Kathmandu University Journal of Science, Engineering and Technology, Vol. 18, No. 2, December 2024

Kathmandu University
Journal of Science, Engineering and Technology

An Artificial Intelligence Driven Semantic Similarity-Based Pipeline for Rapid Literature Review

Abhiyan Dhakal[a], Kausik Paudel[a], and Sanjog Sigdel*[a]

[a]Department of Computer Science and Engineering, Kathmandu University, Nepal.

Abstract

We propose an automated pipeline for performing literature reviews using semantic similarity. Unlike traditional systematic review systems or optimization-based methods, this work emphasizes minimal overhead and high relevance by using transformer-based embeddings and cosine similarity. By providing a paper's title and abstract, it generates relevant keywords, fetches relevant papers from open-access repositories (e.g., ArXiv), and ranks them based on their semantic closeness to the input. Three embedding models: TF-IDF, all-MiniLM-L6-v2, and Specter2 were evaluated. While TF-IDF struggled with capturing deeper semantic meaning, all-MiniLM-L6-v2 provided broader conceptual coverage. Specter2, specifically fine-tuned for scientific texts, exhibited score saturation in similarity scores. A statistical thresholding approach is then applied to filter relevant papers, enabling an effective literature review pipeline. Despite the absence of heuristic feedback or ground-truth relevance labels, the proposed system shows promise as a scalable and practical tool for preliminary research and exploratory analysis.

Keywords: Literature Review; Semantic Similarity; Automated Pipeline; Transformer Embeddings; Artificial Intelligence.

1. Introduction

Conducting a literature review is an important step in any research project. By examining existing studies, researchers understand the progress that has been made in their field. Literature reviews summarizes the current state of research, and identify gaps providing insights to the direction of future studies.

Traditional systematic literature review (SLR) methods are mostly time consuming and require extensive manual efforts to select papers, filter them, and organize relevant papers. Recent automated approaches, such as Swarm SLR [1] has attempted to address these challenges by introducing structured workflow and visualization for systematic reviews, however, it requires extensive configuration, user's expertise for proper parameter tuning and possesses challenges to perform quick, exploratory literature reviews on the provided context.

Considering the research gap, we propose AutoLit, an automated pipeline designed to perform literature reviews using semantic similarity. AutoLit operates on minimal input. It requires the title and abstract of the research paper, and generates a list of relevant papers based on transformer-based embeddings and cosine similarity. The pipeline automatically generates keywords from the input, retrieves papers from open-access repositories like arXiv [2], and filters them based on semantic similarity using a customizable threshold.

This approach is useful for early-stage research and exploratory analysis, focusing on quick and relevant results rather than complete coverage. It minimizes manual intervention, and thus provides an efficient alternative to existing SLR systems for efficient literature discovery and reviews.

2. Related Works

With the development and innovation of Large Language Models (LLMs), semantic embeddings and workflow optimization techniques, automating the process of literature review has seen significant progress in the recent years. In replacement of traditional SLR, Wittenborg et al. [1] introduced SWARM-SLR that has effectively structured the systematic literature review process by defining clear requirements, establishing a modular workflow and creating a framework to evaluate the supporting tools. This has had impact in the early stages of research, such as searching other relevant papers and initial data processing. However, SWARM-SLR lacks the actual process of writing literature review.

In contrast to SWARM-SLR, Tang et al. [3] has explored the feasibility of LLMs in automating various processes of literature reviews such as reference generation, abstraction composition and complete review writing. Their evaluation showed that while LLMs showed impressive results, they often suffer from hallucination in references, favoring the tasks most cited. Additionally, generating incomplete author lists, and variable performance across various disciplines were major limitations. This is partly addressed by Li et al. [4] through their system ChatCite, which combines human workflow guidance within the LLM agent. It is a semi-automated system that generates comparative summaries between studies, which was not addressed by prior LLM based approaches. However, it relies on curated datasets and thus may not generalize summaries without further adaptation.

Reimers and Gurevych [5] introduced Sentence Bert (SBERT) for semantic similarity, which has become a baseline in many automated pipelines largely due to its efficient and accurate generation of sentence embeddings. SBERT enables

*Corresponding author. Email: sanjog.sigdel@ku.edu.np precise comparison using cosine similarity. This significantly improves tasks such as relevant literature retrieval, ranking, and clustering which is useful for understanding the closeness of different papers.

