Effective Generative Artificial Intelligence: Case Study Automation on AI-Powered Reference Curation on Release Management

Jimmy Agung Gunawan1,a), Moses Laksono Singgih2,b), and Raden Venantius Hari Ginardi3,c)

1Interdisciplinary School of Management Technology, Intitut Teknologi Sepuluh Nopember, Surabaya, Indonesia

2Department of Industrial and Systems Engineering, Intitut Teknologi Sepuluh Nopember, Surabaya, Indonesia

3Department of Information Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

a) Corresponding author: [jimmy.a.gunawan@googlemail.com](mailto:jimmy.a.gunawan@googlemail.com)

b) moseslsinggih@its.ac.id

c) hari@its.ac.id

**Abstract.** The integration of Generative Artificial Intelligence (Gen AI) into Release Management methodologies, is a game-changing approach to automate release management end to end. This study analyzes the potential of advanced AI technologies to improve efficiency, security, and resource consumption in software development lifecycles, operations and security measurements. The major goal is to create and build a conceptual framework that uses Gen AI and LLMs to automate operations like code review, testing, and deployment, hence enabling concurrent engineering techniques and reducing lead times. Insights from this review guide the design of a framework that integrates AI capabilities into existing DevSecOps pipelines, emphasizing continuous integration/continuous delivery (CI/CD) and security integration. This study underscores the transformative potential of AI in modern release management, offering a path toward more efficient, secure, and resource-effective DevSecOps practices.

**Keywords:** Generative Artificial Intelligence, Large Language Models, AI-driven automations, Software Development, Release Management

# INTRODUCTION

Academic writing is an essential aspect of research activities, yet it can be an arduous and time-intensive endeavor., researchers often spend significant time and effort in creating study literature references, ensuring that the sources they cite are relevant, accurate, and properly formatted [1]. However, the conventional methods of manually searching for relevant sources and formatting references can be inefficient and prone to errors [2]. To address these challenges, AI support can be developed using methods such as Retrieval Augmented Generation (RAG) and embedding journals. RAG is a method that combines retrieval-based and generative models to generate high-quality study literature references [3]. Integrating AI support empowers researchers to efficiently collect and format references, significantly saving time while enhancing the precision and uniformity of their citations [4]. Moreover, AI tools can assist researchers in identifying the most influential authors, top contributing universities, and dominant themes within their field of study [5]. Using AI tools based on methods like Retrieval Augmented Generation and embedding journals, researchers can streamline the process of gathering and formatting study [6], more efficient and accurate study literature references [3]. While the use of AI in referencing certainly offers efficiency and accuracy, there are concerns regarding the potential limitations and drawbacks of integrating AI support into the academic writing process [7]. One of the primary concerns is the reliance on AI tools for sourcing and formatting references, which may lead to the neglect of critical thinking and scrutiny that are essential in the research process [8]. Moreover, there are concerns regarding intellectual property and copyright issues, if sources are not properly cited or if copyrighted materials are selected without authorization, there could be potential infringement [9].

## Research Background

Incorporating AI into the academic writing process offers significant advantages in enhancing the efficiency and effectiveness of generating literature references. AI tools utilize techniques such as Retrieval Augmented Generation and journal embedding, researchers can drastically speed up the collection and formatting of references, thus saving valuable time [10]. AI integration not only streamlines the referencing process but also improves the accuracy and consistency of citations, addressing issues related to incorrect formatting and inconsistencies [11]. Furthermore, AI tools can assist in creating reliable datasets for specific research by identifying relevant sources, extracting bibliographic details, and organizing references according to the required citation style. This reduces the likelihood of errors and ensures dependable and consistent referencing in academic writing [12]. The effectiveness of AI in supporting literature reviews is evident in its seamless integration, allowing researchers to identify key authors, prominent universities, and prevalent themes in their field of study [13]. This contributes to a comprehensive and insightful literature review, thereby enhancing the quality of research findings. The necessity of adopting AI in generating literature references arises from its significant impact on expediting processes, providing trustworthy datasets, and demonstrating efficiency in literature reviews [14]. As the academic community increasingly embraces technological advancements, it is essential to consider the potential limitations and challenges of AI integration to uphold the highest standards of academic integrity and critical analysis in research [15].

