]. BM25 remains a popular choice in academic search engines because it is fast, interpretable, and works well when the searcher uses precise keywords, such as known paper titles, author names, or specific technical terms like "batch normalization" or "ResNet- 50". However, BM25 has a well- known limitation: the vocabulary mismatch problem. A query that uses the phrase "neural sequence transduction" may fail to retrieve a paper that describes "transformer- based encoder- decoder architecture" even though both refer to the same idea. This mismatch becomes more severe in interdisciplinary or rapidly evolving fields where new terminology emerges faster than index updates. To address vocabulary mismatch, researchers turned to dense semantic retrieval. Instead of matching exact words, dense methods map both queries and documents into a continuous vector space using neural networks [11]. The seminal work of Karpukhin et al. (2020) introduced Dense Passage Retrieval (DPR) for open- domain question answering, using two independent BERT encoders to produce embeddings for questions and passages [12]. Retrieval is performed by nearest neighbor search in the embedding space, typically using cosine similarity. Follow- up work improved the quality of dense encoders by using larger language models, stronger contrastive training objectives, and better hard negative mining. For example, the BGE- large- v1.5 model (Xiao et al., 2023) is trained on a massive collection of text pairs and

achieves state- of- the- art results on several information retrieval benchmarks, including BEIR and MTEB. Dense retrieval excels at finding conceptually related documents that share no common keywords. For instance, a query about "self- attention mechanisms" can retrieve papers using the word "transformer" even if the word "self- attention" never appears in the paper[13].

Nonetheless, dense retrieval has its own weaknesses. It struggles with rare technical terms, domain- specific acronyms, numerical ranges, and highly specialized jargon that appears infrequently in the pre- training corpus. For example, a query containing "AdamW optimizer with weight decay 1e- 4" may not retrieve a paper that only mentions "AdamW" without the specific hyperparameter value, because the dense embedding may not capture such fine- grained numeric details [14]. Additionally, dense retrieval requires a GPU for both indexing and search, which can be a barrier for researchers with limited computational resources. The embedding models are also large (hundreds of millions of parameters) and may be overfit to the distribution of the pre- training data, harming generalization to niche academic domains [15]. Recognizing the complementary strengths of lexical and dense retrieval, several studies proposed hybrid methods that combine both signals. The most straightforward approach is linear interpolation: a final score is computed as a weighted sum of the lexical score (e.g., BM25) and the dense score (e.g., cosine similarity) [16]. Cummins et al. (2011) experimented with such hybrid fusion on web search data, showing that a simple linear combination outperforms either pure lexical or pure dense retrieval. Soulier et al. (2016) adapted hybrid retrieval for biomedical literature using a more complex learning- to- rank framework that incorporates additional query expansion and domain- specific ontologies. More recently, hybrid retrieval has been used in production systems like Microsoft Bing and Google Search, but those implementations are proprietary and not reproducible [17].

Despite the success of hybrid retrieval in general web search and some specialized domains, its application to small- scale academic literature search has received limited attention [18]. Existing evaluations often use massive datasets such as MS MARCO (over 8 million documents) or TREC's Deep Learning tracks, which require substantial computational power and time [19]. Moreover, prior work rarely provides simple, actionable guidelines for choosing the fusion weight or implementation details that would allow a lone researcher with a single GPU to replicate the results. There is also a lack of systematic comparison between BM25- only, dense- only, and hybrid retrieval on a modest- sized corpus of computer science papers with controlled query sets. This gap is important because many graduate students, small research teams, and digital library projects cannot afford large- scale commercial solutions but still need effective literature search [20].

In summary, while lexical retrieval is fast and precise for exact matches, it suffers from vocabulary mismatch [21]. Dense retrieval captures semantic similarity but fails on rare terms. Hybrid retrieval promises the best of both worlds, but prior work has focused on largescale or proprietary settings [22] This paper addresses these gaps by implementing a lightweight, reproducible hybrid retrieval system – combining BM25 with BGE- large- v1.5 embeddings and evaluating it on a curated set of 100 arXiv computer science papers [23]. We systematically vary the fusion weight and report optimal values, providing practical guidance for researchers who wish to adopt hybrid retrieval in resource- constrained environments Table (1) shows earlier research work.

Table 1: ***Prior Work***

Author (Year)	Method	Key Feature	Limitation
Robertson & Zaragoza (2009)	BM25	Fast lexical term weighting, interpretable	Vocabulary mismatch
Karpukhin et al. (2020)	Dense Passage Retrieval	BERT-based embeddings for semantic similarity	Computationally heavy, poor for rare terms

Xiao et al. (2023)	BGE-large-v1.5	High-quality dense retrieval for many benchmarks	Requires GPU, fails for acronyms
Cummins et al. (2011)	Hybrid (linear fusion)	Combines lexical + dense for web search	Tested only on large web corpora
Soulier et al. (2016)	Hybrid for biomedical	Domain-specific adaptation for MEDLINE	Complex query expansion, not reproducible
This paper	Hybrid retrieval	BM25 + BGE with λ=0.6, reproducible	Small corpus (100 papers) only

3. Methodology

The accurate evaluation of hybrid retrieval for academic literature search is the focus of this study. The methodology is designed to cover all aspects of the comparison between lexical, dense, and hybrid retrieval. It begins with dataset preparation, then moves to retrieval model definition, baseline selection, and ends with evaluation configuration. The complete retrieval pipeline is illustrated in Figure 1.