Kasanishi et al. [6] proposed SciReviewGen, a large-scale dataset and framework designed for automatic generation of literature reviews by summarizing multiple scientific papers using transformer-based models like Fusion-in-Decoder [7] extended for literature review generation. It focuses on producing summaries but still faces challenges like hallucinations and missing detailed information.

Additionally, Wang et al. [8] evaluate the capability of ChatGPT in generating Boolean queries for systematic literature searches. Their findings showed that query refinement using ChatGPT can improve the recall and F1 scores. The refined query after single prompting resulted an F1 score of 0.0772 while guided prompting increased the F1 score to 0.5171 which is a significant rise. Also the highest recall they were able to achieve was 0.9128. This indicates that combining base queries with ChatGPT refinement can generate effective Boolen queries suitable for rapid literature reviews where time constraints matter. Although LLMs can approximate human like query construction, they fall short in encoding complex Boolean logic effectively.

Despite these advances, many existing systems are based on multi-stage workflows, which can significantly hamper the performance of the system. In contrast, semantic pipelines that prioritize minimal input and provide relevant results would be beneficial for streamlining the literature review process. This approach addresses the current gap by focusing on semantic similarity-based classification and filtering of open-access repositories such as arXiv. It emphasizes transformer-based embeddings and cosine similarity for semantic search that enables the extraction of a highly relevant set of papers while performing relevant literature review of the given paper.

3. Methodology

This section outlines the pipeline designed to automate the literature review process. An overview of the pipeline is illustrated in Figure 1. The pipeline consisted of seven key stages, each designed to efficiently and systematically process scholarly articles relevant to the research topic.

3.1. Keyword Generation

The initial stage involved generating a set of relevant keywords to guide the literature search. Given the title and abstract of the target paper, 5 to 10 keywords were generated using a LLM, `gemini-2.0-flash` [9], which recent studies have shown to be effective for zero-shot keyphrase extraction [10], to capture both explicit and semantically related terms, as illustrated in Figure 2.

3.2. Fetch Papers

For each keyword generated, a corpus of a maximum of 20 relevant papers was fetched from arXiv via keyword-based search using the arXiv API. Since the papers were being fetched using different keywords, there were some duplicates. Those duplicate papers were filtered out on the basis of their metadata ensuring that only unique papers remained in the corpus.

3.3. Retrieve Relevant Papers

To refine the selection of the papers obtained, a semantic similarity assessment was performed. The sentence transformer model `all-MiniLM-L6-v2` [11] was used to generate 384-dimensional dense vector embeddings for the input query (derived from keywords) and the title and abstract of the papers obtained. The cosine similarities between the embeddings of the fetched papers and that of the input query were then calculated.

A statistical thresholding method based on the interquartile range (IQR) of the similarity scores distribution was applied to filter the papers. The IQR method is a robust, distribution-free approach for outlier detection that does not assume normality of the data [12, 13]. In line with recent NLP work [14], we adopt a conservative upper bound of $Q3+0.5\times\mathrm{IQR}$ to select only the highest-scoring papers, ensuring high precision while retaining adaptability across different similarity score distributions.

This adaptive thresholding approach accounted for the natural variance and skew in the similarity scores, resulting in a more reliable filtering of semantically relevant papers, as illustrated in Figure 3.

For comparison, we also evaluated other embedding models, including Specter2 [15], which is specifically trained on scientific citation data, and a traditional TF-IDF with cosine similarity baseline. Specter2 produced tightly clustered high similarity scores that required stricter thresholding to avoid false positives, while TF-IDF offered greater precision but lacked semantic coverage, often missing conceptually related papers. Based on this evaluation, `all-MiniLM-L6-v2` was selected as a balanced choice for semantic retrieval in this pipeline.

3.4. Text Extraction from PDF

For each filtered paper, either the abstract or the full PDF was processed to extract structured textual content. We used `PyMuPDF` [16] library for full text extraction when PDFs were available. To structure the content of each paper, regular expression (regex) patterns were defined to identify and delineate different sections: Introduction, Methodology, Results, and Conclusion. The defined patterns were as follows:

- **Abstract**: The regex pattern `(?i)\babstract\b` was used to locate the abstract section of the paper.
- **Introduction**: The pattern `(?i)\bintroduction\b` was applied to identify the introduction.
- **Methods**: To capture the methodology section, the pattern `(?i)\b(methodology|methods|approach)\b` was used, accounting for various terminologies that might describe this section.
- **Results**: To capture the results section, the pattern `(?i)\b(results|findings|experiments)\b` was employed, which might also be referred to as findings or experiments.
- **Conclusion**: The regex pattern `(?i)\b(conclusion|discussion|summary)\b` was defined to capture the conclusion, which might also appear under the names ”discussion” or ”summary” in some papers.