## Research Questions

The main research question is how to create datasets that reliable, and very useful to extract information from documents (pdf) that uploaded to the server AI with Large Language Models (LLM) driven. The research question revolves around the creation of reliable datasets and extracting information from uploaded PDF documents using AI with Large Language Models driven servers[7]. Effectiveness of GenAI and LLM: How does the integration of GenAI with LLMs improve the accuracy and reliability of literature references generated from a vast database of PDF journal articles? Optimization techniques: What are the most effective techniques for optimizing the retrieval process in GenAI systems to handle large-scale databases of PDF documents for efficient reference generation? And comparative analysis: How does the performance of a GenAI RAG-enhanced LLM system compared to traditional literature review methods in terms of speed, comprehensiveness, and citation relevance when processing extensive collections of PDF journals?

## Research Purpose and Benefit

The purpose of this research is to develop AI support for creating study literature references using methods like RAG, LLM, and embedded journals in order to streamline the literature review process and improve the accuracy and efficiency of generating references [16]. Large Language Model (LLM) refers to a type of artificial intelligence system designed to understand, generate, and manipulate human language at scale [17]. LLMs are trained on vast datasets containing diverse text samples. The training involves adjusting the model’s parameters to minimize the difference between its outputs and the expected results. This process, known as unsupervised learning, allows the model to predict the probability of a word or sequence of words, given a context [18]. Once trained, LLMs can perform a variety of language tasks, such as translation, summarization, question-answering, and content generation. Their ability to understand context and generate coherent and contextually relevant text makes them powerful tools for natural language processing (NLP) applications [19]. LLMs have been integrated into various industries, including customer service, where they power chatbots and virtual assistants; content creation, where they assist in writing and editing; and education, where they provide tutoring and language learning support [20]. Despite their capabilities, LLMs face challenges such as bias, which can arise from imbalances in training data. Ensuring ethical use and mitigating potential harms, such as misinformation, requires careful oversight and continuous refinement of the models. Large Language Models are advanced AI systems that have revolutionized the field of NLP by enabling machines to interact with human language in unprecedented ways, while also presenting new challenges that necessitate responsible development and deployment [21]. Using methods like RAG and embedding journals, AI can support the process of creating study literature references from extensive collections of scientific articles [17]. AI can significantly aid in creating literature references from extensive collections by utilizing methods like RAG and journal embedding [22] [23]. RAG method combines a retriever to search through documents for relevant information and a generator to produce informed answers. It is particularly beneficial for integrating a private knowledge base with Large Language Models (LLMs) to develop Generative Q&A systems [11]. Embedding Journals involves creating dense numerical representations of words that capture both semantic and syntactic information. These representations can be utilized to match articles to journals and enhance content models for scientific articles [24] [25], a Hammer PDF developed and help to do specific journal reading on specific research pdf papers [26]. These AI-driven methods improve the efficiency and accuracy of creating literature references by leveraging advanced search and retrieval techniques [7]. All of the RAG and LLM processes develop on-premises to save cost on tokens in commercial LLM like: ChatGPT, Claude, Mistral, or Gemini. And datasets or vector databases can be reused over and over again or even redevelop by recompiling with state of the art library or journals.

