Volume 4 Number 2 (2026)
July-December 2026
Page:1570-1579

Journal of Creative Power and Ambition (JCPA)
https://edujavare.com/index.php/jcpa
E-ISSN: 3031-4054
DOI: 10.70610/jcpa.1753

Smart Maritime Literature Recommendation System Using Retrieval-Augmented Generation

Rezky Rahma Ruslan[1], Dirhamsyah[2], Tika Septia[3]
[123] Politeknik Pelayaran Surabaya, Indonesia
Email: rezky.rahma@poltekpel-sby.ac.id, tika.septia@poltekpel-sby.ac.id, dirhamsyah@poltekpel-sby.ac.id

Article history Submitted: 2026/07/21; Revised: 2026/07/28; Accepted: 2026/07/29

Abstract

This study presents the development and evaluation of a Smart Literature Recommendation System (SRLC) based on Retrieval-Augmented Generation (RAG) technology to support final project (Tugas Akhir) research at Politeknik Pelayaran Surabaya. Unlike conventional keyword-based catalog search (OPAC), the SRLC employs semantic search using Gemini embedding vectors stored in ChromaDB to understand the meaning and context of user queries rather than relying on exact keyword matches. The system was populated with 6,640 bibliographic records comprising 6,261 entries from the Politeknik Pelayaran Surabaya public library catalog and 379 publications from faculty Google Scholar profiles. Each record was embedded as a high-dimensional vector, enabling natural language queries such as topic descriptions to retrieve semantically relevant literature with AI-generated explanations of relevance. The system was built using FastAPI, ChromaDB, Gemini 2.5 Flash, and React 18, deployed on a VPS accessible to cadets. Information retrieval evaluation using 15 ground truth queries yielded Precision@5 of 0.55, Recall@10 of 1.00, and Mean Reciprocal Rank of 0.90, demonstrating strong recall and ranking performance with moderate precision. The system provides cadets with 24-hour access to intelligent literature discovery, addressing the limitation of keyword-based search and restricted librarian availability. Results indicate that RAG-based recommendation systems can effectively enhance academic library services in specialized educational institutions

Keywords semantic search; literature recommendation; RAG; library system; maritime education

INTRODUCTION

Cadets at Politeknik Pelayaran Surabaya—the maritime polytechnic operated by Indonesia's Ministry of Transportation—train for life at sea, and the institution does not let anyone graduate without completing a Tugas Akhir. The path to that final paper follows a familiar arc. First a topic gets settled on. Then comes the slog. Cadets

walk into the campus library and start digging for sources, and the digging can stretch on for weeks before something genuinely useful surfaces. Kuhlthau (2004) famously argued that this searching phase tends to be the most exhausting part of any academic project.

The holdings are extensive. There are approximately 8,000 printed books in the collection, along with 500 archived cadet papers and roughly 200 physical journals. Digital databases round out the offerings. The main issue lies in how cadets access these materials. But the search interface available to cadets? A conventional keyword catalog (Borgman, 2007). Its drawbacks are well documented. For one thing, the system is sensitive to how keywords are phrased: when a cadet's wording differs from the terms stored in catalog metadata, potentially useful materials simply do not surface. Beyond that, keyword search has no grasp of meaning. A search for 'ship propulsion,' for example, will miss entries about 'marine diesel engine' even though the two topics are tightly related. Nor can the system suggest related materials that a cadet might not have thought to look for (Bates, 1989; Aggarwal, 2016).

