SYM-OP-IS 2025
52nd Symposium on Operational Research
Palić, 7 - 10.9.2025.

UNIVERSITY OF BELGRADE
FACULTY OF ORGANIZATIONAL SCIENCES

Review article **https://doi.org/10.5281/zenodo.17532060**

A SYSTEMATIC REVIEW OF VECTOR DATABASE USE IN RETRIEVAL-AUGMENTED GENERATION FOR LLM-BASED EDUCATIONAL PLATFORMS

FILIP STAMENKOVIĆ[1], JELICA STANOJEVIĆ[2], MIROSLAV MINOVIĆ[3]

[1] University of Belgrade, Faculty of Organizational Sciences, Belgrade, filip.stamenkovic@fon.bg.ac.rs, 0009-0008-4575-5892
[2] University of Belgrade, Faculty of Organizational Sciences, Belgrade, jelica.stanojevic@fon.bg.ac.rs, 0000-0002-0570-0167
[3] University of Belgrade, Faculty of Organizational Sciences, Belgrade, miroslav.minovic@fon.bg.ac.rs, 0000-0002-4270-7595

Abstract: *This systematic review explores the use of vector databases in Retrieval-Augmented Generation (RAG) for educational platforms based on large language models (LLMs). As RAG becomes a promising approach to enhance the contextual accuracy of LLM outputs by retrieving relevant content, vector databases serve as a core component for storing and retrieving embedded educational materials. This review is comprised of 9 studies from 2023 to 2025, focusing on use cases in higher education, including domain-specific applications and chatbots for student and educator support. Findings show diverse choices of vector stores, such as FAISS, Chroma, Qdrant, Weaviate, Milvus, Vectara, MongoDB and Postgres with pgVector, often combined with orchestration frameworks like LangChain or LlamaIndex. The reporting on embedding models, orchestration frameworks and system architecture is inconsistent, limiting the comparability of studies and reducing confidence in synthesizing performance trends, which impacts the reliability of conclusions drawn from the review. The findings provide a reference point for researchers and developers creating context-aware, LLM-based educational platforms, and suggest future research directions including performance benchmarking, model transparency, and evaluating learning outcomes.*

Keywords: *Retrieval-Augmented Generation, Large Language Models, Vector Databases, LLMs in Education.*

1. INTRODUCTION

Generative AI, especially large language models (LLMs), can support both students and teachers by making complex material easier to understand (Laato et al., 2023).

One useful method that enhances LLM performance by adding relevant outside information during the response process is Retrieval-Augmented Generation (RAG). RAG retrieves data from a database using vector search and gives it to the model as context (Gao et al., 2024). This makes the model's answers more accurate and useful, especially in education, managing data like lecture notes, textbooks, or slides (Mullins et al., 2024).

Vector databases play an important role in these systems. They store content as numerical embeddings, which helps the system better understand what the user is asking. This leads to better answers and also cuts down on hallucinations, made-up or wrong responses that can happen when LLMs rely only on their training data (Jing et al., 2024).

Different systems use different tools, depending on their goals. Educational systems using RAG vary depending on their purpose. Some are made as chatbots that help students by answering their questions and giving support in learning (Wölfel et al., 2023), while others are aimed at educators, helping them generate exam questions, prepare teaching materials, or analyze course content (Hennekeuser et al., 2024; Nikolovski et al., 2025). In this review, educational platforms are defined as tools and environments primarily aimed at higher education.

Even though the number of educational applications which utilize RAG is increasing, there is limited research that systematically examines how vector databases are integrated and applied in these systems. To better understand how these systems work, this review explores two main questions:

- RQ1: Which vector databases are used in educational RAG-based systems?
- RQ2: How are these databases used in different studies, and what kinds of content and tasks do they support?

By analyzing recent research, this paper provides a clearer picture of how RAG and vector databases are used in education and identifies directions for future research.

In the following sections, the paper describes the methodology used for selecting and analyzing relevant studies (Section 2), presents key findings on the use of vector databases in educational RAG systems (Section 3), discusses the findings (Section 4), and concludes with a summary of main insights and application possibilities (Section 5).

2. METHODS

The literature research was conducted using two major scientific databases: Scopus and Web of Science. These databases were selected due to their wide use in systematic reviews and their extensive coverage of high-quality papers in the chosen fields. The databases feature a reliable advanced search engine based on user-defined criteria.

