Intelligent Human Systems Integration (IHSI), Vol. 200, 2026, 63–73
https://doi.org/10.54941/ahfe1007065

Conceptual Framework for Designing Domain-Specific LLM-Based Information Systems

Dominik Ullrich, Jens Wallys, and Sven Hinrichsen
Ostwestfalen-Lippe University of Applied Sciences and Arts, Lemgo, 32657, Germany

ABSTRACT

Retrieval-augmented generation (RAG) based on large language models (LLMs) has established itself as a key technology for combining domain-specific information with generative language skills, thereby providing transparent, up-to-date information. Many firms are already piloting such LLM-based information systems, but report a high degree of complexity in planning and implementation. A generally accepted regulatory framework that consistently maps key decisions is not yet available to companies. This article therefore presents a multi-level system that organizes design decisions throughout the configuration process. This framework is intended to support users in the planning, realizing, evaluation, and further development of an LLM-based information system. To achieve this goal, a qualitative-empirical research design was chosen. First, publications from the period 2022 to 2025 were identified and selected using a systematic literature search in accordance with the PRISMA guideline. The selected publications were then evaluated using a qualitative content analysis. The result is a system that was reviewed, revised and finalized at an expert workshop.

Keywords: LLM-based information systems, System configuration, Retrieval-augmented generation, Large language model

INTRODUCTION

Large language models (LLMs) represent an important milestone in digital transformation (Arslan et al., 2024). In the field of automated text generation, they facilitate the creation of targeted and consistent texts (Liang et al., 2024). They can also be used for content summarization by capturing relationships between documents and creating coherent syntheses (Godbole et al., 2024). In addition, LLMs form the basis for AI information systems, such as chatbots that provide expertise and enable user-friendly interactions (Zheng et al., 2023). These systems can also be referred to as assistance systems, as they support humans in performing individual tasks. In the fields of business and administration, LLMs are increasingly transitioning from pilot phase to real-world operation, with more and more investment going into the further development of such LLM-based information systems. Significant challenges include data protection, security, budget and integration (Newswire 2025). It is generally assumed that the implementation of context-adaptive information systems will lead to a significant increase in productivity. However, this increase depends not only on the technical capabilities of the LLM, but

Received September 11, 2025; Revised November 10, 2025; Accepted November 25, 2025; Available online February 1, 2026

also on the human-centered design of the information system (Weber et al., 2024).

Retrieval-Augmented Generation (RAG) enhances the generative capabilities of an LLM by providing it with additional domain-specific information. The key advantage of this approach is that it allows the use of organization-specific, sensitive data without having to retrain the LLM itself (Klesel & Wittmann, 2025). Furthermore, the use of RAG reduces hallucinations and increases traceability, timeliness, and relevance (Sefton, 2025).

A fundamental challenge is emerging in the further development of LLMs. The larger, more complex and more universally applicable LLMs become, the higher the costs and development effort (Stoica et al., 2024). Against this backdrop, a modular structure that allows important features to be prioritized is profiting, but so far there is a lack of systematic approaches and standards to ensure reproducible results (Wang et al., 2024). In addition, the multitude of technical components and their complex interactions make it difficult to design LLM-based information systems in a targeted manner (Shah, 2024). Furthermore, scientific literature has identified a lack of overarching structure in the investigation of individual factors (Lambiase et al., 2025).

Against this background, the aim of this article is to develop a system for configuring LLM-based information systems. This system should map central components, characteristics and thus design options, thereby supporting the process of designing LLM-based information systems.

Method

To achieve this goal, a three-step approach was chosen. In the first step, a systematic literature review was conducted in accordance with the PRISMA guidelines (Page et al., 2021). The databases IEEE Xplore, Springer Link and Google Scholar were used. The search string combined the terms "large language model", "configuration", "information system", and "retrieval-augmented generation". Due to the topicality of the subject, the time period was limited from 2023 to the present (May 2025). Duplicates were first removed from the total of 452 hits. The titles and abstracts were then screened to exclude irrelevant publications. The full texts of the remaining articles were reviewed based on previously defined inclusion and exclusion criteria. Only scientific publications directly related to LLM configuration, architecture, or components were considered. As a result, 75 sources were included in the evaluation. In a second step, a qualitative content analysis was performed. The category system was derived deductively from various existing subsystems on the one hand and completed inductively from the material on the other. Rather than taking a sequential approach to analyzing the publications, a procedure was chosen that involved specifically searching and evaluating the literature for individual content or categories. In the third step, the category system was reviewed in a workshop with five

experts, particularly with regard to the criteria of completeness, clarity, and practicality.