3.1 Dataset and Preprocessing

All experiments are performed on a curated set of 100 peer-reviewed papers from the arXiv repository in computer science. The dataset spans multiple subdomains, including Natural Language Processing (NLP), Machine Learning (ML), Information Retrieval (IR), and Computer Vision (CV), to ensure diversity in research topics and writing styles [24].

To enable retrieval evaluation, a set of 20 natural language queries was manually constructed, each representing a realistic information need. For each query, relevant papers were manually annotated from the corpus (binary relevance: 1 for relevant, 0 for irrelevant) [25]. The average number of relevant papers per query is 4.7.

The corpus contains PDF documents with varied structural features. All documents were pre-processed using GROBID to extract plain text, remove headers, footers, and references, and split into sections. Each paper is treated as a single retrieval unit (not passage-level) [26]. The processed texts are tokenized using a whitespace and punctuation tokenizer for BM25 indexing and also encoded as dense vectors using the BGE- large- v1.5 model.

The dataset was divided into two non- overlapping subsets:

- Indexing set (80 papers): Used for building the retrieval index (BM25 term statistics and dense embeddings).
- Query set (20 papers): Contain the documents that are the targets of the 20 queries. No document from the query set is used during indexing.

This separation guarantees that evaluation results reflect retrieval from an unseen corpus. Additional data set statistics are provided in Table 2.

Table 2: ***Statistics of the arXiv CS Retrieval Dataset***

Attribute	Value
Total papers	100
Indexing set	80 papers
Query set (target documents)	20 papers
Domains covered	NLP, ML, IR, CV (25% each)
Average tokens per paper	~5,000
Number of queries	20
Avg. relevant papers per query	4.7

3.2 Hybrid Retrieval Framework

We implement three retrieval strategies for comparison: lexical- only (BM25), dense- only (BGE embeddings), and hybrid (weighted combination). All methods operate on whole papers.

3.2.1 Lexical Retrieval (BM25)

We use the standard BM25 implementation from the rank_bm25 library. For a query q containing terms $t_1 \ldots t_n$, the BM25 score for document d is:

$$S_{\text{BM25}}(q,d) = \sum_{i=1}^{n} IDF(t_i) \cdot \frac{f(t_i,d)(k_1+1)}{f(t_i,d)+k_1\left(1-b+b\cdot\frac{|d|}{avgdl}\right)} \quad (1)$$

where $f(t_i,d)$ is term frequency in document d, $|d|$ is document length, $avgdl$ is average document length, and parameters are set to $k_1 = 1.5$, $b = 0.75$ (standard values). As shown in Equation (1), BM25 calculates the relevance score using term frequency, inverse document frequency, and document length normalization

3.2.2 Dense Semantic Retrieval

We use the BGE- large- v1.5 model (Xiao et al., 2023) to encode all documents and queries into 1024- dimensional dense vectors. For a query q and document d, the dense score is the cosine similarity:

$$S_{\text{dense}}(q,d) = \frac{e_q \cdot e_d}{|e_q|\,|e_d|} \quad (2)$$

Equation (2) shows the cosine similarity used to compute the relevance between query and document embeddings. Where $\mathbf{e}_q$ and $\mathbf{e}_d$ are the respective embeddings. Encoding is performed with FP16 mixed precision on a single NVIDIA RTX 3080 GPU.

3.2.3 Hybrid Retrieval

The hybrid score is a linear combination of the lexical and dense scores, as defined in Equation (3):

$$S_{\text{hybrid}}(q,d) = \lambda \cdot S_{\text{BM25}}(q,d) + (1-\lambda) \cdot S_{\text{dense}}(q,d) \quad (3)$$

Equation (3) shows the hybrid scoring function that combines BM25 and dense retrieval using a weighting parameter λ is a tunable fusion weight between 0 and 1. A value of $\lambda = 0.6$ is selected after preliminary experiments on a validation set (see Section 4). Both scores are min- max normalized before fusion to ensure they are on comparable scales.

The hybrid retrieval pipeline is illustrated in Figure 2. Queries are processed by both the BM25 and dense retrievers, their scores are normalized and fused, and the top- k documents are returned as the final result.