The goal of the review was to identify studies that explore the use of Retrieval-Augmented Generation (RAG) in combination with large language models (LLMs) within educational platforms, with a particular focus on the selection and use of vector databases that support the RAG technique.

To identify relevant studies, queries were constructed for each database using logical operators (AND, OR) and truncation symbols (*). The search fields were limited to title, abstract, and keywords. The queries targeted papers that mention key concepts related to: Retrieval-Augmented Generation, Large Language Models, Platform or API integration, and educational context. The following search queries were used, tailored to the syntax of each database:

- **Scopus:** TITLE-ABS-KEY ("retrieval-augmented generation" OR "RAG") AND TITLE-ABS-KEY ("large language model" OR "LLM") AND TITLE-ABS-KEY ("LLM-based" OR "LLM-powered" OR "API" OR "application programming interface" OR "chatbot*") AND TITLE-ABS-KEY ("education" OR "educational platform*").
- **Web of Science**: TS=("retrieval-augmented generation" OR "RAG") AND TS=("large language model" OR "LLM") AND TS=("LLM-based" OR "LLM-powered" OR "API" OR "application programming interface" OR "chatbot*") AND TS=("education" OR "educational platform*").

Although the focus of this review is on the use of vector databases in RAG LLM-based systems, terms such as "vector database" or "vector store" were not included in the search queries. Preliminary searches including the terms "vector database" or "vector store" were conducted to test their impact on retrieval. When these terms were added to the Web of Science query, only 2 studies were returned, fully overlapping with results already captured by the broader query. In Scopus, adding these terms yielded 6 studies, again fully overlapping with results already captured by the broader query. This confirmed that including these terms unnecessarily narrows the search without contributing additional relevant studies. The search was conducted on June 6, 2025.

The study selection process followed the PRISMA 2020 guidelines (Page et al., 2021). After conducting the database search, duplicate records were removed. The remaining papers were then screened by examining their titles, keywords, and abstracts. Studies that were not relevant were excluded. Then, the full texts of the selected papers were read. More papers were excluded if they did not use RAG, did not include an actual LLM-based system, or did not give out vector database implementation details. Ultimately, only studies that met all inclusion criteria were retained. Two reviewers independently screened all titles, abstracts, and full texts, resolving any disagreements through discussion to reach consensus. Although this dual-review process reduces the risk of selection bias, inter-rater reliability statistics were not calculated, which limits the ability to quantitatively assess the consistency of reviewer judgments.

3. RESULTS

After finding the initial 70 papers and then removing duplicates, a total of 55 unique studies were identified. The titles, keywords and abstracts of these papers were then screened, and 25 studies were excluded for not meeting basic relevance criteria.

Of the 30 remaining studies that were sought for retrieval, 8 of them were not retrieved successfully, having resulted in a reading and review of the 22 retrieved papers. During the reading phase, an additional 12 papers were excluded. The number of excluded papers for each exclusion reasons were: absence of RAG use in the implementation (n = 1), lack of actual implementation of an LLM-based system (n = 3) and absence of vector database implementation details (n = 8). The full selection process is summarized in the PRISMA 2020 flow diagram (Page et al., 2021) shown in Figure 1.

