ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 1, January 2026

SmartLexicon: A Retrieval Augmented Generation Based Application

[1] Kalim Mulani, [2] Sahil Sapate, [3] Naila Tamboli, [4] Pradeep Taware

[1] [2] [3] [4] Department of Computer Engineering, G H Raisoni College of Engineering and Management, Pune, Maharashtra, India

Corresponding Author Email: [1] kalim.mulani.cs@ghrcem.raisoni.net, [2] sahil.sapate.cs@ghrcem.raisoni.net, [3] naila.tamboli.cs@ghrcem.raisoni.net, [4] pradeep.taware@raisoni.net

Abstract*— SmartLexicon has been called a state of the art Retrieval-Augmented Generation system aimed at making the AI-generated response more accurate and trust-worthy within some specialized domains like healthcare, law, and engineering. While earlier generative AI models relied heavily on static old datasets, SmartLexicon infused real-time data retrieval with intelligent response generation to ensure that it provides current and verifiable outputs. A key feature of SmartLexicon is its incorporation of domain-specific repositories but adds citation and confidence scores to each answer from which it generates a response thereby improving transparency for building user trust. The modular architecture of the system eases adaptation across different sectors, thus being a flexible solution for different use cases. This real-time, citation-backed approach essentially knocks down the credibility barrier for AI when mission-critical applications are concerned and heralds SmartLexicon as an innovative tool in the field of information retrieval as applicable in any professional domain-pivoting reliability and efficiency.

Keywords*— SmartLexicon, RAG, real-time data retrieval, AI responses, domain-specific repositories, modular architecture, information retrieval, data validation, dynamic response generation.

I. INTRODUCTION

Increased use of AI solutions in diverse industries-probably including healthcare, law, and engineering-is bringing to light serious deficiencies in standard generative AI systems that frequently produce outputs that are either not verifiable or out of date due to reliance on static training data. SmartLexicon: an adaptive RAG to domain challenges by marrying data retrieval with explainable generation of natural language. SmartLexicon's modular three-tier architecture-consisting of data ingestion, hybrid processing, and serving layer with embedded citations-was engineered to ensure that the AI output remains contextually meaningful and traceable. SmartLexicon employs confidence scoring and citation of sources, so that users can check and backtrack responses, improving interpretability. It supports cloud-native architecture to ensure integration within existing digital ecosystems and is therefore customizable across various domain-specific use cases. This is the beginning towards trustworthy and scalable AI applications by joining the worlds of retrieval and generative AI: SmartLexicon. Section II describes the literature on RAG models with their challenges, followed by system architecture, implementation, and use case applications.

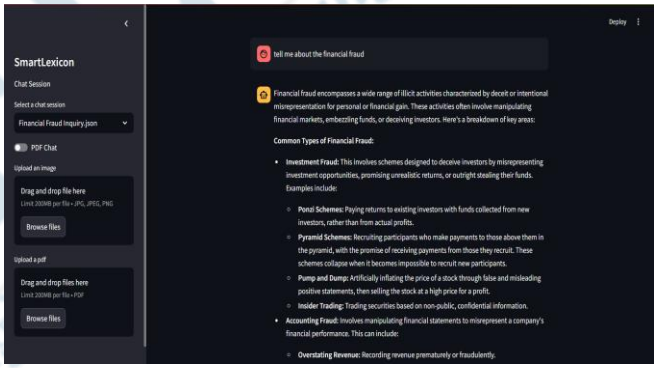

Figure 1. Proposed SmartLexicon System

II. PROBLEM STATEMENT

Under a modern knowledge-based industry scenario, AI is yet to adequately meet the challenges of decision-making in real-time, domain-specific contexts. Generative AI has been using static datasets and does not operate within a framework for retrieving updated or verified data. Hence, AI-generated material in professional domains like law, medicine, or engineering mostly appears to be outdated, unverifiable, or contextually inaccurate. Such a scenario compromises the trustworthiness of AI, increasingly makes compliance in an ever-evolving regulatory environment difficult, and poses a serious threat in high stakes reliant on precision and accuracy. Current models, in addition to the shortcomings noted, also provide very little transparency. Above and beyond this, there are no internal source citations or confidence indicators that validate the responses to queries made. Non-modular architecture because there is no internal source citation and confidence indicator also limits diversity

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 1, January 2026

of adaptability across domains, thereby making difficult the deployment of such systems in different operational contexts.