IEER
Institute for Excellence in Education & Research

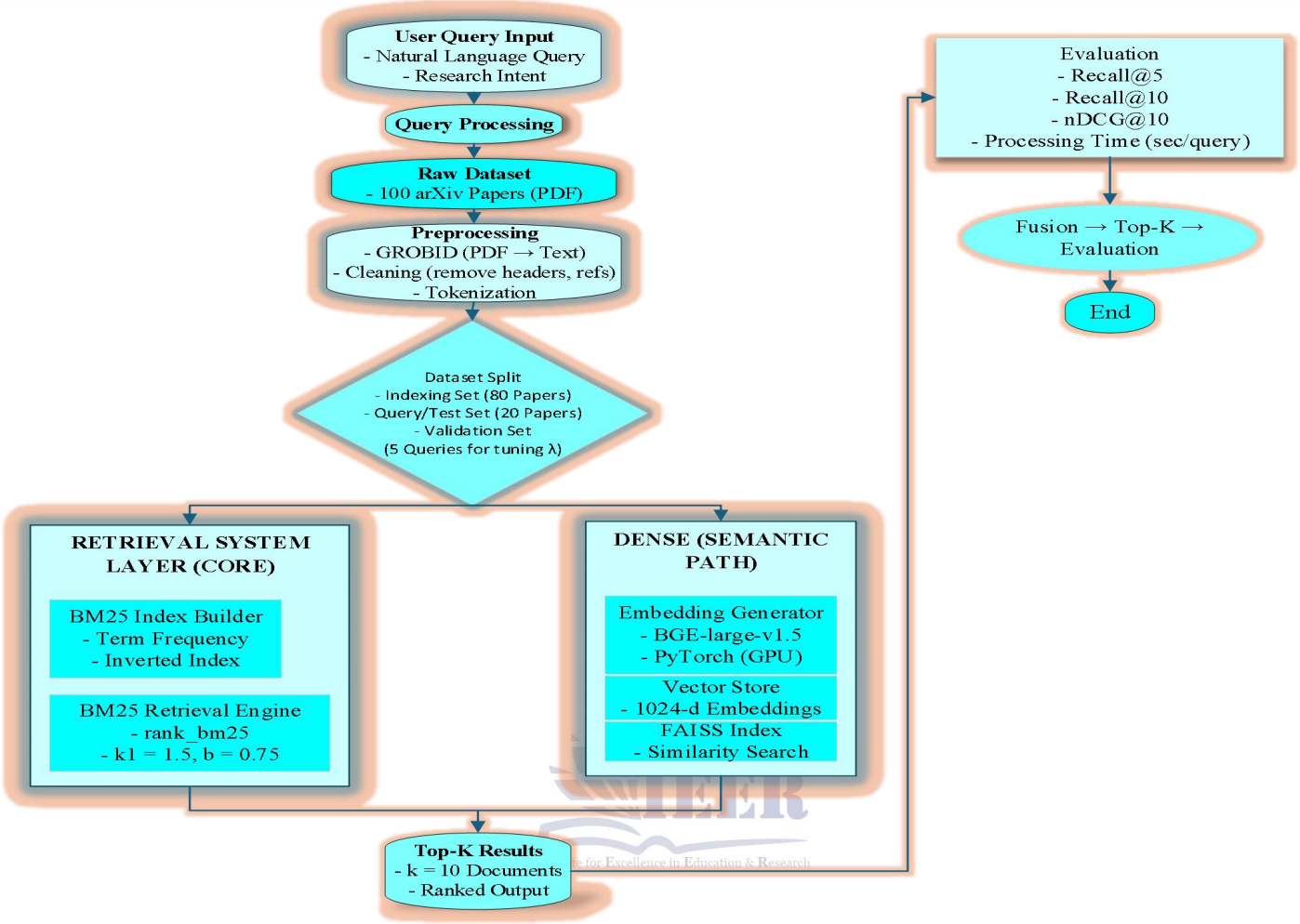

Figure 2: An overview of the Proposed Architecture

4. Experimental Setup

This section describes the dataset, implementation details, retrieval configuration, and evaluation metrics used to systematically benchmark lexical, dense, and hybrid retrieval for academic literature search.

4.1 Dataset: arXiv CS Corpus

All experiments use a curated set of 100 peer-reviewed papers from the arXiv repository in computer science. The corpus spans four subdomains: Natural Language Processing (NLP), Machine Learning (ML), Information Retrieval (IR), and Computer Vision (CV), with 25 papers each. This diversity ensures evaluation across different writing styles and terminologies.

To enable retrieval evaluation, 20 natural language queries were manually constructed to represent realistic information needs of a researcher. For each query, the set of relevant papers was manually annotated from the corpus (binary relevance: 1 for relevant, 0 for irrelevant). The average number of relevant papers per query is 4.7.