## Retrieval Augmented Generation (RAG)

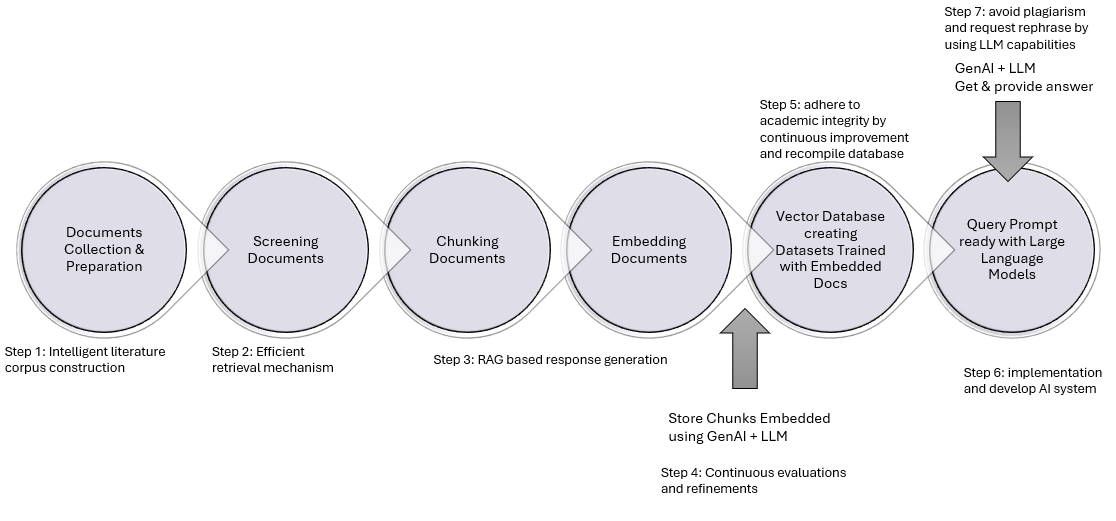
Scientific literature plays a pivotal role in disseminating groundbreaking discoveries. However, valuable data often remain buried within the vast ocean of publications. In materials engineering, critical information is scattered across technical handbooks, specification sheets, journal articles, and laboratory notebooks [27]. In this review, we explore the foundations of RAG, practical applications, and potential future advancements. RAG techniques are advanced methods used to enhance the capabilities of language models by incorporating external information. RAG techniques work best in four ways: Relevance vs Similarity which understanding the difference between relevance, which is about the connectedness of ideas, and similarity, which is about matching words, chunking strategy or segmenting text into meaningful ‘chunks’ to improve the efficiency of information retrieval and context location in a RAG system [28], query augmentation by enhancing queries with additional context to retrieve more relevant information for the language model [29], setup use cases since RAG techniques are applied in various domains, such as building production apps, question answering services, and chat-with-data applications [11] [30].

## Embedding Documents

Using Retrieval-Augmented Generation (RAG) to embed documents entails several steps aimed at boosting the performance of language models through retrieving pertinent information from a knowledge base [31]. The RAG system utilizes various techniques to embed documents effectively, including encoding the text into numerical representations that capture both semantic and syntactic information [11]. After that chunking and processing by dividing documents into smaller "chunks" that can be processed more efficiently by the model [32] [33]. This process entails segmenting the text into meaningful and manageable-sized portions [34]. Embedding Generation, this step converting text from every chunk into an embedding, which is a dense vector representation that captures the semantic meaning of the text [35]. These embeddings are generated using an encoder model, which is trained to produce vectors that reflect the content and context of the text, once the embeddings are generated, they are stored in a vector database. This database acts as an internal search engine that can quickly retrieve relevant document chunks based on their embeddings [36]. When a user query is received and identified by LLM, the RAG system uses the query to search the vector database for the most relevant document embeddings [37]. The retrieved document chunks are fed to the Reader Model, a language model that synthesizes the information from the retrieved chunks to provide a coherent and contextually relevant response [38] [39]. Last but not least, the Reader Model generates an answer to the user’s query, considering both the original question and the additional context provided by the retrieved document chunks [40]. This process allows the RAG system to provide more accurate and contextually relevant answers by leveraging the vast amount of information contained in the knowledge base. It’s a powerful way to augment the capabilities of language models with external knowledge sources for a practical example of implementing RAG.