3.5. Summarization

The extracted text from each section of the relevant papers was then summarized using the `gemini-2.0-flash` LLM. To ensure consistent information extraction, the LLM was instructed to structure the summaries under the `summary` key, which contained four distinct categories: `problem_statement`, `methodology`, `key_findings`, and `conclusion_recommendations` (see Listing 1). Each of

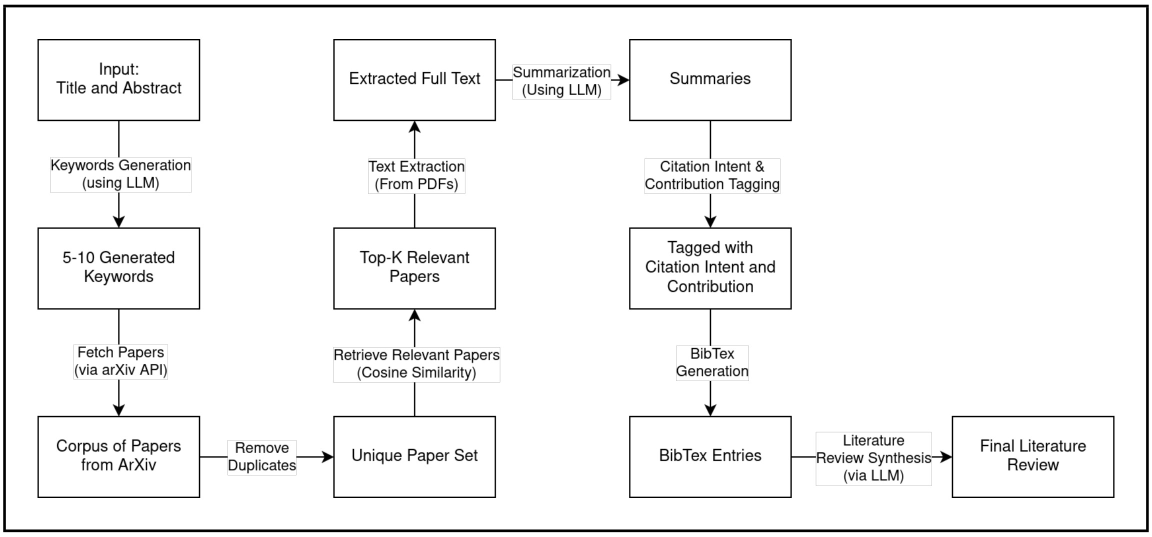

Figure 1: Pipeline Overview

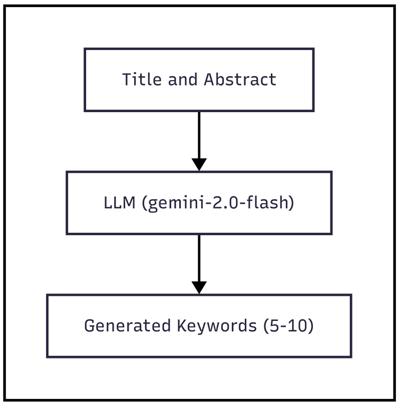

Figure 2: Keyword Generation Processs

these categories was designed to hold a list of bullet points summarizing the respective information extracted from the paper.

Listing 1: Example of structured summary JSON

```
{
  ”summary”: {
    ”problem_statement”: [
      ”Bullet point 1”,
      ”Bullet point 2”
    ],
    ”methodology”: [
      ”Bullet point 1”,
      ”Bullet point 2”
    ],
    ”key_findings”: [
      ”Bullet point 1”,
      ”Bullet point 2”
    ],
    ”conclusion_recommendations”: [
      ”Bullet point 1”,
      ”Bullet point 2”
    ]
  }
}
```

Specifically, under the `problem_statement` category, the LLM would list the identified research problems or gaps addressed in the paper as individual bullet points. The `methodology` category would contain bullet points describing the research approach, methods, or techniques employed by the authors. The `key_findings` category would present the most significant results or discoveries reported in the paper in a bulleted format. Finally, the `conclusion_recommendations` category would list the main conclusions drawn by the authors, along with any recommendations for future work or applications of their findings.