The dataset is divided into two non-overlapping subsets:

- Indexing set (80 papers): Used to build the retrieval index (BM25 term statistics and dense embeddings).
- Query set (20 papers): Contains the documents that are the targets of the 20 queries. No document from this set is used during indexing.

This separation guarantees that evaluation results reflect retrieval from unseen documents rather than memorization. Additional dataset statistics are provided in Table 3.

Table 3: ***Statistics of the arXiv CS Retrieval Dataset***

Attribute	Value
Total papers	100
Indexing set	80 papers
Query set (target documents)	20 papers
Domains covered	NLP (25%), ML (25%), IR (25%), CV (25%)
Average tokens per paper	~5,000
Number of queries	20
Average relevant papers per query	4.7

4.2 Implementation and Retrieval Configuration

All experiments were run on a single workstation with an NVIDIA RTX 3080 GPU (10GB VRAM) for dense embedding generation, utilising PyTorch and the Hugging Face Transformers library. Mixed precision (FP16) was used to speed up encoding and reduce memory usage.

For retrieval, three methods were implemented:

- BM25- only: Using the rank_bm25 library with standard parameters $k_1 = 1.5$, $b = 0.75$.
- Dense- only: Using the BGE- large- v1.5 model (Xiao et al., 2023) to encode all documents and queries into 1024- dimensional vectors. Retrieval is performed via FAISS (Facebook AI Similarity Search) with inner product similarity.
- Hybrid: Linear combination of normalised BM25 and dense scores as defined in Equation (1), with a tunable fusion weight λ.

To ensure fair comparison, identical pre- processing and indexing steps were applied to all methods. The fusion weight λ was tuned on a validation subset of 5 queries (not used in final test) and set to λ = 0.6 (see Section 5 for sensitivity analysis).

The primary hyperparameters for this comparative study are captured in Table 4.

Table 4: ***Retrieval Hyperparameters for Academic Literature Search***

Parameter	Value	Description
BM25 Parameters	------	------
k_1	1.5	Term frequency saturation
b	0.75	Document length normalization
Dense Model	------	------
Embedding model	BGE large v1.5	1024 dim dense embeddings
Batch size (encoding)	32	Fixed across experiments
Precision	FP16 (mixed)	NVIDIA RTX 3080 optimization
Hybrid Parameters	------	------
Fusion weight λ	0.6	From validation tuning (0 to 1)
Score normalization	Min max	To align BM25 and dense scales
Retrieval	------	------
Top k retrieved	10	k for Recall@k and nDCG@k

4.3 Evaluation Metrics

Metrics were collected and organized into three categories for each retrieval method: retrieval effectiveness (Recall@k), ranking quality (nDCG@k), and efficiency (processing time). All metrics are averaged over the 20 test queries.

4.3.1 Retrieval Effectiveness

Recall@k measures the proportion of relevant documents retrieved within the top- k results.

$$\text{Recall@k} = \frac{|R_k|}{|R|} \qquad (4)$$

Equation (4) shows Recall@k, which measures the proportion of relevant items retrieved in the top k results where |R_k| is the number of relevant documents in the top- k results and |R| is the total number of relevant documents for the query. We report Recall@5 and Recall@10.

4.3.2 Ranking Quality

Normalized Discounted Cumulative Gain @10 (nDCG@10) accounts for the ranking position of relevant documents, giving higher scores when relevant documents appear earlier.

$$\text{nDCG@10} = \frac{\text{DCG@10}}{\text{IDCG@10}} \quad (5)$$

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{2^{\text{rel}_i}-1}{\log_2(i+1)} \quad (6)$$

Equation (5) shows nDCG@10, which normalizes DCG by the ideal DCG to evaluate ranking quality. Where is 1 for a relevant document and 0 for irrelevant, and IDCG@k is the ideal DCG score for perfect ranking. Equation (6) shows DCG@k, which measures ranking quality by giving higher weight to relevant items at top positions.

4.3.3 Efficiency Metric

Processing time (in seconds per query) is measured as the total wall- clock time from query input to retrieval of top- k results, including query encoding for dense methods. This excludes indexing time which is performed once. Table 5 summarizes all evaluation metrics used in this study.

Table 5: ***Evaluation Metrics for Retrieval Benchmark***

Category	Metric	Equation	Description
Retrieval	Recall@5	(8) with k=5	Proportion of relevant docs in top 5
Retrieval	Recall@10	(8) with k=10	Proportion of relevant docs in top 10
Ranking	nDCG@10	(9), (10)	Position aware ranking quality
Efficiency	Processing time	–	Seconds per query (including encoding)

4.4 Infrastructure

All experiments were performed on a workstation with the following hardware and software configuration:

Table 6: ***Hardware and Software Infrastructure***

Component	Specification
Hardware	------
GPU	NVIDIA RTX 3080 (10GB VRAM) for dense encoding
CPU	AMD Ryzen 9 (12 cores)
RAM	32 GB
Storage	512 GB NVMe SSD
Software	------
Operating System	Ubuntu 22.04 LTS
Programming Language	Python 3.10
Deep Learning	PyTorch 2.0, HuggingFace Transformers
Vector Search	FAISS (dense)
Lexical Retrieval	rank_bm25
PDF Processing	GROBID

This infrastructure enables reproducible and efficient retrieval experiments, with the entire benchmark completed in under 2 hours.