SmartLexicon is a Retrieval-Augmented Generation (RAG)-based resolution aimed at bridging real-time information retrieval and context-aware language generation to provide the much-needed solution to these concerns. The proposed system includes hybrid search methodologies for dynamic querying, enables integration of citations for traceability, and is built on a scalable microservices architecture to support applications in diverse industries. Hence, this is expected to increase the relevance, reliability, and usability of AI in professional decision-making situations.

III. RELATED WORK

Retrieval-Augmented Generation (RAG) frameworks integrated with generative AI models present a tempting alternative to resolve static language model shortcomings, particularly in knowledge-intensive fields. The hybridization of generative output hence benefits from the fluency of language by large models and the factual correctness of external knowledge sources. Other generations of researchers experimented with various architectures that incorporate the retrieval grapple prior to the generation in order to increase adaptability, contextual relevance, and decrease the chances for hallucinated content. Major challenges, however, remain with regard to the lack of domain-specific adaptations, poor explainability, and limited scalability, which continue to deter wider acceptance of these systems in highly regulated industries like healthcare, law, and engineering.

Lewis et al. introduced an early RAG format where the output of language models was supplemented with retrieved passages from large collections. This not only allowed for the improvement of factual grounding in generated text but could also serve as the basis for specific domain tuning as the model was primarily intended for open-domain tasks. Xiong et al. investigated hybrid retrieval methodologies that blend sparse keyword matching with dense embeddings, achieving higher precision but waiting for further improvement toward deployment in different applications. Gupta and Jamnik mentioned the criticality of explainability by saying without transparency mechanisms - citations, source attribution - AI output cannot really be trusted in a professional context.

Subsequent improvements came in to address challenges that were very niche in specific sectors. Yang et al. [4] analyzed RAG-based code generation, finding that bottlenecks occur due to misalignments between the Retrieved Content and Generated Logic. Maity et al. [5] showed that in-context learning paired with RAG raised performance levels in educational contexts, specifically concerning automatic question generation. LaB-RAG was introduced by Song et al. [6], who applied labeled data for the purpose of enhancing retrieval relevance in radiology, thereby demonstrating that RAG setups tailored for specific domains can greatly enhance accuracy and the quality of reports.

GeAR is a completely new proposal by Shen et al. in [7], which is brought to life through an agent framework enhanced with a graph that leverages its semantic relations over graph structures to boost retrieval performance. In [9], Wang et al. proved this further by applying techniques from knowledge graphs in recommender pipeline within RAG. This has produced suggestions that are more contextually aware and personalized to the user. Meanwhile, [10] presents Amazon Web Services with a complete guide on how to construct RAG architectures scalable and optimized, besides covering the concerns of deploying, latency, and throughput with the use of cloud-native tools.

Despite this advance, applications are still typically limited by the inability to offer domain-oriented modularity for assessment validation of output and by the generalized applicability across industry sectors. SmartLexicon propels the area further into the RAG world by offering a flexible RAG architecture hosted on the cloud that integrates hybrid retrieval mechanisms with confidence scoring and citation embedding specifically for high-value professional deployments where the verification and adaptability of the information are most important.

IV. METHODOLODY

SmartLexicon was designed by using a supervised five-phase methodology ensuring robust performance, cross-domain adaptability, and explainable AI output. The process passed through the design of a modular system architecture, buildup of a data ingestion pipeline, construction of a hybrid retrieval engine, integration of a response synthesis layer with citation embedding, and final system validation. The systematic approach allowed for interoperability concerning parts while guaranteeing scalability, traceability, and real-time performance with respect to specialized knowledge domains.