3.6. Citation Intent and Contribution Tagging

To further analyze the nature and contributions of each selected paper, the Gemini LLM was employed to perform two key tasks:

(i) **Citation intent tagging**, following a predefined taxonomy: *Background*, *Comparison*, *Extension*, *Criticism*, *Application*, *Future Work*, and *Other*.

(ii) **Contribution classification**, assigning each paper to one of: *Dataset*, *Algorithm*, *Framework*, *Review*, *Benchmark*, *Survey*, *System*, *Theoretical Analysis*, or *Other*.

This approach follows prior research on citation function classification and contribution categorization in scientific literature [17, 18, 19] and recent advances employing LLMs for automated citation intent and contribution tagging.

3.7. BibTex Entry and Literature Review Generation

The overall pipeline for generating the literature review is shown in Figure 4. It begins with metadata retrieval from the arXiv API, followed by BibTeX entry generation, paper summarization and tagging using the Gemini LLM, and finally synthesis of the literature review paragraph based on these structured inputs.

The metadata of the relevant papers, retrieved from the arXiv API in earlier stages, was utilized to generate BibTeX entries for each paper. This ensured proper citation formatting for any inclusion of these sources in the final paper.

For the generation of the literature review itself, the summaries, citation intent tags, and contribution type tags, previously obtained using the `gemini-2.0-flash` LLM, served as the primary input. The Gemini model was instructed to synthesize this information into a comprehensive overview of the existing literature related to the research topic. This step

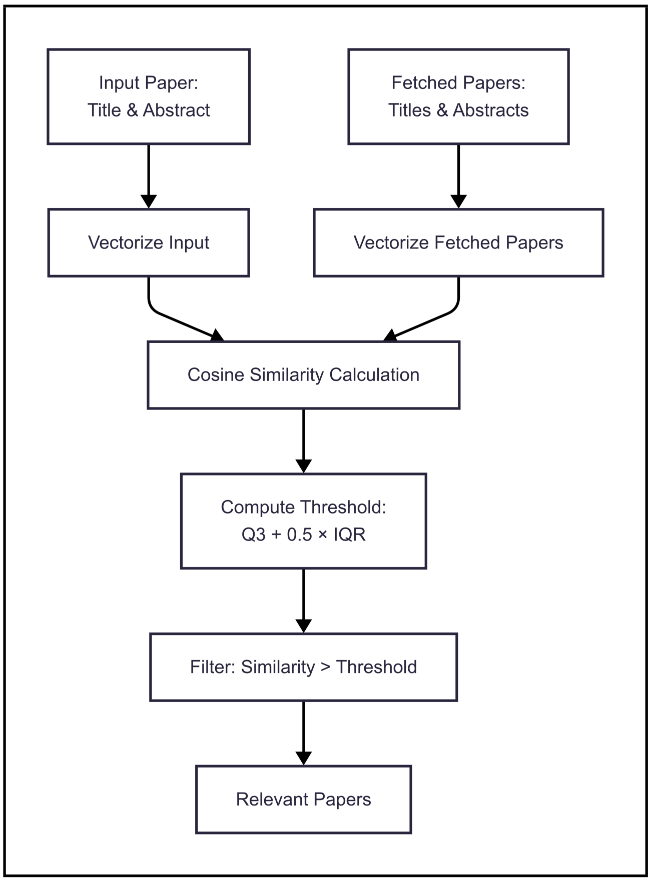

Figure 3: Relevant Papers Retrieval Processs

aimed to provide a high-level understanding of the current state-of-the-art and key contributions in the field, leveraging the automated analysis performed in the preceding stages of the pipeline.

4. Evaluation

To evaluate the effectiveness of embedding models in filtering relevant scientific papers for automated literature review, we experimented with three distinct approaches for research paper similarity:

- **TF-IDF**: Classical lexical vectorization.
- **all-MiniLM-L6-v2**: General-purpose transformer embeddings.
- **Specter2**: Scientifically tuned transformer embeddings.

A total of 178 papers were fetched from arXiv API. Each input paper's vector was compared to fetched candidates using cosine similarity. Table 1 summarizes the distributions.

4.1. TF-IDF

TF-IDF is a purely syntactic model, it lacks semantic encoding capabilities. With a calculated threshold of 0.204, it retained 19 papers. Although 19 articles were filtered, the cosine similarity distribution was highly positive and skewed toward the range [0.010, 0.294], indicating generally low similarity scores across the corpus, which is illustrated in Figure 5 and Figure 6. Also, the model failed to capture the semantic meaning of the input paper even among the selected relevant papers above the threshold. Thus, TF-IDF excels in high lexical overlap but misses conceptual matches.