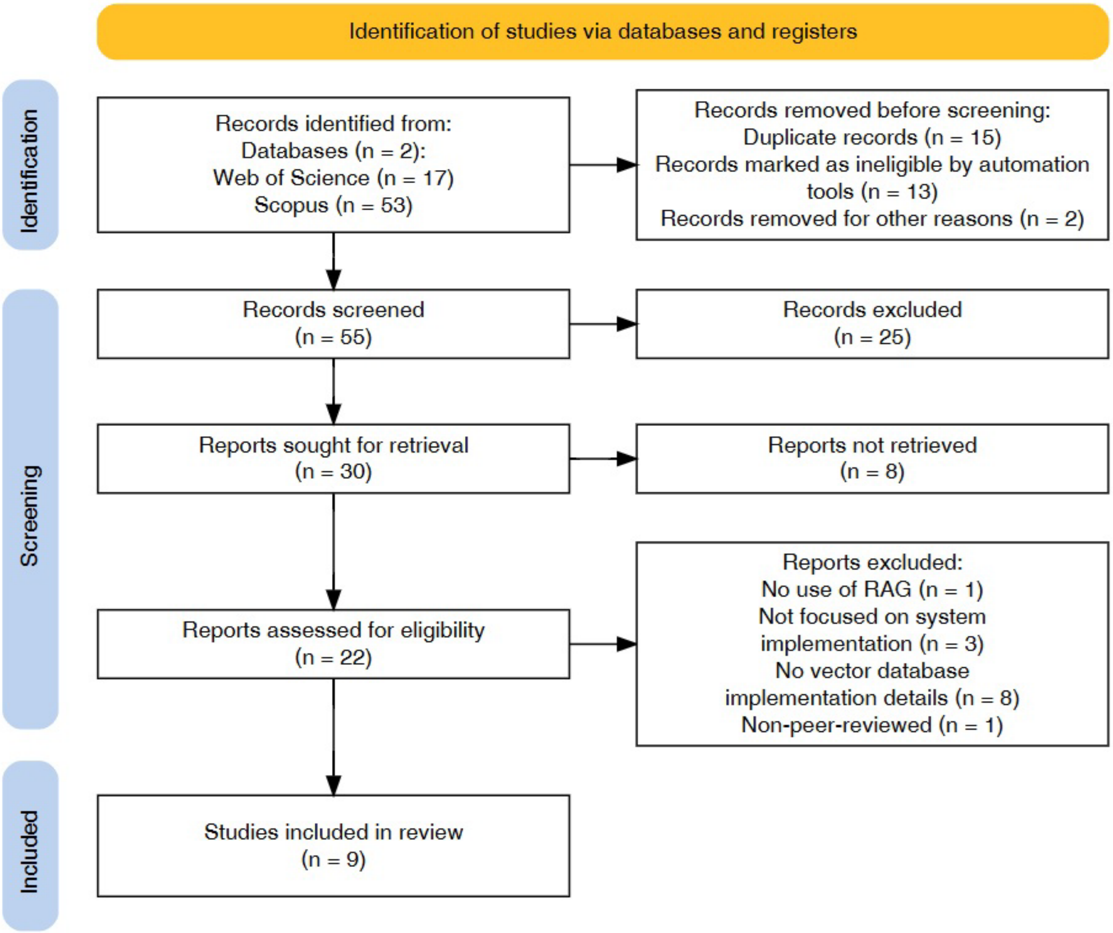
Identification of studies via databases and registers
Identification
Screening
Included
Records identified from:
Databases (n = 2):
Web of Science (n = 17)
Scopus (n = 53)
Records removed before screening:
Duplicate records (n = 15)
Records marked as ineligible by automation tools (n = 13)
Records removed for other reasons (n = 2)
Records screened
(n = 55)
Records excluded
(n = 25)
Reports sought for retrieval
(n = 30)
Reports not retrieved
(n = 8)
Reports assessed for eligibility
(n = 22)
Reports excluded:
No use of RAG (n = 1)
Not focused on system implementation (n = 3)
No vector database implementation details (n = 8)
Non-peer-reviewed (n = 1)
Studies included in review
(n = 9)

Figure 1: PRISMA 2020 flow diagram

The review included 9 studies published between 2023 and 2025, with all of them focused on the use of Retrieval-Augmented Generation (RAG) with large language models (LLMs) in educational settings. The selected papers mainly describe applications and tools designed to assist students and educators in higher education through improved content retrieval and generation. Below is a summary of the key characteristics of the papers which were included:

Table 1: Summary of the included papers

Paper	Education domain	LLM	Vector database	Additional framework	Embedding model
Advancing AI in Higher Education: A Comparative Study of Large Language Model-Based Agents for Exam Question Generation, Improvement, and Evaluation (Nikolovski et al., 2025)	General education assistance for educators	Gemini 2.0	Qdrant	LangChain	Not specified
Towards an AI tutor for undergraduate geotechnical engineering: a comparative study of evaluating the efficiency of large language model application programming interfaces (Tophel et al., 2024)	Geotechnical engineering	GPT-4 Llama-3	FAISS	LangChain	OpenAI's embedding model
Enlarged Education - Exploring the Use of Generative AI to Support Lecturing in Higher Education (Hennekeuser et al., 2024)	General education assistance for educators	GPT-4	Milvus	Not specified	OpenAI's text-embedding-ada-002 model
ChatPapers: An AI Chatbot for Interacting with Academic Research (Dean et al., 2023)	Academic research	GPT-4	Postgres with pgVector	LangChain	all-MiniLM-L6-V2 model from