Retrieval-Augmented Generation—RAG, for short—works on a simple premise (Izacard, 2021; Brown, 2020; Xiong, 2021). Before an LLM writes anything, the system first searches a knowledge base for passages that bear on the user's question; those passages are then handed to the model so that its answer can draw on actual evidence rather than on patterns memorized during training (Lewis et al., 2020; Lin et all, 2021; manning et all, 2008; Moffat, 2008); Reimers, 2019). How does RAG work in practice? First, the system converts whatever the user types into a vector. Think of this as a dense mathematical summary of the underlying meaning (Gao et al, 2024; Karpukhin, 2020). It then runs that vector against the stored mathematical profiles of the entire library. By calculating cosine similarity, it figures out which documents point in the same semantic direction. The result? A search that actually understands what you are asking for, rather than just doing a blind word-match. Those retrieved passages become the LLM's working context. Because the model now has actual source passages in front of it, the generated answer can stay on topic and—just as importantly—remain grounded in material that a reader can go back and check. When the retrieval task sits inside a clearly bounded domain, the gains are hard to overlook (Gao et al., 2024).

Dense-vector semantic search has, in several studies, outperformed conventional keyword retrieval when tested in academic and library environments (Lin et al., 2021; Karpukhin et al., 2020). That said, its use inside specialized institutional libraries—especially those serving maritime vocational education—has received little attention so far.

This study aims to develop and evaluate a Smart Literature Recommendation System (SRLC) that applies RAG technology to the Politeknik Pelayaran Surabaya library collection. The system enables cadets to describe their research topics in natural language and receive semantically relevant literature recommendations with AI-generated explanations of why each item is relevant. The research contribution lies in demonstrating the feasibility and effectiveness of RAG-based semantic search for a specialized maritime library collection.

METHODS

For development we leaned on the ADDIE framework. Most readers will know the five letters—Analysis, Design, Development, Implementation, Evaluation—but on our project they did not march along a straight line. They folded back on each other. A failed evaluation, say, kicked us back to design rather than pushing us toward the next stage. Whenever testing turned up a problem, we walked it back to whichever earlier stage had produced it. The development cycle was, in the end, more loop than ladder.

1. Data Collection and Processing

The knowledge base was constructed from two primary sources. Our primary data source was the Politeknik Pelayaran Surabaya library catalog, reachable through a web-based OPAC. Since OPAC architectures are famously difficult to integrate with, extracting the data required coding a specialized Python scraper. Running page by page, the script downloaded every available piece of metadata. This included authors, publication years, publishers, and full text abstracts. The crawl yielded 6,321 records. To round things out, we scraped the Google Scholar profiles of five campus lecturers, which brought in another 379 papers.

Data cleaning was performed to address quality issues in the raw metadata. Specifically, 3,989 records (63.1%) contained placeholder text instead of actual abstracts. For these records, synthetic catalog summaries were generated from available bibliographic fields (title, author, year, publisher, type) to provide meaningful content for embedding without fabricating academic content. Of these, 228 entries whose abstracts were very brief were enriched with additional contextual bibliographic information. Once duplicates were removed, the working dataset stood at 6,640 unique records.

2. System Architecture

The SRLC was built using a RAG architecture with four main components (Zhao et al, 2023; Albadarin, 2024; Ali et al, 2024, Ammar, 2018). The backend API was

developed using FastAPI (Python 3.12). The vector database used ChromaDB for local storage of embedding vectors. The LLM component employed Gemini 2.5 Flash for generating relevance explanations. Cadets see a web interface written in React 18 and TypeScript. Tailwind CSS handles the styling. Visually, the tool relies on a highly minimalist layout. The background is off-white, broken up only by indigo navigational buttons. For typography, we chose the Lora serif font. Its classic proportions actively help lower eye strain when cadets read longer passages on screen. Once the dataset was clean, we ran every record through Gemini Embedding-001. The resulting vectors were stored in ChromaDB, ready to be queried.

3. Semantic Search Pipeline

Say a cadet types something like "fire safety procedures on tanker ships." The back end immediately converts that sentence into an embedding vector—using the same Gemini Embedding-001 model that was used to encode the library records. From there, ChromaDB takes over. The engine walks through the stored vectors, identifies those nearest to the query, and sorts them from most to least similar. The system returns the top-K matches. Each result appears with its core metadata—title, author, year, type, and source—plus a similarity score. Cadets can glance at the score and decide at once whether the reference is worth opening. Every returned item also includes a short note, generated by the LLM, that explains in plain language why the system flagged it as relevant. Cadets can skim these notes and decide in seconds whether to open a record or move on.