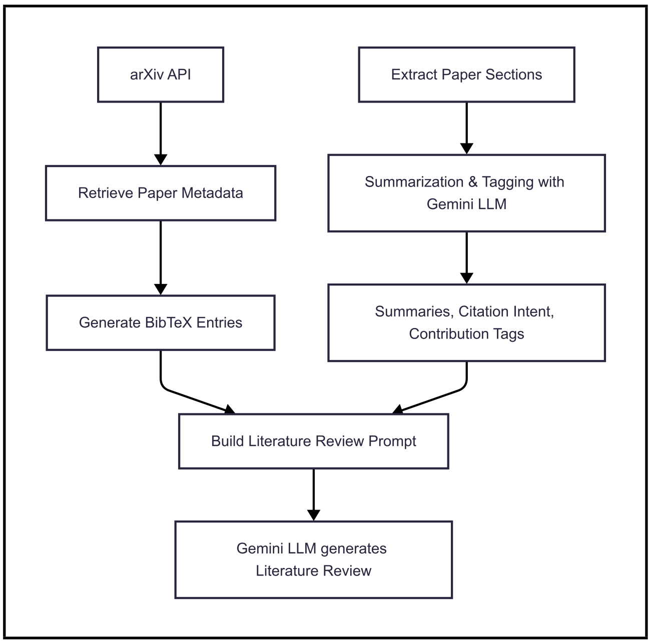

Figure 4: Litereature Review Synthesis

4.2. all-MiniLM-L6-v2

The all-MiniLM-L6-v2 model is a general-purpose semantic embedding model. With a calculated threshold set at 0.659, it retrieved 20 papers. The cosine similarity scores was slightly positively skewed in the range [0.070, 0.804] with similarity score of 0.390. Performance reflects a moderately spread distribution of similarity scores. Figure 5 and Figure 6 shows that this model provided convincing separations between relevant and less relevant documents compared to both TF-IDF and specter2. Unlike TF-IDF, this model captured semantic relations, enabling retrieval of relevant papers even when exact terminology differed.

4.3. Specter2

Specter2 is a fine-tuned model explicitly for scientific document embeddings. As a result, it achieved the highest absolute similarity scores. A threshold of 0.924 resulted in the retrieval of 11 papers, with the score distribution skewed in the range [0.756, 0.945] with a negative skewness value of 0.963. Specter2 effectively captures both semantics and domain-specific knowledge. However, with our threshold set at 0.924, the distribution exhibited higher density near 1.0. All the values were clustered at a range of 0.75 to 0.95 requiring an extremely precise threshold, which might not always be the case for varied input papers. This can result in false positive values.

5. Limitations

5.1. Lack of Ground-Truth Relevance Labels

The current system operates without any human-annotated or expert-verified relevance labels. This makes it challenging to assess whether the papers retrieved by the semantic similarity filter are truly relevant to the input query. In the absence of proper benchmarks, only indirect metrics such as similarity score distributions can be used to approximate relevance. This may not always reflect actual correctness.

Table 1: Comparative Summary of Embedding Models and Their Performance

Embedding Model	Threshold	Skewness	Value Range	# Retrieved Papers
TF-IDF	0.204	0.622	[0.010, 0.294]	19
all-MiniLM-L6-v2	0.659	0.390	[0.070, 0.804]	20
Specter2	0.924	−0.963	[0.756, 0.945]	11