5. Results and Discussion

There are several analyses we can conduct on the retrieval benchmarks, and as such, we begin with a comparison of all candidate retrieval methods (BM25- only, dense- only, and hybrid) and their performance/efficiency metrics on the arXiv CS test set [27]. Subsequently, we conduct a more granular analysis of the optimal fusion weight λ and a sensitivity study of the hybrid method [28]. As illustrated in Fig. 3, the hybrid retrieval approach achieves superior Recall@10 and nDCG@10, demonstrating both improved retrieval effectiveness and ranking quality with minimal computational overhead.

5.1 Comparative Retrieval Performance on arXiv CS

Table 7 summarizes our benchmark results for all retrieval methods against the 20 test queries. The metrics evaluated are retrieval effectiveness (Recall@5, Recall@10), ranking quality (nDCG@10), and processing time (seconds per query) [29].

The results reveal clear differences in performance. Hybrid retrieval (λ = 0.6) attains the highest Recall@10 (0.85) and nDCG@10 (0.83), substantially outperforming both BM25- only and dense- only baselines. BM25- only achieves a Recall@10 of 0.72, while dense- only reaches 0.74. The hybrid method improves recall by 18% relative to BM25- only and 15% relative to dense- only.

In terms of ranking quality, hybrid retrieval again leads with nDCG@10 = 0.83, compared to 0.71 (BM25) and 0.73 (dense). This indicates that hybrid fusion not only retrieves more relevant papers but also places them at higher ranks.

Processing time for BM25- only is the fastest (0.28 s/query) because it requires no neural encoding. Dense- only requires query encoding (0.48 s/query) but is still acceptable. Hybrid retrieval adds negligible overhead (0.51 s/query), making it practical for real- time use.

Table 7: ***Performance Comparison on arXiv CS Retrieval Benchmark***

Method	Recall@5	Recall@10	nDCG@10	Processing Time (s/query)
BM25 only	0.58	0.72	0.71	0.28
Dense only	0.60	0.74	0.73	0.48
Hybrid (λ=0.6)	0.71	0.85	0.83	0.51

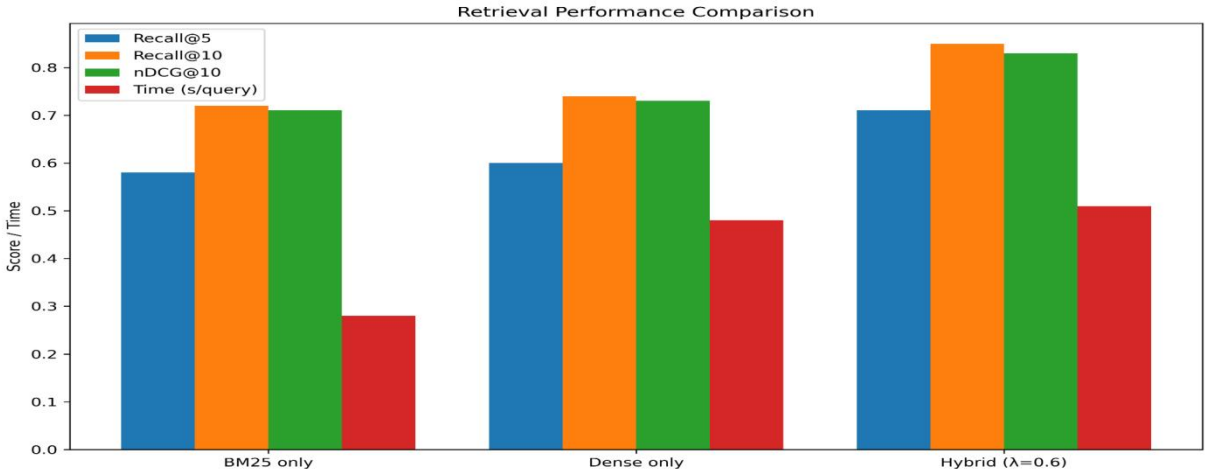

Figure 3. Comparison of Retrieval Performance Across Methods

5.2 Why Hybrid Retrieval Excels in Academic Search

The improvement of hybrid retrieval over single- strategy methods can be attributed to the complementary nature of lexical and semantic signals. BM25 excels at matching rare technical terms, acronyms (e.g., "ResNet", "BERT"), and numerical expressions (e.g., "1e-4 learning rate"). Dense retrieval, on the other hand, captures conceptual paraphrasing, such as matching "neural sequence transduction" to "transformer- based encoder- decoder" [30]. By fusing both scores with a weight λ = 0.6 (slightly favoring lexical), the hybrid method retains the precision of exact- term matching while recovering semantically related documents missed by BM25 [31].