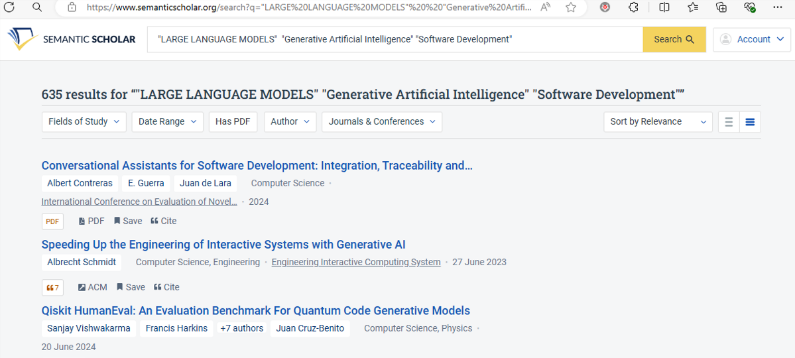
# METHODS

This paper is targeting to implement AI support for creating study literature references using methods RAG and embedding journals [11], the following steps are recommended as shown in **FIGURE 1**.



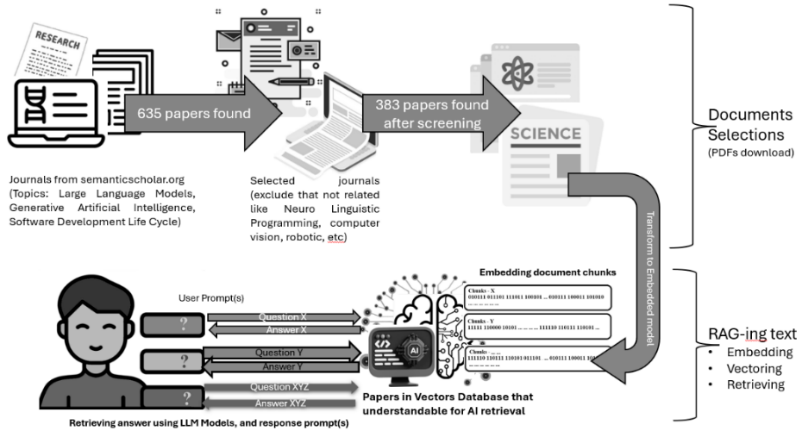
**FIGURE 1**. Steps of Effective GenAI and LLM on Processing Literatures

Step 1: Construct a Comprehensive Literature Corpus, by gather a large corpus of study literature in pdf, including research papers, journal articles, and other relevant documents. Step 2: Integration of Retrieval Mechanism, by implement a retrieval mechanism that is based on dense vector representations to efficiently retrieve relevant passages from the literature corpus when presented with a question or prompt. Step 3: Utilize the RAG Model for Response Generation, by leverage the capabilities of the RAG model to generate a response by synthesizing the retrieved passages with the original query. Step 4: Evaluation of Generated Responses, by evaluate the accuracy and relevance of the generated responses by comparing them to established references and guidelines for study literature references. Step 5: Development of Training Methods, by developing a training method, such as the proposed inFO-RAG, to optimize the RAG model for generating concise, accurate, and complete study literature references. Step 6: Implementation of AI System for Reference Generation, deploy the developed AI system to generate study literature references by inputting queries or prompts related to the desired reference. Step 7: Avoiding Plagiarism, by ensuring that the retrieved passages are used as references and sources for creating original study literature references, avoiding any form of plagiarism. Following those steps enables effective implementation of AI support for creating literature references using advanced methods like RAG and embedding journals. The integration of AI-driven techniques significantly streamlines the process of literature reference creation while maintaining the highest standards of accuracy and integrity.