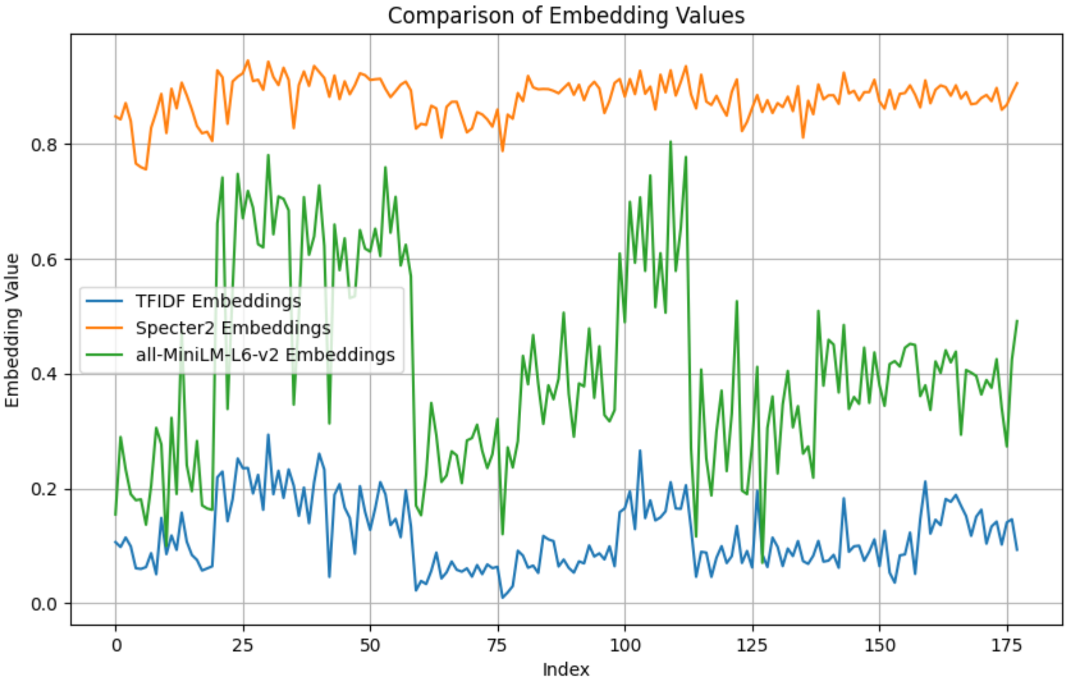

Figure 5: Line plot showing cosine similarity scores across models

5.2. Generalization Constraints of Embedding Models

While models like Specter2 are trained on scientific text, general-purpose transformers such as `all-MiniLM-L6-v2` are optimized for broad coverage across diverse domains rather than the scholarly literature specifically. Prior studies have shown that such models often underperform in specialized domains, as they may not capture domain-specific terminology, citation context, or methodological nuances with the same fidelity as models trained on in-domain corpora [20, 21, 22]. This mismatch can lead to embeddings that overlook subtle but important conceptual relationships, reducing retrieval precision in fields with specialized jargon or structured discourse, such as medicine, chemistry, or legal research.

5.3. Heuristic Thresholding Method

The thresholding method employed, using the third quartile plus half the interquartile range ($Q3 + 0.5 \cdot IQR$) is a statistical heuristic designed to filter out low-similarity papers. However, this approach does not guarantee semantic relevance. It may exclude important but lexically distant papers and include irrelevant ones that happen to lie within the upper tail of the similarity distribution. The threshold's effectiveness is also dependent on the characteristics of the embedding model and the corpus.

5.4. Lack of Qualitative Evaluation

Evaluation of the retrieved papers depends entirely on quantitative metrics such as cosine similarity distributions, mean scores, and standard deviations. There is no qualitative assessment of the relevance of the chosen papers in terms of their content or research methods. Conducting such studies would require curated gold-standard datasets or involvement of subject matter experts, which was not feasible within the current resource. Without involvement of human reviewers or case studies, it remains uncertain how well the system supports real-world tasks like literature reviews or knowledge discovery.

6. Conclusion

This work presents an automated pipeline for conducting targeted literature reviews using semantic similarity. By using transformer-based embeddings and large language models, it efficiently retrieves, filters, and summarizes academic literature with minimal human intervention. By integrating TF-IDF, all-MiniLM-L6-v2, and Specter2 embeddings, the system evaluated and filtered 178 papers based on their relevance to a given query derived from title and abstract prompts.

TF-IDF filters documents based on exact text matches, resulting in loss of semantic meaning. all-MiniLM-L6-v2 provides a balanced performance but lacks domain-specific tuning. Specter2 is best aligned with scientific language, how-