RESULTS

This method resulted in a category system comprising seven top-level categories: 1. System, 2. System context, 3. LLM configuration, 4. RAG configuration, 5. Prompt engineering, 6. Data provision and security measures, and 7. Evaluation. There are subcategories at a secondary level within these seven categories. Some of these subcategories can be subdivided again.

The **system** category includes the fundamental decisions regarding the architecture of the LLM-based information system (see Fig. 1). A fundamental decision concerns whether to use a proprietary or open-source LLM. In this case, criteria such as performance (Zolfaghari et al., 2024), costs and support, data sovereignty and adaptability must be taken into account when making this decision (Kumar et al., 2025). Another system decision involves the question of whether to use LLM frameworks or end-to-end platforms. LLM frameworks such as LangChain, LlamaIndex, or Haystack are used for modular integration, retrieval, agent logic, and orchestration (Gao et al., 2023; Kumar et al., 2025). End-to-end platforms are used when development, deployment, monitoring, and governance are to be bundled in an integrated environment (Pahune & Akhtar 2025).

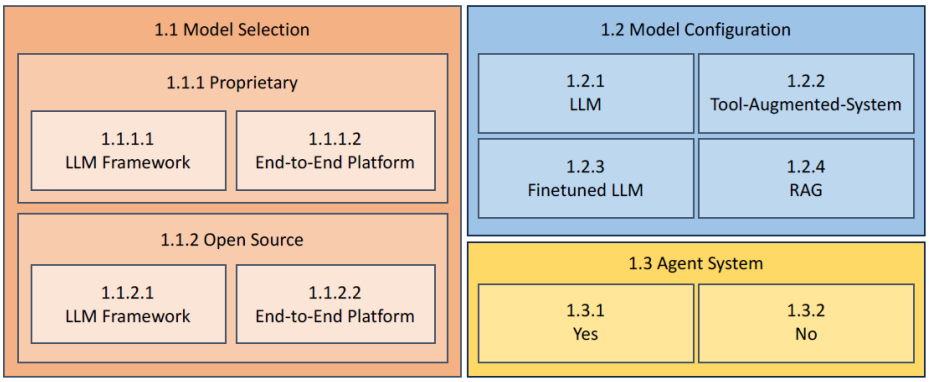

Figure 1: Structure and contents of the system category.

Another system decision relates to model configuration, for which there are four options: a generally trained LLM (without any extension) for elementary Q&A tasks (Kotkar et al., 2024), tool-augmented systems (extension of the LLM with functions) for function calls and external queries (Hou et al., 2025; Yager 2024), fine-tuned LLM for specialized tasks (Shukla & Parker 2024) and RAG for generating answers based on connected domain-specific databases (Gao et al., 2023). Optionally, one or more agent systems can also be integrated into the information system (Aquino et al., 2025).

The second category, referred to as **system context**, describes the content and technical context in which the LLM-based information system is to be used. The following subcategories can be distinguished: 2.1 Domain, 2.2 Task type, and 2.3 Type of provision. The domain describes the context of the information that an LLM-based system is to provide. A systematic literature review identified a wide variety of areas of application. These relate, for example, to the domains of medicine, law, finance, manufacturing, science and education.

In case of LLM-based information systems, a basic distinction can be made between the following two types of tasks: 2.2.1 Question-answering (Liu et al., 2025) and 2.2.2 Dialogue/chatbot (Nsaif et al., 2024). While question-and-answer systems are designed to answer questions briefly and precisely, dialogue systems focus on ongoing interaction. The type of provision involves the question of which technical infrastructure should be used to implement an LLM-based information system. There are two options: 2.3.1 Deployment via cloud platforms or API access to corresponding services (Taulli & Deshmukh 2025), 2.3.2 On-premises deployment for maximum control and data protection (Paloniemi et al., 2025).

The third category concerns **LLM configuration**. A distinction is made between the following two subcategories: 3.1 parameter adjustment and 3.2 model adjustment. Parameter adjustment involves setting various decoding parameters. These include temperature (Ahmed et al., 2025; Gadiraju et al., 2024), top p (Ahmed et al., 2025; Ruman, 2024), and top k (Ruman, 2024), control variables such as frequency penalty and presence penalty (Prabhugaonkar 2024), and maximum tokens (Prabhugaonkar, 2024). The goal is to achieve a desired balance of predictability, coherence, diversity, length, and latency. Higher temperature and lower top k increase variability, stricter penalties reduce repetitions, and maximum tokens limit response length and costs (Ruman, 2024).