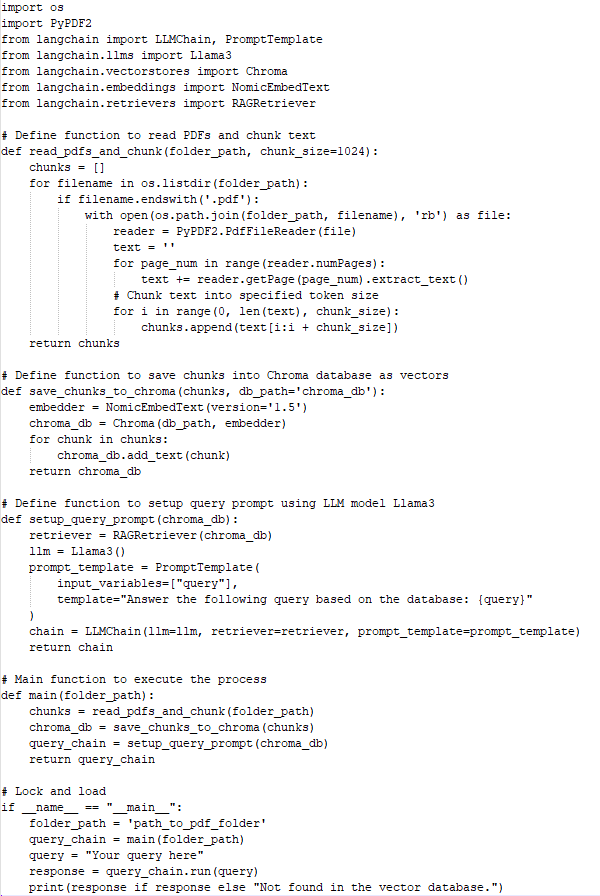
**FIGURE 2**. Search and Found Topic GenAI, LLM and Software Development in Semantic Scholar

This research collected journal documents using the Semantic Scholar features (<https://semanticscholar.org/>) as shown in **FIGURE 2**, there are 635 results found for the keywords: “**Large Language Models**”, “**Generative Artificial Intelligence**” and “**Software Development**” with the limitations of having PDF version in the archived database. All of these documents will be downloaded, collected into one folder, and then run as feeders for the GenAI, then clustered and stored into database. Every document categorized into abstracts, main findings, methodology, limitation, study gap, and study objectives. Then stored as dataset into the database, these datasets will be used as knowledges for the AI. Chat prompt served to answer the questions based on the datasets of knowledges. This process shown in **FIGURE 3**.



**FIGURE 3**. Process and Transfer Documents into On-Premises Vector Database

Document selections part still need manually read by researchers to perform a well screening for the pre-datasets. But the RAG-ing and LLM parts provided by the help of Python-Code as shown in **FIGURE 4**.

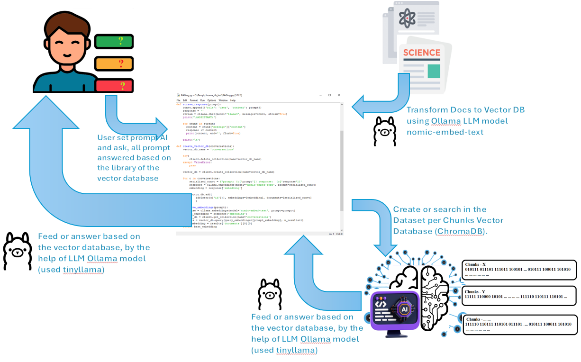


**FIGURE 4.** Python Code to RAG and LLM Prompt Retrieval Answer in 56 lines

Model nomic-embed-text was selected to produce a better summary and findings for the chunks information, this model proves that very fast, effective and small to load into memory (RAM) but provide an efficient embed model. On the other hand, the llama3 model was selected because of its accuracy and capability of running on 8GB RAM. Hardware used for this research using specification: CPU (Intel Core i5 1135G7 Gen 11th 2.4GHz), Memory (DDR4 2x8GB), and Disk (NVME M.2 VGen 512GB). And software to run: Linux Ubuntu 64Bits 24.04 LTS, Python ver. 3.12.3, Ollama model llama3 for prompt answer and nomic-embed-text version 1.5 for embedding documents.