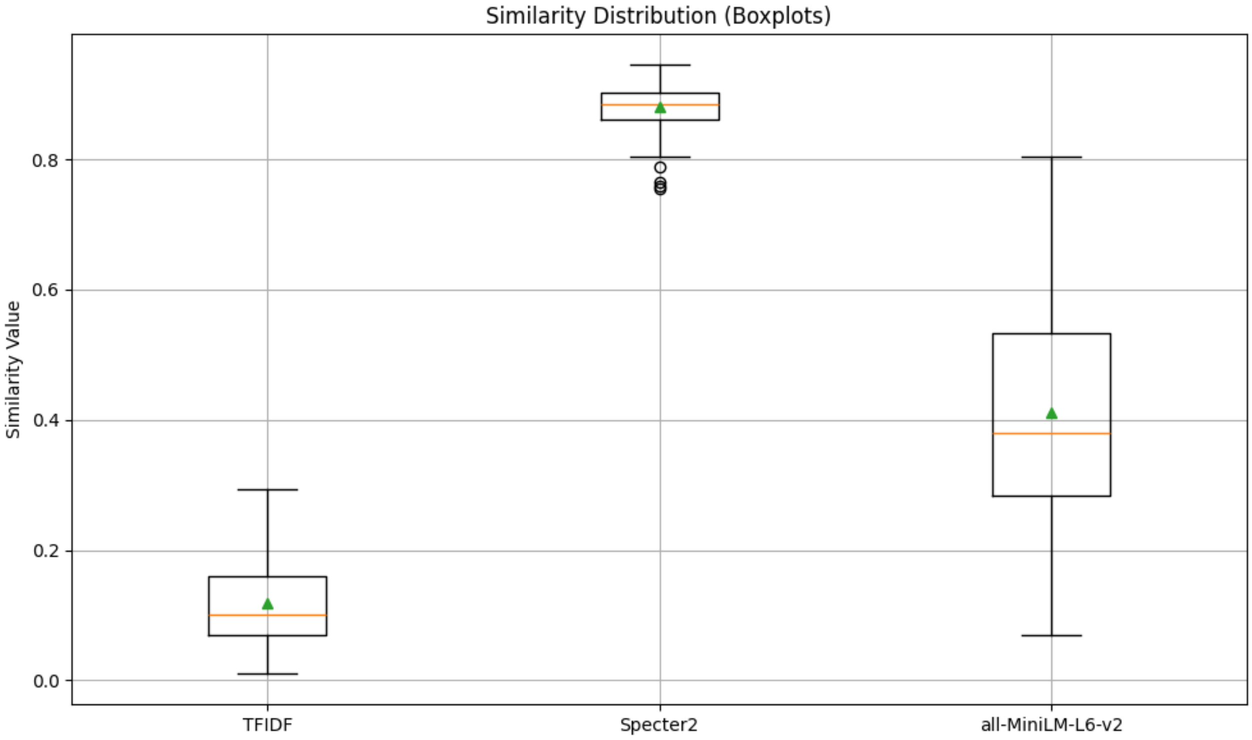

Figure 6: Similarity distribution visualization of models using box plots

ever suffered from similarity saturation, due to rigid threshold calibration.

In future work, we plan to focus on implementing adaptive thresholding to dynamically adjust based on score distribution characteristics. Incorporating human-in-the-loop evaluation will provide qualitative insights and a ground-truth basis for refining automated decisions. Re-ranking techniques will be explored to help improve the precision and contextual relevance of the retrieved papers.

Beyond thresholding and ranking improvements, domain adaptation and model fine-tuning (e.g., on specialized corpora) will be investigated to better capture field-specific semantics. Integration with citation network data sources such as Semantic Scholar [23] can enable network-based relevance scoring (e.g., PageRank-style importance measures, co-citation patterns), which may enhance both retrieval accuracy and literature synthesis. Finally, optimizing query formulation techniques will be considered to increase recall while maintaining high relevance.

This work has shown the potential to reduce manual overhead in writing literature reviews, while maintaining relevance on topics, proving it is an scalable and accessible alternative to traditional SLR systems.

References

[1] Wittenborg T, Karras O & Auer S. Swarm-slr - streamlined workflow automation for machine-actionable systematic literature reviews. In: *International Conference on Theory and Practice of Digital Libraries* (2024). URL https://api.semanticscholar.org/CorpusID:271516258.

[2] University C. arXiv API. https://arxiv.org/help/api (n.d.). Accessed: 2025-07-17.

[3] Tang X, Duan X & Cai Z G. Large language models for automated literature review: An evaluation of reference generation, abstract writing, and review composition (2024). URL https://api.semanticscholar.org/CorpusID:274822175.

[4] Li Y, Chen L, Liu A, Yu K & Wen L. Chatcite: Llm agent with human workflow guidance for comparative literature summary. In: *International Conference on Computational Linguistics* (2024). URL https://api.semanticscholar.org/CorpusID:268248855.