Model adaptation involves the targeted further development and fine-tuning of pre-trained LLMs for specific tasks or contexts. Instruction tuning is used to strengthen instruction compliance and task generalization (Bergmann 2025; Feng et al., 2024; Yan et al., 2025). Fine-tuning is used when an LLM needs to be retrained with domain-specific knowledge (El Hassani et al., 2025). Adapter-based fine-tuning, such as LoRA adapters, is used when efficiency, rapid iteration, and limited resources are paramount, as only small additional modules are optimized (Feng et al., 2024). Joint training links the generator and retriever when RAG is used, so that retrieval and generation are coordinated (Fan et al., 2024; Gao et al., 2023). Feedback mechanisms such as Reinforcement Learning from Human Feedback (RLHF) are provided to align responses with preferences, quality standards, and desired behavior (Gao et al., 2023; Yan et al., 2025). Retriever fine-tuning is used to increase semantic consistency between query and context and prevent hallucinations without completely retraining the base model (Béchard & Ayala 2025; Gao et al., 2023).

The fourth category includes **RAG configuration**. This consists of the following subcategories: 4.1 Indexing, 4.2 Retrieval, and 4.3 Post-Retrieval

Processing. The purpose of Indexing (4.1) is to prepare documents or data for quick and efficient searching. The indexing process can be broken down as follows: 4.1.1 Chunking: Dividing documents into small segments; 4.1.2 Embedding: Each chunk is converted into a high dimensional vector; 4.1.3 Storage in a vector database: The vectors are stored in a vector database, which enables fast similarity searches. Various chunking methods (4.1.1) can be distinguished (fixed size, sentence, paragraph, sliding window, recursive split, and hierarchical chunking). Smaller chunks increase hit accuracy, while larger chunks preserve more context (Wang et al., 2025). The content is represented using embeddings (4.1.2). There are three types of embeddings: sparse, dense, and hybrid (Ahluwalia & Wani 2024). Vector databases (4.1.3) such as FAISS, Milvus, Pinecone, Weaviate, ChromaDB, Annoy, and Elasticsearch are used to store the vectors (Ma et al., 2023; Wang et al., 2025).

Retrieval (4.2) refers to the process of retrieving information from a data source. The following types of retrieval (4.2.1) can be distinguished: sparse, dense, hybrid, and generative retrieval (Ma et al., 2023; Wang et al., 2025). Context retrieval (4.2.2) is a special retrieval approach in the RAG environment that takes into account the context of the query or the ongoing conversation instead of considering a single query in isolation. Singular retrieval provides context once, iterative retrieval enriches it step by step, recursive retrieval refines searches via feedback, and adaptive retrieval controls retrievals depending on need (Gao et al., 2023). Prompt integration and retrieval granularity determine how much context is added and at what level (Fan et al., 2024). Query optimization (4.2.3) involves transforming unclear user queries into more precise, semantically accurate ones to improve retrieval results. Methods such as query expansion, subquery, query rewrite, query routing, and chain of verification are used for query optimization (Gao et al., 2023).

Post-retrieval processing (4.3) encompasses all processing steps performed after a RAG's primary function has been fulfilled, but before information is transferred to the LLM or user. This process is also intended to contribute to high-quality information output. Important process steps include reranking and filtering. Reranking reorders the RAG results according to their relevance, while filters remove irrelevant content (Gao et al., 2023; Feng et al., 2024; Choi et al., 2025).

The fifth category deals with **prompt engineering**. A distinction must be made between prompt engineering techniques (5.1) and prompt templates (5.2). Important prompt engineering techniques include zero-shot (5.1.1), few-shot (5.1.2), chain-of-thought (5.1.3), ReAct (5.1.4), prompt chaining (5.1.5), self-consistency (5.1.6), step-back (5.1.7), and meta-prompting (5.1.8) (Schulhoff et al., 2024; Wang et al., 2025). Prompt templates (5.2) are structured templates for prompts with placeholders that allow dynamic content to be inserted (e.g., entering an article number for which information is to be retrieved) in order to obtain consistent responses from LLMs. Using these templates standardizes inputs and thus supports the output of consistent responses (Mao et al., 2025).

The sixth category includes **data provision** (6.1) and **security measures** (6.2). In the case of data provision (6.1), a distinction can be made between unstructured, semi-structured, and structured data. Unstructured data includes text, images, or other media without a schema. This makes automated processing of this data considerably more difficult (Gao et al., 2023; Zhou et al., 2025). Its integration into LLM-based systems requires specialized tools and complex preprocessing, combined with high computational effort (Liu et al., 2025; Mahadevkar et al., 2024; Rani et al., 2024). Semi-structured data, such as XML or JSON, has structural features, but does not adhere to a fixed schema, thus posing special challenges for processing by LLMs (Gao et al., 2023; Zhou et al., 2025). Particularly challenging are inconsistently formatted tables, which require the use of specialized tools such as the LangChain Text Splitter (Olawore et al., 2025). Structured data, such as in the form of knowledge graphs, is available in a schema (Gao et al., 2023). It can be easily processed by the RAG system and tends to improve the relevance and traceability of results (Rani et al., 2024).