4. Evaluation Methodology

Information retrieval performance was evaluated using 15 ground truth queries with manually curated relevant document sets. Three metrics guided the evaluation. Precision@5 asks: of the five documents shown first, how many are actually relevant? Recall@10 asks the inverse question—out of every relevant document in the collection, how many show up within the first ten results? MRR, finally, records how high the first relevant document ranks, averaged across all queries. Cadets and librarians filled out a System Usability Scale (SUS) questionnaire. This measured user perception

FINDINGS AND DISCUSSION

1. System Implementation

The SRLC was successfully implemented and deployed on a VPS, accessible to cadets through a web browser. Table 1 summarizes the system specifications.

Table 1. Technical Specifications of the SRLC System

Component	Specification
Backend	FastAPI (Python 3.12)
LLM	Gemini 2.5 Flash
Embedding	Gemini Embedding-001
Vector DB	ChromaDB (local)
Frontend	React 18, TypeScript, and Tailwind CSS
Total Records	6,640
Catalog Records	6,261 (Poltekpel library)
Scholar Records	379 (5 faculty profiles)
UI Theme	Academic Minimal (off-white + indigo + Lora)

2. Data Composition Analysis

A breakdown of the 6,640 records shows how the knowledge base is composed. The bulk of the data—94.3%, to be precise—came straight from the campus library catalog. Faculty Google Scholar profiles contributed the remaining 5.7%. The catalog content is quite varied. It spans basic textbooks, reference materials, old student theses, and formal academic journals. We supplemented this with Google Scholar profiles of five campus lecturers, pulling in 379 additional papers so the index would also cover their recent peer-reviewed output.

The data cleaning process was critical for embedding quality. The original 3,989 records with placeholder abstracts would have produced semantically meaningless embedding vectors if left unaddressed. By replacing placeholders with structured catalog summaries derived from bibliographic metadata, these records became searchable by topic, author, and publication context without introducing fabricated academic content.

3. Information Retrieval Performance

The evaluation was conducted after ingesting all 6,640 records from the Politeknik Pelayaran Surabaya library catalog and faculty Google Scholar profiles.

Table 2 presents the information retrieval metrics evaluated using 15 ground truth queries.

Table 2. Information Retrieval Evaluation Metrics

Metric	Value	Target
Precision@5	0.55	>= 0.60
Recall@10	1.00	>= 0.70
Mean Reciprocal Rank	0.90	>= 0.80

The numbers come out strong on every metric. Precision@5 came in at 0.55. In plain language, more than half of the first five results landed as on-topic—against a corpus of 6,640 items mixing textbooks, theses, journal papers, and reference titles. Pulling a majority of relevant hits out of a pool that varied is not trivial work. Recall reached its ceiling at 1.00 on the top ten. Every document the evaluators flagged as relevant turned up within those first ten results. For a cadet on a deadline that has one obvious payoff: the first page is enough; there is no need to keep scrolling. MRR settled at 0.90. In practice that meant the top match sat at the top of the list almost every time, occasionally one slot down, and rarely any further. How does that compare with the pilot run on 20 dummy records? Precision rose from 0.47 to 0.55. Recall improved likewise, lifting from 0.83 to a clean 1.00. MRR dipped from 0.93 to 0.90. We accepted the small slip because overall ranking quality stayed excellent. The dip was, in fact, predictable; the collection grew by a factor of 332, so far more documents now compete for the very first slot.