					Hugging Face
An LLM-Driven Chatbot in Higher Education for Databases and Information Systems (Neumann et al., 2025)	Computer science	GPT-4	Weaviate	LangChain LlamaIndex	OpenAI's embedding model
Knowledge-Based and Generative-AI-Driven Pedagogical Conversational Agents: A Comparative Study of Grice's Cooperative Principles and Trust (Wölfel et al., 2023)	General education assistance for students	GPT-4 GPT-3.5	Chroma	LangChain	Not specified
MerryQuery: A Trustworthy LLM-Powered Tool Providing Personalized Support for Educators and Students (Tabarsi et al., 2025)	General education assistance for students and educators	GPT-4o	MongoDB	LangChain	Not specified
Enhancing Engineering Education Through LLM-Driven Adaptive Quiz Generation: A RAG-Based Approach (Gopi et al., 2024)	Engineering education and quiz generation	GPT-4	Vectara	Not specified	Not specified
MEDCO: Medical Education Copilots Based on a Multi-agent Framework (Wei et al., 2024)	Medical education	GPT-3.5 Claude 3.5 Sonnet	Chroma	Not specified	OpenAI's embedding model

The described systems vary mainly in their purpose. Most studies present chatbots that help students learn different subjects, while others focus on tools for generating exam questions or academic explanations. These systems use embedding techniques to turn educational content into vectors, making it easier to find relevant information based on meaning, not just keywords. The reviewed studies used a range of vector databases, with integration methods differing across implementations.

Vector databases in the nine studies were: FAISS (1), Chroma (2), Qdrant (1), Weaviate (1), Milvus (1), Vectara (1), MongoDB with vector search (1), and Postgres with pgVector (1). All but Vectara were open-source. LangChain was used in 6 studies, and LlamaIndex in 1 study (combined with LangChain). Embedding models were inconsistently reported: 2 studies used OpenAI's text-embedding-ada-002, 2 unspecified OpenAI models, 1 used all-MiniLM-L6-v2, and 4 studies did not report the model.

In systems built for educators, Nikolovski et al. (2025) used Qdrant with LangChain to store and access data from lecture slides, notes, code examples, textbook excerpts, and lab guides to help create and check exam questions. The PDFs were turned into markdown files, split into chunks, and embedded for quick search. However, the authors did not provide further explanation about why Qdrant was chosen over other available options. Hennekeuser et al. (2024) built a similar assistant using Milvus and OpenAI's text-embedding-ada-002 model. It helped teachers access content from PDFs for course preparation, like books and scripts. The authors write that Milvus's support for multiple similarity search functions makes it well-suited for efficient retrieval in this educational setting.

For students, vector search changed how chatbots work, fitting them closer to student needs. Wölfel et al. (2023) used Chroma and LangChain to build a system that worked with lecture slides and transcripts, helping students through more responsive conversations. Chroma was used as vector database due to its efficiency in leveraging embeddings to represent documents as vectors and retrieving relevant information based on semantic similarity. However, the authors did not explain why Chroma was chosen over other options. Tabarsi et al. (2025) used MongoDB with vector search to support both students and teachers, letting them retrieve PDFs with text, tables, and images. MongoDB was also used to store interaction logs. This choice simplified system architecture by combining transactional data and vector search in one platform.

In engineering education, vector search was key for retrieving specialized knowledge. Tophel et al. (2024) used FAISS and LangChain to help create a chatbot in geotechnical engineering, working with files in PDF, DOCX, and TXT formats. OpenAI embeddings helped store and find the right information. FAISS was used because of its proven efficiency in large-scale similarity search, being well suited for technical domains like engineering. Gopi et al. (2024) used Vectara and the MathVista dataset to make quiz questions for engineering students, based on subject-specific reasoning. Vectara, an API platform, simplified data ingestion and retrieval with optimized defaults for chunking and preprocessing.