# RESULTS

Integrating Generative AI (GenAI or GAI) and Large Language Models (LLMs) with Retrieval-Augmented Generation (RAG) offers a revolutionary method for automating information extraction from PDF documents. Companies can create custom datasets for specific operational procedures or confidential content and run them locally on-premises. Applying RAG to a large document corpus shown promising result on deep-learning representations that can be reused effectively. The AI processes and stores information in a 'machine language' format using vector databases, enabling dataset sharing, re-learning, and adaptation to new information. Empirical data analysis and a conceptual framework provide a comprehensive understanding of the impact and feasibility of using Gen AI and LLMs to automate tasks such as extracting conclusions, limitations, methodologies, and summaries from individual or large collections of documents.



**FIGURE 5**. Generate Datasets and Prompt Ready LLM

The RAG model shown in **FIGURE 5**, with the help of Ollama LLM (https://ollama.com/), offers a revolutionary method for automating information extraction from PDF documents. The Ollama models used divide into two models, nomic-embed-text for creating text-data (per chunks) and then saved into Sqlite Database (chromaDB). The Database will serve to the prompt, to avoid bias the first prompt generated is finding conclusions, limitations, methodology and finding in the documents. The first prompt also stated that LLM should behave like researchers of Machine Learning, LLM and GenAI point of view to make a library from the chunks. The prompt will help to rephrase and help to avoid plagiarism in the future. The metadata collected (datasets) in **TABLE 1** can be used to support the future reference that will be needed later.

**TABLE 1**. Results prompt answer using dataset generated

|  |
| --- |
| **PROMPT & ANSWERING** |
| **Prompt:** ﻿How does the integration of GenAI with LLMs improve the accuracy and reliability of literature references generated from a vast database of PDF journal articles?  **Answer:** |
| **Prompt:** ﻿What are the most effective techniques for optimizing the retrieval process in GenAI systems to handle large-scale databases of PDF documents for efficient reference generation? Explain in short less than 200 words, but if you cannot find the answer from the localdoc just say you didn't find the answer.  **Answer:** |
| **Prompt:** ﻿﻿How does the performance of a GenAI RAG-enhanced LLM system compared to traditional literature review methods in terms of speed, comprehensiveness, and citation relevance when processing extensive collections of PDF journals? Explain in short less than 200 words, but if you cannot find the answer from the localdoc just say you didn't find the answer.  **Answer:** ﻿ |
| **﻿ Prompt: ﻿﻿**create python script working as api that receive picture and doing a compare face on picture with database, put comment and definition in details.  **Answer: ﻿**Sure, I can guide you through the process of creating such API using Python Flask for backend service, OpenCV library to do facial recognition. For simplicity's sake, let's assume we have already trained our model (FaceNet) on a database with known faces and labels. |

# CONCLUSIONS

Implementing GenAI and LLMs poses challenges such as data quality control, bias avoidance, resource management, model fine-tuning, security, privacy, interpretability, and scalability. To overcome these hurdles, technical expertise, domain knowledge, and thoughtful design are required. This paper explores using open-source and on-premises solutions for software development and release management, offering cost-effective AI, confidentiality, scalability, custom development, collaboration, decision-making, and automation. These technologies enable companies to maintain confidential documents on-premises also, avoiding cloud limitations. Ongoing research is necessary to harness the full potential of these technologies in practical use cases.

# Acknowledgments

I would like to express my deepest gratitude to my advisor, Professor Moses L. Singgih, and Dr. Raden Venantius Hari Ginardi, for their unwavering support and guidance. Their feedback and expertise were invaluable. Thanks to the Interdisciplinary School of Management Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, and the university staff for their support. Lastly, I really appreciate my family, colleagues, superiors, and friends for their constant support, encouragement and inspiration via online social media or offline.