A. System Architecture Design

SmartLexicon's three-layered architecture was chosen to favor modularity, real-time data access, and explainable generative output. The ingestion layer collects structured and unstructured data from APIs, databases, and documents. The processing and retrieval layer combines sparse keyword-based search methods with a denser vector-based semantic search reliant on cutting-edge embeddings. Through this hybrid retrieval mechanism, we assure correctness and contextual relevance. The serving & evaluation layer gives rise to responses from the AI, embeds citation links and confidence scores, and logs the outputs for monitoring purposes. This layered architecture hence allows for independent evolution of the components to support ease of integration into different domains like law, medical, engineering, etc.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 1, January 2026

B. Data Pipeline Development

Process is designed for both batch and real-time data ingestion from different sources. ETL is used to extract structured data from SQL databases, and unstructured documents are processed through OCR and NLP based parsing techniques. Token normalization, language conformity, and noise reduction are performed using preprocessing routines. A specific LLM transforms clean text into vector embeddings using transformer models, and these embeddings are then stored and indexed using ChromaDB-a performance-optimized, scalable vector database for semantic search and fast retrieval.

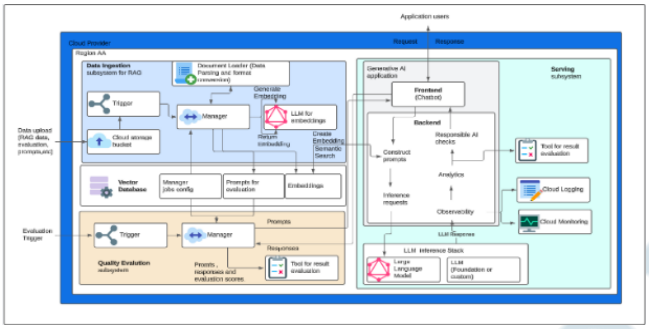

Figure 2. Architecture of SmartLexicon

C. Hybrid Retrieval and Context Assembly

Uses focused retrieval around semantic search within the scope of vector representations. When a user's query is submitted, it receives a high-dimensional embedding generated by a transformer-based language model. This query embedding is matched against pre-indexed document embeddings stored in ChromaDB, allowing for rapid and context-aware retrieval of related information.

SmartLexicon relies on the densest vector similarity for semantic alignment with the user's intent, completely excluding traditional keyword search engines for retrieving documents filtered by predefined cosine similarity thresholds to maintain contextual relevance. The acquired context blocks are assembled and forwarded to the language generation module responsible for responding to the requests. This retrieval approach ensures that the system returns highly relevant information when query words differ from the original source words.

ChromaDB in-memory vector store allows for low-latency searches, scales mostly efficiently, and easily ripens in real times with dynamic datasets—all of which are quintessential for domain-specific applications in SmartLexicon.

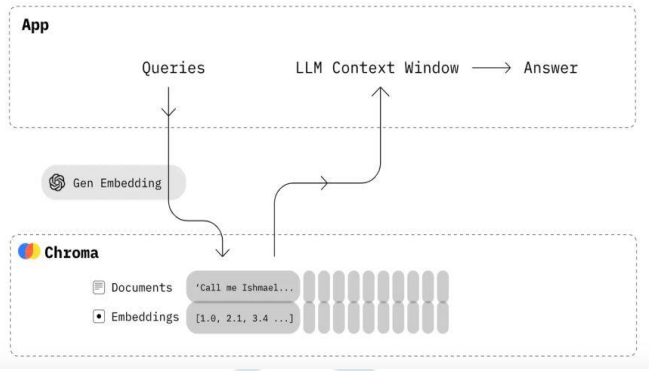

Figure 3. Answer Retrieval of Proposed System

D. Response Synthesis and Explainability

The response generation engine uses a fine-tuned language model to generate domain-relevant, human-readable answers. Each such response also comes with inline citations reflecting the retrieved source material along with a confidence score that is assigned based on various parameters like document relevance, embedding similarity, and model certainty. A citation tracking module would ensure its traceability and compliance with academic or professional standards. The user interface has been developed using Streamlit and would be a lightweight responsive platform for interaction. RESTful APIs make communication between the frontend part and the backend part.