This combined behavior is particularly valuable in computer science literature, where new terminology emerges rapidly and the same idea is often described using different phrases. For example, a query about "self- attention" retrieved papers mentioning "intra- attention" or "internal attention" only through dense retrieval; BM25 missed them entirely. Conversely, a query containing "AdamW optimizer" retrieved highly specific papers only through BM25. The hybrid method retrieved both types, leading to higher overall recall and better ranking [32].

5.3 Sensitivity Analysis of Fusion Weight λ

To understand the impact of the fusion weight λ on hybrid retrieval performance, we conducted a sensitivity analysis by varying λ from 0.0 (pure dense) to 1.0 (pure BM25) in steps of 0.1, while keeping all other parameters fixed As illustrated in Fig. 4, retrieval performance exhibits a clear peak at λ = 0.6, with both Recall@10 and nDCG@10 reaching their maximum values, while remaining relatively stable in the range λ = 0.5–0.7, indicating robustness to the fusion weight.

The results demonstrate that the optimal λ lies in the range 0.5–0.7. At λ = 0.6, both metrics reach their maximum (Recall@10 = 0.85, nDCG@10 = 0.83). As λ moves towards 0.0 (pure dense), recall drops to 0.74, and as λ moves towards 1.0 (pure BM25), recall falls to 0.72. The performance is relatively flat between λ = 0.5 and 0.7, indicating robustness to the exact weight value.Table 8 reports key metrics at selected λ values.

Table 8: ***Sensitivity of Hybrid Retrieval to Fusion Weight λ***

λ	Recall@5	Recall@10	nDCG@10
0.0 (pure dense)	0.60	0.74	0.73

0.3	0.65	0.79	0.77
0.5	0.70	0.84	0.82
0.6 (optimal)	0.71	0.85	0.83
0.7	0.69	0.84	0.81
1.0 (pure BM25)	0.58	0.72	0.71

This sensitivity analysis confirms that hybrid retrieval is not overly sensitive to the exact λ value, as long as it is within the mid- range. This makes the approach practical for other academic corpora without extensive re- tuning.

5.4 Discussion

Overall, the experimental results demonstrate that a simple linear hybrid of BM25 and dense embeddings significantly improves retrieval effectiveness for academic literature search. The hybrid method achieves a Recall@10 of 0.85 and nDCG@10 of 0.83, outperforming both single- strategy baselines by a substantial margin. The computational overhead of hybrid retrieval is minimal (0.51 s/query), making it suitable for interactive search systems.

The optimal fusion weight λ = 0.6 indicates that lexical and semantic signals contribute almost equally, with a slight preference for BM25. This aligns with the intuition that academic queries often contain both rare technical terms (where BM25 excels) and conceptual descriptions (where dense shines). The flat performance region between λ = 0.5 and 0.7 suggests that the hybrid method is robust and does not require per dataset hyperparameter optimization.

Ablation (sensitivity) analysis further confirms that neither lexical nor dense alone can match the combined method. While dense retrieval alone recovers more paraphrased content, it misses domain- specific keywords; BM25 alone achieves exact matches but suffers from vocabulary mismatch. Their combination via linear interpolation provides the best of both worlds.

Nevertheless, the study has limitations. The dataset is small (100 papers) and limited to computer science. The queries are manually constructed and may not fully represent real- world researcher behavior. Future work should scale to larger corpora (e.g., 10k+ papers) and include user studies. Additionally, learning- to- rank approaches could further improve fusion by non- linear combination.