# References

1. A. Kali, "Reference management: A critical element of scientific writing," *Journal of Advanced Pharmaceutical Technology & Research,* vol. 7, no. 1, pp. 27-29, 2016.
2. A. Duffy and F. Lawler, "Getting it right: the art of academic referencing," *British Journal of Healthcare Assistants,* vol. 13, no. 9, pp. 444-448, 2019.
3. G. Wagner, R. Lukyanenko, and G. Paré, "Artificial intelligence and the conduct of literature reviews," *Journal of Information Technology,* vol. 37, no. 2, pp. 209-226, 2022.
4. M. A. Kacena, L. I. Plotkin, and J. C. Fehrenbacher, "The use of artificial intelligence in writing scientific review articles," *Current Osteoporosis Reports,* vol. 22, no. 1, pp. 115-121, 2024.
5. Z. Lin, "Techniques for supercharging academic writing with generative AI," *Nature Biomedical Engineering,* pp. 1-6, 2024.
6. G. Conroy, "How ChatGPT and other AI tools could disrupt scientific publishing," *Nature,* vol. 622, no. 7982, pp. 234-236, 2023.
7. A. Carobene, A. Padoan, F. Cabitza, G. Banfi, and M. Plebani, "Rising adoption of artificial intelligence in scientific publishing: evaluating the role, risks, and ethical implications in paper drafting and review process," *Clinical Chemistry and Laboratory Medicine (CCLM),* vol. 62, no. 5, pp. 835-843, 2024.
8. M. Hosseini, L. M. Rasmussen, and D. B. Resnik, "Using AI to write scholarly publications," ed: Taylor & Francis, 2023, pp. 1-9.
9. A. Y. Gasparyan, M. Yessirkepov, A. A. Voronov, A. N. Gerasimov, E. I. Kostyukova, and G. D. Kitas, "Preserving the integrity of citations and references by all stakeholders of science communication," *Journal of Korean medical science,* vol. 30, no. 11, pp. 1545-1552, 2015.
10. K. Roy, V. Khandelwal, V. Vera, H. Surana, H. Heckman, and A. Sheth, "GEAR-Up: Generative AI and External Knowledge-Based Retrieval: Upgrading Scholarly Article Searches for Systematic Reviews," vol. 38, pp. 23823-23825.
11. Y. Gao *et al.*, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997,* 2023.
12. T. Koltay, "Facing the challenge of data-intensive research: Research data services and data literacy in academic libraries," in *Innovation in libraries and information services*, vol. 35: Emerald Group Publishing Limited, 2016, pp. 45-61.
13. H. I. A. Razack, S. T. Mathew, F. F. A. Saad, and S. A. Alqahtani, "Artificial intelligence-assisted tools for redefining the communication landscape of the scholarly world," *science editing,* vol. 8, no. 2, pp. 134-144, 2021.
14. F. Bolanos, A. Salatino, F. Osborne, and E. Motta, "Artificial intelligence for literature reviews: Opportunities and challenges," *arXiv preprint arXiv:2402.08565,* 2024.
15. C. França, "AI empowering research: 10 ways how science can benefit from AI," *arXiv e-prints,* pp. arXiv-2307, 2023.
16. K. Kousha and M. Thelwall, "Artificial intelligence technologies to support research assessment: A review," *arXiv preprint arXiv:2212.06574,* 2022.
17. W. X. Zhao *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223,* 2023.
18. P. Ramachandran, P. J. Liu, and Q. V. Le, "Unsupervised pretraining for sequence to sequence learning," *arXiv preprint arXiv:1611.02683,* 2016.
19. S. Pahune and M. Chandrasekharan, "Several categories of large language models (llms): A short survey," *arXiv preprint arXiv:2307.10188,* 2023.