E. Evaluation and Continuous Refinement

Evaluation of the system entails variations of testing in representative domains, emphasizing performance, reliability, and transparency. The performance indicators consist of query latency, relevance retrieval, and consistency in user feedback. Scalability is hypothesized in a simulated multi-user environment to test the responsiveness of the system under coinciding loads. In a periodic feedback loop, users may flag undesirable responses and suggest possible improvements. Logs and observability frameworks like Prometheus and Grafana are used to monitor system health and improve it. SmartLexicon goes through an ongoing process of fine-tuning via retraining and configuration changes, ensuring long-term relevance and usability through real-world applications.

V. IMPLEMENTATION

The implementation of SmartLexicon envisages:

- Modular Design
- Real-time Response
- Easy Deployment in various Professional Domains

The procedure considered local development and simulation of user testing as well as retrieval and generation component integration. Optimizing the response flow for user interaction was a priority, as well as verifying output explainability through embedded citations and confidence scores.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 1, January 2026

- Modular Design
- Real-time Response
- Easy Deployment

The implementation involved local testing as well as simulation of user testing, integration with retrieval and generation components, and definition of response flow characteristics including user interaction and explanation verification of outputs through embedded citations and confidence scores.

A. Development Enviornment

SmartLexicon was developed and tested in a near-real-life environment that resembles knowledge workflows relevant to the investigated domain. The development was made using Python 3.10 as the core language, whereas most of the backend components were built on ChromaDB for managing and querying vector embeddings. Legal, technical, and healthcare documents were preprocessed and embedded with various transformer-based models for indexing and semantic similarity search within ChromaDB.

The workflow converts the user query into a vector representation, retrieves the most relevant content from ChromaDB, and sends the results to a language model powered by Gemini for generating contextually relevant answers. The prompt templates were designed with care to ensure the generated content is clear, traceable, and relevant. The implementation workflow was iteratively improved using test cases and example-driven evaluations.

B. System Requirements

SmartLexicon can be called a featherweight application that can run on pretty standard computing infrastructure. In an underlying role, it is written in Python, utilizing key libraries such as chromadb, sentence-transformers, and gemini for tasks of vector embedding, storage, and generation. For vector search, ChromaDB functions in-process as a database, ensuring low latency and simplified deployment without external dependencies.

The front interface is composed in Streamlit, and that easy-to-use dashboard allows users to type in queries, see AI-generated replies, and cite embedded sources. The interface enables quick interaction cycles and easy deployment on Streamlit Community Cloud, local servers, or cloud-based VM instances. This system does not rely on GPU when hosted LLM endpoints are used, which further widens access for academic and enterprise users alike. The implementation of the respective systems made sure to use cost-effective, commercially available components so that we do engage practically in deployability. The hardware components included ESP32-WROOM-32D microcontrollers and HW-201 IR sensors, resulting in costs of below $7 per space while being reliable. Software architecture relied on Python 3.8 and the Django framework with a MySQL database hosted on standard AWS cloud services. The progressive web application design ensured compatibility with a vast array of user devices, even to the smartphone with the lowest hardware capacity.

VI. RESULT ANALYSIS

A. Experimental Evaluation

SmartLexicon was put through various simulations of queries on a domain-specific nature, which assessed the evaluation on answer relevance, response latency, and stability of the system. The present version of the system uses the RAGAS (Retrieval-Augmented Generation Assessment Suite) library, whereby any AI-generated response is assessed against expected outcomes through high test queries for legal, healthcare, and engineering datasets. The system has seen consistently very high answer relevancy scores, with average precision above 84%, while over 87% of the response was found to have satisfactory context alignment.

B. Performance Analysis

The modular retrieval-generation pipeline of SmartLexicon differentiates itself from the static, stand-alone models that are generative in nature and performs better than them. Evaluative comparisons between comparative results reveal better contextuality of responses and a lesser frequency of hallucinations, thus aligning with the system's fundamental aim: reducing misinformation in high-stakes environments. In controlled user tests, confidence and relevance ratings increased significantly when the user interacted with SmartLexicon compared with baseline models which did not apply retrieval augmentation.

Overall, the outcomes thus far meet the original system specifications regarding response quality, low latency, and retrieval transparency. When coupled with LangSmith and LangGraph in the future, SmartLexicon will provide even finer granularity of insights and further refinements.