In computer science, Neumann et al. (2025) built a chatbot using Weaviate, LangChain, and LlamaIndex to help students access learning content. PDF materials were embedded with OpenAI's model to get accurate answers. The authors did not provide further details on the selection of Weaviate for the system. Dean et al.

(2023) built a research assistant using Postgres with pgVector and LangChain. It used Hugging Face's all-MiniLM-L6-V2 model to store and find papers and metadata for computer science research. The system leveraged vector search to quickly retrieve relevant research papers based on user queries, improving the efficiency of literature review processes. PgVector allowed the efficient storage and querying of both vectorized data and associated metadata, such as paper titles, authors, and publication details.

In medical education, Wei et al. (2024) used Chroma for efficient storage and rapid queries in a multi-agent tutoring system, with roles like patient, radiologist, and expert, to deliver content across specialties. Data was embedded using OpenAI's model, and vector search pulled relevant cases from the MVME dataset. This helped create a more reliable learning experience for medical students.

4. DISCUSSION

The reviewed studies show various methods for using vector search in educational RAG-based systems, providing answers to both RQ1 and RQ2. All systems rely on vector-based retrieval, but vary in their choice of vector database, embedding model, and orchestration tools.

Used vector stores include FAISS, Chroma, Qdrant, Weaviate, Milvus, Vectara, MongoDB and Postgres with pgVector. These databases stored different data types including lecture slides, research papers, textbooks, and course materials, depending on the system's domain and goals. They were often combined with orchestration frameworks like LangChain or LlamaIndex. This distribution indicates a clear preference for open-source solutions and highlights a major gap in reporting: 4 out of 9 of studies omitted embedding model details. Such omissions reduce reproducibility and complicate meaningful comparisons of retrieval performance across implementations. While OpenAI's embedding models are most common, alternatives like all-MiniLM-L6-V2 are also shown as viable.

A notable gap in the reviewed studies is the lack of explanation for why specific vector databases were chosen. Most papers only mention which database was used and note that it worked efficiently, without discussing the reasons behind the choice. Some studies, like those using Postgres or MongoDB, stored both vectors and metadata or interaction logs in one system instead of using a dedicated vector database. This simplifies architecture but raises questions about trade-offs compared to specialized solutions. Institutional preference may also influence these decisions. Future work should explain database selection more clearly to enable better comparisons of effectiveness across contexts.

For practitioners, this review suggests that RAG systems in education can be adapted to different institutional needs. Future work should benchmark vector stores and embedding models using metrics like retrieval accuracy, latency, scalability, and cost-performance trade-offs, comparing open-source solutions with cloud-hosted proprietary databases like Vectara. Additionally, studies should assess how retrieval design impacts student learning outcomes, trust in AI, and engagement.

Ethical considerations are essential when discussing AI in education. Huang (2023) highlights the importance of student data privacy, risks of unauthorized data use, and the need for transparency in AI processing of sensitive information. They stress the necessity of accountability mechanisms from institutions and developers. While this review focuses on technical aspects, these ethical concerns are crucial to ensure responsible AI use in education, respecting students' privacy rights.

One methodological limitation of this review is its reliance solely on Scopus and Web of Science for literature retrieval. While these databases offer high-quality content, expanding the search to include other sources, such as Google Scholar, could uncover additional implementations and broaden the understanding of RAG-based systems in education.

5. CONCLUSION

This review explores how vector databases are used in educational RAG systems, emphasizing their adaptability to various educational needs, whether for student support, teacher assistance, or subject-specific focus. This adaptability allows for the creation of RAG systems tailored to different educational goals, although more details are needed to fully understand the potential of these systems.

A major limitation is the lack of detailed technical descriptions, especially regarding embedding models and system architecture. While selected studies mention which vector databases were used, they rarely explain why they were chosen, and several omit embedding model details entirely. This omission is significant, as it limits reproducibility, hinders meaningful comparisons of system performance, and reduces confidence in the broader conclusions of the review. This review highlights that the lack of standardized reporting not only complicates synthesis but also makes it difficult to determine which combinations of databases and models work best for specific educational tasks.

Despite these challenges, the findings provide a useful starting point for designing more context-aware educational systems. Future research could focus on benchmarking different vector stores and models, exploring their impact on learning outcomes, and establishing clearer reporting practices to support more stable comparisons.