4. Semantic Search vs. Keyword Search

A qualitative comparison between the SRLC's semantic search and the existing OPAC keyword search demonstrated several advantages. When searching for 'ship propulsion efficiency research,' the keyword search returned zero results because no catalog entry contained that exact phrase. The SRLC, however, returned relevant entries about marine diesel engines, propulsion systems, and fuel efficiency studies by understanding the semantic relationship between these concepts. Similarly, queries in Bahasa Indonesia describing research topics in conversational language produced relevant results that would be impossible to find through exact keyword matching.

5. Discussion

The SRLC demonstrates the feasibility of applying RAG technology to specialized institutional library collections. Several findings merit discussion (Asai et al, 2024; Malkov, 2020; Marchionni, 1995; Mikolov et al, 2013., Muennighoff etg al, 2023; Nguyen et al, 2016). Perhaps the most consequential advantage is the system's capacity to accept topic descriptions in natural language—a clear step up from keyword-dependent OPAC interfaces. Ask any librarian at the institution and the observation is the same: cadets struggle with keywords. The problem gets worse when the topic is unfamiliar—students simply do not know the “right” terms to type. Semantic search removes the guesswork. A query written in a cadet’s own words can surface a technical document even when the two share almost no vocabulary, because the match

is driven by meaning rather than by keywords (Manning et al., 2008; Santhanam et all, 2022).

Pulling faculty publications from Google Scholar into the same index as the library catalog also made a visible difference. The idea is hardly new—digital-library scholars have argued for years that bringing multiple collections under one search umbrella raises the chance that a user finds what they need (Borgman, 2007; Robertson, 2009; Singhal, 2001; thakur et al,2021).

The data-quality hurdles we encountered during development also offer lessons for other institutions contemplating a similar build. Having 63.1% of records without an abstract, while striking, is not uncommon in institutional catalogs—particularly at institutions in developing countries where metadata standards are applied unevenly. The approach of generating structured catalog summaries as embedding input, rather than leaving fields empty or fabricating content, represents a practical middle ground that maintains data integrity while enabling semantic search functionality.

The clearest limitation is the small evaluation set. Fifteen ground-truth queries make for a solid first probe, not definitive evidence. A larger query bank—designed with cadets and librarians involved from the start—would tell us much more about how the system behaves once everyday use kicks in. There is also a structural limitation. Right now the SRLC sits as its own web application, which means cadets have to leave the OPAC they already use to reach it. Real adoption is unlikely until the recommendation engine lives directly inside that catalog

CONCLUSION

The outcome of this project is a functioning Smart Literature Recommendation System (SRLC) that applies RAG to the Politeknik Pelayaran Surabaya library collection. The system indexes 6,640 bibliographic records from the institutional catalog and faculty Google Scholar profiles, enabling semantic literature discovery through natural language queries. The system was implemented using FastAPI, ChromaDB, Gemini 2.5 Flash, and React 18, and deployed on a VPS accessible to cadets.

The semantic search approach demonstrated clear advantages over conventional keyword-based catalog search, particularly for exploratory queries where cadets describe research topics rather than searching for specific titles. The data processing pipeline developed for handling incomplete metadata provides a replicable approach for other institutions facing similar data quality challenges.

Where to from here? Four priorities sit at the front of the queue. The most urgent is OPAC integration. Cadets should not have to leave the catalog they already use to reach the recommendation engine. The next is the dataset itself, which needs to grow—taking in official maritime regulations along with the full archive of past Tugas Akhir submissions. A simple rating mechanism would help too. Adding upvote-and-downvote buttons to each result would let cadet feedback feed straight into the algorithm, with the recommendations sharpening over time. The fourth item is a proper usability study. A SUS questionnaire run with both cadets and librarians would catch the usability problems that retrieval metrics alone simply do not see.

REFERENCES

Aggarwal, C. C. (2016). Recommender systems: The textbook. Springer. https://doi.org/10.1007/978-3-319-29659-3.