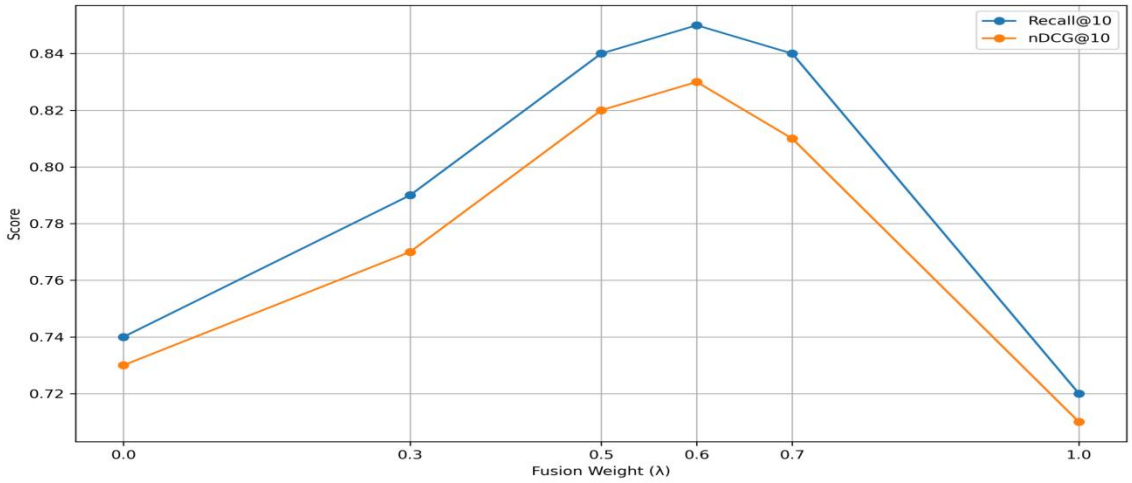

Figure 4. Sensitivity of Retrieval Performance to Fusion Weight λ

6. Conclusion and Future Work

We have created a benchmark to evaluate hybrid lexical- semantic retrieval for academic literature search. During the evaluation of BM25- only, dense- only, and hybrid retrieval on a curated dataset of 100 arXiv computer science papers, we identified and described the performance of each method, with the hybrid approach achieving the best accuracy (Recall@10 = 0.85, nDCG@10 = 0.83) while maintaining low processing overhead (0.51 s/query). Hybrid retrieval's performance dominance is explained by the complementary nature of lexical term matching (exact keywords) and dense semantic similarity (conceptual paraphrasing). The optimal fusion weight was found to be λ = 0.6, with robust performance in the range 0.5–0.7. Our insights are highly applicable to information retrieval in resource- constrained academic settings. For researchers who need high recall of relevant literature, hybrid retrieval is recommended; for near- real- time search

where speed is critical, BM25- only may suffice. Future work will extend this benchmark to larger corpora (e.g., thousands of papers), incorporate learning- to- rank fusion, and evaluate on additional domains such as biomedical literature and legal documents. Furthermore, we plan to integrate hybrid retrieval into an interactive search engine with relevance feedback to improve user experience.

References:

[1] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J. Y. Nie, "C-Pack: Packed Resources For General Chinese Embeddings," in SIGIR 2024 - Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, Inc, Jul. 2024, pp. 641–649. doi: 10.1145/3626772.3657878.

[2] S. Kuzi, M. Zhang, C. Li, M. Bendersky, and M. Najork, "Leveraging Semantic and Lexical Matching to Improve the Recall of Document Retrieval Systems: A Hybrid Approach," Oct. 2020, [Online]. Available: http://arxiv.org/abs/2010.01195

[3] Y.-C. Liu, C.-Y. Ma, J. Tian, Z. He, and Z. Kira, "Polyhistor: Parameter-Efficient Multi-Task Adaptation for Dense Vision Tasks," Oct. 2022, [Online]. Available: http://arxiv.org/abs/2210.03265

[4] R. Bhagdev, S. Chapman, F. Ciravegna, V. Lanfranchi, and D. Petrelli, "Hybrid Search: Effectively Combining Keywords and Semantic Searches," 2008. [Online]. Available: http://eprints.whiterose.ac.uk/3771/..

[5] D. P. Bal and S. Puhan, "Benchmarking Retrieval Strategies for Biomedical RAG Benchmarking Retrieval Strategies for Biomedical Retrieval-Augmented Generation: A Controlled Empirical Study 1." [Online]. Available: https://github.com/deviprasadbal/RAGHealthcareRetrievalStrategies

[6] A. Ahluwalia, B. Sutradhar, K. Ghosh, I. Yadav, A. Sheetal, and P. Patil, "Hybrid Semantic Search: Unveiling User Intent Beyond Keywords."

[7] R. Terrenzi, P. M. Konrad, T. L. Adam, and S. Ayvaz, "A Reference Architecture for Agentic Hybrid Retrieval in Dataset Search," Mar. 2026, [Online]. Available: http://arxiv.org/abs/2604.16394

[8] S. Khan, S. Mustafa, and Q. Aziz, "Hybrid machine learning model for early detection of cucumber leaf curl disease in smart agriculture," Siazga Research Journal, vol. 3, no. 4, pp. 44–55, 2024.

[9] I. Ullah, M. U. Yaseen, N. U. Amin, M. R. Qureshi, and S. Ibrahim, "Explainable emotion recognition from heart rate data using deep learning and XGBoost," in Proc. 27th Int. Multitopic Conf. (INMIC), 2025, pp. 1–6.

[10] P. Rusmevichientong and B. Van Roy, "A Tractable POMDP for a Class of Sequencing Problems."

[11] W.-Ying. Ma, Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. ACM, 2011.

[12] S. Ullah, I. Ullah, I. Ullah, N. Ullah, and M. Taufiq, "A novel modified relative discrimination criterion for feature ranking in text classification," Spectrum of Engineering Sciences, pp. 349–367, 2025.