LITERATURE

Dean, M., Bond, R. R., McTear, M. F., & Mulvenna, M. D. (2023). ChatPapers: An AI Chatbot for Interacting with Academic Research. *2023 31st Irish Conference on Artificial Intelligence and Cognitive Science (AICS)*, 1–7. https://doi.org/10.1109/AICS60730.2023.10470521

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). *Retrieval-Augmented Generation for Large Language Models: A Survey* (No. arXiv:2312.10997). arXiv. https://doi.org/10.48550/arXiv.2312.10997

Gopi, S., Sreekanth, D., & Dehbozorgi, N. (2024). Enhancing Engineering Education Through LLM-Driven Adaptive Quiz Generation: A RAG-Based Approach. *FIE 2024 Conference Proceedings*, 1–8. https://doi.org/10.1109/FIE61694.2024.10893146

Hennekeuser, D., Vaziri, D. D., Golchinfar, D., Schreiber, D., & Stevens, G. (2024). Enlarged Education – Exploring the Use of Generative AI to Support Lecturing in Higher Education. *International Journal of Artificial Intelligence in Education*. https://doi.org/10.1007/s40593-024-00424-y

Huang, L. (2023). Ethics of Artificial Intelligence in Education: Student Privacy and Data Protection. *Science Insights Education Frontiers*, *16*(2), 2577–2587. https://doi.org/10.15354/sief.23.re202

Jing, Z., Su, Y., & Han, Y. (2024). *When Large Language Models Meet Vector Databases: A Survey* (No. arXiv:2402.01763). arXiv. https://doi.org/10.48550/arXiv.2402.01763

Laato, S., Morschheuser, B., Hamari, J., & Björne, J. (2023). AI-Assisted Learning with ChatGPT and Large Language Models: Implications for Higher Education. *2023 IEEE International Conference on Advanced Learning Technologies (ICALT)*, 226–230. https://doi.org/10.1109/ICALT58122.2023.00072

Mullins, E. A., Portillo, A., Ruiz-Rohena, K., & Piplai, A. (2024). *Enhancing classroom teaching with LLMs and RAG* (No. arXiv:2411.04341). arXiv. https://doi.org/10.48550/arXiv.2411.04341

Neumann, A. T., Yin, Y., Sowe, S., Decker, S., & Jarke, M. (2025). An LLM-Driven Chatbot in Higher Education for Databases and Information Systems. *IEEE Transactions on Education*, *68*(1), 103–116. https://doi.org/10.1109/TE.2024.3467912

Nikolovski, V., Trajanov, D., & Chorbev, I. (2025). Advancing AI in Higher Education: A Comparative Study of Large Language Model-Based Agents for Exam Question Generation, Improvement, and Evaluation. *Algorithms*, *18*(3), 144. https://doi.org/10.3390/a18030144

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, n71. https://doi.org/10.1136/bmj.n71

Tabarsi, B., Basarkar, A., Liu, X., Xu, D. (DK), & Barnes, T. (2025). MerryQuery: A Trustworthy LLM-Powered Tool Providing Personalized Support for Educators and Students. *Proceedings of the AAAI Conference on Artificial Intelligence*, *39*(28), 29700–29702. https://doi.org/10.1609/aaai.v39i28.35372

Tophel, A., Chen, L., Hettiyadura, U., & Kodikara, J. (2024). *Towards an AI Tutor for Undergraduate Geotechnical Engineering: A Comparative Study of Evaluating the Efficiency of Large Language Model Application Programming Interfaces*. In Review. https://doi.org/10.21203/rs.3.rs-4658661/v1

Wei, H., Qiu, J., Yu, H., & Yuan, W. (2024). *MEDCO: Medical Education Copilots Based on A Multi-Agent Framework* (No. arXiv:2408.12496). arXiv. https://doi.org/10.48550/arXiv.2408.12496

Wölfel, M., Shirzad, M. B., Reich, A., & Anderer, K. (2023). Knowledge-Based and Generative-AI-Driven Pedagogical Conversational Agents: A Comparative Study of Grice's Cooperative Principles and Trust. *Big Data and Cognitive Computing*, *8*(1), 2. https://doi.org/10.3390/bdcc8010002