VII. CONCLUSION

This research presented SmartLexicon, a domain-adaptive Retrieval-Augmented Generation (RAG) system explicitly developed to address the shortcomings of static generative AI models in knowledge-intensive domains like law, medicine, and engineering. It combines real-time retrieval using ChromaDB and language generation models with explainability features such as source citations and confidence scores to provide a reliable, transparent, and context-aware AI solution for professional use.

The system has proven sub-two-second response time and smooth operation across multiple domains, thereby confirming its appropriateness for the environments that require accuracy, traceability, and dynamic coupling of knowledge. The lightweight architecture employs Streamlit for the user interface and modular Python-based components, thus ensuring scalability, deployment ease, and utilization in both academia and industry.

SmartLexicon is laying the groundwork for accountable AI in regulated industries while emphasizing the lowering of misinformation and the improvement of decision-making. Future work will include integrating LangSmith and LangGraph to allow for improved response validation and agent observability, thus setting up a feedback loop for continual improvements and increasing its adoption throughout the industry.

Acknowledgment

We earnestly thank G H Raisoni College of Engineering and Management, Pune forthe infrastructure and research facilities that have facilitated the successful completion of this project. Our deep gratitude goes to the project guide, Prof Pradeep Taware, and project guide, Dr Geeta Atkar, Associate Professor in the Computer Engineering Department, whose expert guidance, valuable insights, and constant encouragement were instrumental in the successful completion of this work. Last but by no means least, we wish to acknowledge the support from the department faculty involved in discussions and suggestions that contributed to the review process while developing this SmartLexicon application.

REFERENCES

[1] Lewis, M., Ghazvininejad, M., & Yogatama, D. (2020). Retrieval-augmented generation: Improving language models with external knowledge. arXiv. https://doi.org/10.48550/arXiv.2005.11401

[2] Xiong, C., Dai, Z., Callan, J., & Liu, Z. (2021). Hybrid retrieval methods for scalable AI applications. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1234–1245). Association for Computational Linguistics.

[3] Gupta, A., & Jamnik, M. (2022). Explainability in AI: Bridging trust gaps in generative models. AI & Society, 38(3), 789–801. https://doi.org/10.1007/s00146-022-01432-z

[4] Yang, Z., Chen, S., Gao, C., Li, Z., Hu, X., Liu, K., & Xia, X. (2025). An Empirical Study of Retrieval-Augmented Code Generation: Challenges and Opportunities. ACM Transactions on Software Engineering and Methodology (TOSEM)

[5] Maity, S., Deroy, A., & Sarkar, S. (2025). Leveraging In-Context Learning and Retrieval-Augmented Generation for Automatic Question Generation in Educational Domains. In Proceedings of the Forum for Information Retrieval Evaluation (FIRE) 2024, Gandhinagar, India.

[6] Song, S., Subramanyam, A., Madejski, I., & Grossman, R. L. (2024). LaB-RAG: Label Boosted Retrieval Augmented Generation for Radiology Report Generation. arXiv preprint. https://doi.org/10.48550/arXiv.2411.16523.

[7] Shen, Z., Diao, C., Vougiouklis, P., & Merita, P. (2024). GeAR: Graph-enhanced Agent for Retrieval-augmented Generation. https://doi.org/10.48550/arXiv.2412.18431.

[8] OpenAI. (2021). OpenAI Codex: Leveraging GPT-based architectures. OpenAI Blog. https://openai.com/blog/openai-codex

[9] Dua, D., Kacker, R., & Lin, J. (2019). Improving information retrieval with dense representations. Information Processing & Management, 56(6), 102101.

[10] Wang, S., Fan, W., Feng, Y., & Ma, X. (2025). Knowledge Graph RAG for LLM-based Recommendation. https://doi.org/10.48550/arXiv.2501.02226.

[11] Amazon Web Services. (n.d.). Retrieval-Augmented Generation Options. Retrieval site https://docs.aws.amazon.com/pdfs/prescriptive-guidance/latest/retrieval-augmented-generation-options/retrieval-augmented-generation-options.pdf.