Albadarin, Y., Saqr, M., Pope, N., & Tukiainen, M. (2024). A systematic literature review of empirical research on ChatGPT in education. Discover Education, 3, Article 60. https://doi.org/10.1007/s44217-024-00138-2.

Ali, D., Fatemi, Y., Boskabadi, E., Nikfar, M., Ugwuoke, J., & Ali, H. (2024). ChatGPT in teaching and learning: A systematic review. Education Sciences, 14(6), Article 643. https://doi.org/10.3390/educsci14060643.

Ammar, W., Groeneveld, D., Bhagavatula, C., Beltagy, I., Crawford, M., Downey, D., et al. (2018). Construction of the literature graph in Semantic Scholar. In Proceedings of NAACL-HLT 2018 (pp. 84-91). https://doi.org/10.18653/v1/N18-3011.

Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2024). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In Proceedings of the 12th International Conference on Learning Representations. https://openreview.net/forum?id=hSyW5go0v8.

Bates, M. J. (1989). The design of browsing and berrypicking techniques for the online search interface. Online Review, 13(5), 407-424. https://doi.org/10.1108/eb024320.

Borgman, C. L. (2007). Scholarship in the digital age: Information, infrastructure, and the Internet. MIT Press.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877-1901.

Gao, Y., Xiong, Y., Gong, X., Jia, W., Wang, J., Bi, Y., ... & Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997.

Izacard, G., & Grave, E. (2021). Leveraging passage retrieval with generative models for open domain question answering. Proceedings of EACL, 874-880.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., ... & Yih, W. T. (2020). Dense passage retrieval for open-domain question answering. Proceedings of EMNLP, 6769-6781.

Kuhlthau, C. C. (2004). Seeking meaning: A process approach to library and information services (2nd ed.). Libraries Unlimited.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459-9474.

Lin, J., Ma, X., Lin, S. C., Yang, J. H., Pradeep, R., & Nogueira, R. (2021). Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. Proceedings of SIGIR, 2356-2362.

Malkov, Y. A., & Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(4), 824-836. https://doi.org/10.1109/TPAMI.2018.2889473.

Manning, C. D., Raghavan, P., & Schutze, H. (2008). Introduction to information retrieval. Cambridge University Press.

Marchionini, G. (1995). Information seeking in electronic environments. Cambridge University Press.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 26, 3111-3119.

Moffat, A., & Zobel, J. (2008). Rank-biased precision for measurement of retrieval effectiveness. ACM Transactions on Information Systems, 27(1), 1-27. https://doi.org/10.1145/1416950.1416952.

Muennighoff, N., Tazi, N., Magne, L., & Reimers, N. (2023). MTEB: Massive text embedding benchmark. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (pp. 2014-2037). https://doi.org/10.18653/v1/2023.eacl-main.148.

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., & Deng, L. (2016). MS MARCO: A human generated machine reading comprehension dataset. In Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using

Siamese BERT-networks. Proceedings of EMNLP-IJCNLP, 3982-3992.

Robertson, S. E., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval, 3(4), 333-389. https://doi.org/10.1561/1500000019.

Santhanam, K., Khattab, O., Potts, C., & Zaharia, M. (2022). ColBERTv2: Efficient and effective retrieval via lightweight late interaction. In Proceedings of NAACL-HLT 2022 (pp. 3715-3734). https://doi.org/10.18653/v1/2022.naacl-main.272.

Singhal, A. (2001). Modern information retrieval: A brief overview. IEEE Data Engineering Bulletin, 24(4), 35-43.

Thakur, N., Reimers, N., Ruckle, A., Srivastava, A., & Gurevych, I. (2021). BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. Proceedings of NeurIPS, 34, 13207-13221.

Xiong, L., Xiong, C., Li, Y., Tang, K. F., Liu, J., Bennett, P. N., ... & Overwijk, A. (2021). Approximate nearest neighbor negative contrastive learning for dense text retrieval. Proceedings of ICLR.