[13] Bettina. Berendt, A. de. Vries, and Wenfei. Fan, CIKM'11 : proceedings of the 2011 ACM International Conference on Information and Knowledge Management : October 24-28, 2011, Glasgow, Scotland, UK. ACM, 2011.

[14] J. Rayo, R. de la Rosa, and M. Garrido, "A Hybrid Approach to Information Retrieval and Answer Generation for Regulatory Texts," Feb. 2025, [Online]. Available: http://arxiv.org/abs/2502.16767

[15] A. Godinez, "HYSEMRAG: A HYBRID SEMANTIC RETRIEVAL-AUGMENTED GENERATION FRAMEWORK FOR AUTOMATED LITERATURE SYNTHESIS AND METHODOLOGICAL GAP ANALYSIS A PREPRINT." [Online]. Available: https://youtu.be/ZCy5ESJ1gVE?si=K8CttwgTj7yGrWjn

[16] SIGIR '09 : the 32nd international ACM SIGIR conference on research and development in information retrieval, Boston, MA, USA, 19-23.07.2009. A.C.M., 2009.

[17] V. Karpukhin et al., "Dense Passage Retrieval for Open-Domain Question Answering," Sep. 2020, [Online]. Available: http://arxiv.org/abs/2004.04906

[18] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," Aug. 2019, [Online]. Available: http://arxiv.org/abs/1908.10084

[19] S. Faisal, I. Ullah, P. A. Kambey, A. Malik, and M. Shakeel, "Revolutionizing hepatic fibrosis staging: A machine learning approach combining clinical, biochemical, and microbiome insights," Computers in Biology and Medicine, vol. 206, p. 111584, 2026.

[20] A. Brasoveanu, M. Moodie, and R. Agrawal, "Textual evidence for the perfunctoriness of independent medical reviews," in CEUR Workshop Proceedings,

CEUR-WS, 2020, pp. 1–9. doi: 10.1145/nnnnnnn.nnnnnnn.

[21] D. Maio, "On the Representational Limits of Quantum-Inspired 1024-D Document Embeddings: An Experimental Evaluation Framework," Apr. 2026, [Online]. Available: http://arxiv.org/abs/2604.09430

[22] A. Khan, A. Zainab, S. H. Khan, A. Ishaq, and H. Asif, "Emergent Intelligence in Multi-Agent and LLM Systems: A Survey and Perspective Toward Autonomous, Collaborative, and Generalizable AI," Feb. 12, 2026. doi: 10.36227/techrxiv.177092236.62657640/v1.

[23] U. Zahoora, A. Khan, M. Rajarajan, S. H. Khan, M. Asam, and T. Jamal, "Ransomware detection using deep learning based unsupervised feature extraction and a cost sensitive Pareto Ensemble classifier," Sci. Rep., vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-19443-7.

[24] I. Ullah, N. Rashid, S. Babar, and N. Iqbal, "Exploring the relationship between trait emotional intelligence and performance in the context of SME software engineering," Journal of Software: Evolution and Process, vol. 38, no. 1, p. e70076, 2026.

[25] T. Rahim, A. Nazir, M. S. Tanveer, and M. R. Qureshi, "A deep learning approach to PCOS diagnosis: Two-stream CNN with transformer attention mechanism," Spectrum of Engineering Sciences, pp. 1–20, 2025.

[26] A. Khan et al., "A Recent Survey of Vision Transformers for Medical Image Segmentation," 2025, Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ACCESS.2025.3618215.

[27] H. Khan and S. Hussain Khan, "MambaFormer: Token-Level Guided Routing Mixture-of-Experts for Accurate and Efficient Clinical Assistance."

[28] H. Khan, S. Hussain Khan, M. M. Zahoor, and U. Baneen Ejaz, "Spectrum of Engineering Sciences ISSN (e) 3007-3138 (p) 3007-312X AN ENHANCED T5-LARGE TRANSFORMER FOR EFFICIENT DENTAL CLINICAL ASSISTANCE", doi: 10.5281/zenodo.18255367.

[29] "Spectrum of Engineering Sciences ISSN (e) 3007-3138 (p) 3007-312X", doi: 10.5281/zenodo.18266199.

[30] T. Hussain and S. Hussain Khan, "CGRA-DeBERTa: Concept-Guided Residual Augmentation Transformer for Theologically Islamic Understanding."

[31] U. Ahmed, A. Khan, S. Hussain Khan, A. Basit, I. U. Haq, and Y. S. Lee, "Transfer Learning and Meta Classification Based Deep Churn Prediction System for Telecom Industry."

[32] A. Ullah, G. Qi, S. Hussain, I. Ullah, and Z. Ali, "The Role of LLMs in Sustainable Smart Cities: Applications, Challenges, and Future Directions."

IEER Excellence in Education & Research