20. W. Gan, Z. Qi, J. Wu, and J. C.-W. Lin, "Large language models in education: Vision and opportunities," pp. 4776-4785: IEEE.
21. H. Xiong, J. Bian, S. Yang, X. Zhang, L. Kong, and D. Zhang, "Natural language based context modeling and reasoning with llms: A tutorial," *arXiv preprint arXiv:2309.15074,* 2023.
22. J. Lála, O. O'Donoghue, A. Shtedritski, S. Cox, S. G. Rodriques, and A. D. White, "Paperqa: Retrieval-augmented generative agent for scientific research," *arXiv preprint arXiv:2312.07559,* 2023.
23. J. de la Torre-López, A. Ramírez, and J. R. Romero, "Artificial intelligence to automate the systematic review of scientific literature," *Computing,* vol. 105, no. 10, pp. 2171-2194, 2023.
24. A. Bordes, S. Chopra, and J. Weston, "Question answering with subgraph embeddings," *arXiv preprint arXiv:1406.3676,* 2014.
25. N. Gu and R. H. R. Hahnloser, "SciLit: A Platform for Joint Scientific Literature Discovery, Summarization and Citation Generation," *arXiv preprint arXiv:2306.03535,* 2023.
26. S.-F. Wang *et al.*, "Hammer pdf: An intelligent pdf reader for scientific papers," pp. 5019-5023.
27. C. H. Ward, J. A. Warren, and R. J. Hanisch, "Making materials science and engineering data more valuable research products," *Integrating Materials and Manufacturing Innovation,* vol. 3, no. 1, pp. 292-308, 2014.
28. G. Yang, E. Sutinen, and D. Wen, "Chunking and extracting text content for mobile learning: A query-focused summarizer based on relevance language model," pp. 126-128: IEEE.
29. L. A. Ramshaw and M. P. Marcus, "Text chunking using transformation-based learning," in *Natural language processing using very large corpora*: Springer, 1999, pp. 157-176.
30. I. Ilin, "Advanced rag techniques: an illustrated overview," ed, 2023.
31. B. Cao *et al.*, "Retrieval is accurate generation," *arXiv preprint arXiv:2402.17532,* 2024.
32. T. Cakaloglu and X. Xu, "A multi-resolution word embedding for document retrieval from large unstructured knowledge bases," *arXiv preprint arXiv:1902.00663,* 2019.
33. M. Kuribayashi, T. Fukushima, and N. Funabiki, "Robust and secure data hiding for PDF text document," *IEICE TRANSACTIONS on Information and Systems,* vol. 102, no. 1, pp. 41-47, 2019.
34. C. Jebari, M. J. Cobo, and E. Herrera-Viedma, "A new approach for implicit citation extraction," pp. 121-129: Springer.
35. A. Padmakumar and A. Saran, "Unsupervised text summarization using sentence embeddings," *Technical Report, University of Texas at Austin,* pp. 1-9, 2016.
36. S. Kanwal, S. Nawaz, M. K. Malik, and Z. Nawaz, "A review of text-based recommendation systems," *IEEE Access,* vol. 9, pp. 31638-31661, 2021.
37. Q. Le and T. Mikolov, "Distributed representations of sentences and documents," pp. 1188-1196: PMLR.
38. A. Abdallah and A. Jatowt, "Generator-retriever-generator: A novel approach to open-domain question answering," *arXiv preprint arXiv:2307.11278,* 2023.
39. X. Chen, P. Gao, J. Song, and X. Tan, "HiQA: A Hierarchical Contextual Augmentation RAG for Massive Documents QA," *arXiv preprint arXiv:2402.01767,* 2024.
40. A. Amini, R. Cotterell, J. Hewitt, L. Malagutti, C. Meister, and T. Pimentel, "Generating text from language models," pp. 27-31.