[5] Reimers N & Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks. In: *Conference on Empirical Methods in Natural Language Processing* (2019). URL https://api.semanticscholar.org/CorpusID:201646309.

[6] Kasanishi T, Isonuma M, Mori J & Sakata I. Scireviewgen: A large-scale dataset for automatic literature review generation. In: *Annual Meeting of the Association for Computational Linguistics* (2023). URL https://api.semanticscholar.org/CorpusID:258866124.

[7] Izacard G & Grave E. Leveraging passage retrieval with generative models for open domain question answering (2021). URL https://arxiv.org/abs/2007.01282.

[8] Wang S, Scells H, Koopman B & Zuccon G, Can chatgpt write a good boolean query for systematic review literature search?, *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval.* URL https://api.semanticscholar.org/CorpusID:256627169.

[9] Google. Gemini 2.0 Flash Large Language Model. Accessed via Google AI Studio API (2025). URL https://ai.google.dev/gemini-api/docs/models.

[10] Kang B & Shin Y. Empirical study of zero-shot keyphrase extraction with large language models. In: Rambow O, Wanner L, Apidianaki M, Al-Khalifa H, Eugenio B D & Schockaert S, eds., *Proceedings of the 31st International Conference on Computational Linguistics*. Association for Computational Linguistics, Abu Dhabi, UAE (2025), pp. 3670–3686. URL https://aclanthology.org/2025.coling-main.248/.

[11] Reimers N & Gurevych I. all-minilm-l6-v2 sentence transformer. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2 (2021). [Online; accessed 19-August-2025].

[12] Tukey J. *Exploratory Data Analysis*. No. v. 2 in Addison-Wesley series in behavioral science. Addison-Wesley Publishing Company (1977). ISBN 9780201076165. URL https://books.google.com.np/books?id=UT9dAAAAIAAJ.

[13] Ben-Gal I. Outlier detection. In: Maimon O & Rokach L, eds., *Data Mining and Knowledge Discovery Handbook*. Springer, Boston, MA (2005), pp. 131–146.

[14] Zeng X, He H, Shi H & Li T. Towards multiple references era – addressing data leakage and limited reference diversity in machine translation evaluation. In: *Findings of the Association for Computational Linguistics: ACL 2024* (2024), pp. 259–272.

[15] Singh A, D'Arcy M, Cohan A, Downey D & Feldman S. Scirepeval: A multi-format benchmark for scientific document representations. In: *Conference on Empirical Methods in Natural Language Processing* (2022). URL https://api.semanticscholar.org/CorpusID:254018137.

[16] McKie J X et al. Pymupdf: Python bindings for mupdf, a high-performance pdf and document processing library. Version 1.26.3, Artifex Software (2025). URL https://pypi.org/project/PyMuPDF/, [Online; accessed 19-August-2025].

[17] Teufel S, Siddharthan A & Tidhar D. Automatic classification of citation function. In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06. Association for Computational Linguistics, USA (2006). ISBN 1932432736, p. 103–110.

[18] Jurgens D, Finethy T, McCorriston J, Xu Y & Ruths D. Citation intent classification using deep neural networks. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)* (2018), pp. 3724–3733.

[19] Cohan A, Goharian N & Grave E. Structural scaffolds for citation intent classification in scientific publications. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)* (2019), pp. 3809–3819.

[20] Gu Y, Tinn R, Cheng H, Lucas M, Usuyama N, Liu X, Naumann T, Gao J & Poon H, Domain-specific language model pretraining for biomedical natural language processing, *ACM Trans. Comput. Healthcare*, 3(1). https://doi.org/10.1145/3458754. URL https://doi.org/10.1145/3458754.

[21] Beltagy I, Lo K & Cohan A. SciBERT: A pretrained language model for scientific text. In: Inui K, Jiang J, Ng V & Wan X, eds., *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. 3615–3620. https://doi.org/10.18653/v1/D19-1371. URL https://aclanthology.org/D19-1371/.

[22] Cohan A, Feldman S, Beltagy I, Downey D & Weld D. SPECTER: Document-level representation learning using citation-informed transformers. In: Jurafsky D, Chai J, Schluter N & Tetreault J, eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online (2020), pp. 2270–2282. https://doi.org/10.18653/v1/2020.acl-main.207. URL https://aclanthology.org/2020.acl-main.207/.

[23] Allen Institute for AI. Semantic scholar. https://www.semanticscholar.org/. [Online; accessed 11-August-2025].