In the planning of LLM-based information systems, security measures and protection mechanisms (6.2) must be taken into account in order to protect sensitive data and prevent manipulation of the system. Various measures can be implemented. Input filters can be used to intercept prompt injection and jailbreaks before generation. Output filters can be used to check generated responses for sensitive content, personal data, and risky instructions. Monitoring and traceability can be established to log interactions and detect conspicuous patterns. Access controls can be applied to manage authentication and authorization (Aquino et al., 2025).

The **evaluation of the performance** of LLM-based information systems, the seventh category of the classification system, involves the systematic evaluation of all central components, including indexing, retrieval, and response generation. Such an evaluation requires the application of specialized methods and tools that enable the comparison of different system configurations. A distinction is made between retrieval quality (7.1), generation quality (7.2), and other tools (7.3). The evaluation can be based on standardized question-answer datasets and take into account indexing, retrieval and response generation together, thus enabling a holistic view of the system (Brehme et al., 2025).

For retrieval quality (7.1), recall and precision can be used to measure completeness and accuracy (Chen, L.-C. et al., 2025; Jung et al., 2025). Hit Rate@k can be used to check for the presence of at least one relevant context among the top k (Khan et al., 2025). Mean Reciprocal Rank and nDCG can be used to map the rank position and sorting quality (Chen, L.-C. et al., 2025; Oro et al., 2025). Cosine Similarity is a metric for measuring the semantic proximity between vector representations of texts (Chondamrongkul et al., 2025; Gadiraju et al., 2024).

Evaluating generation quality (7.2) requires metrics that go beyond measuring formal correctness (Yu et al., 2025). Exact Match checks for complete text similarity in a binary manner and averages the individual results to produce an overall score between 0 and 1 (Yan et al., 2025). The F1 score

combines precision and recall as a harmonic mean and evaluates word overlap in a balanced way for tasks such as classification and answer comparison (Liu et al., 2025; Yan et al., 2025). BLEU measures how much consecutive word sequences in the generated text overlap with those in the reference text. The greater this overlap, the higher the BLEU score (Choi et al., 2025; Liu et al., 2025). ROUGE is an evaluation method that measures the overlap of text segments between generated and reference responses (Yan et al., 2025). Accuracy measures the agreement with ground truth, often using a threshold on cosine similarity or natural language inference, and returns a value between 0 and 1 (Gadiraju et al., 2024; Jung et al., 2025; Yan et al., 2025). BERTScore measures the semantic similarity between model output and reference using context-dependent token representations from pre-trained BERT models (Ahmed et al., 2025; Liu et al., 2025).

Other evaluation tools (7.3) include tools specifically designed to evaluate RAG systems. For example, RAGAS (Es et al., 2023) evaluates retrieval and generation multidimensionally and uses a second LLM for criteria such as relevance, factual accuracy, contextual accuracy, response similarity, and correctness (Olawore et al., 2025; Oro et al., 2025). ARES (Saad-Falcon et al., 2023) automates evaluation based on the criteria of contextual relevance, answer fidelity, and answer relevance, and reduces the administrative annotation effort by using synthetically generated test data (Olawore et al., 2025; Oro et al., 2025).

DISCUSSION

Many companies are involved in the development and implementation of LLM-based information systems. This requires a large number of decisions to be made. The presented system is intended to help developers and decision-makers by showing them options and possibilities in the design of LLM-based information systems. It should be noted that individual methods and technologies are developing dynamically to a high degree, so that even a regulatory framework such as this must be adapted regularly.

ACKNOWLEDGMENT

This article was created as part of the KIPRO project. The project is funded by the Bundeswehr Centre for Digitalization and Technology Research (dtec.bw) with grants from the European Union (NextGenerationEU).

REFERENCES

Ahluwalia, A. & Wani, S. (2024). Leveraging Large Language Models for Web Scraping. DOI: https://doi.org/10.48550/arXiv.2406.08246.

Ahmed, B. S., Baader, L. O., Bayram, F., Jagstedt, S. & Magnusson, P. (2025). Quality Assurance for LLM-RAG Systems: Empirical Insights from Tourism Application Testing, 200–207. DOI: https://doi.org/10.48550/arXiv.2502.05782.

Aquino, G. d. A. e., Da Azevedo, N. S. d., Okimoto, L. Y. S., Camelo, L. Y. S., Bragança, H. L. d. S., Fernandes, R., Printes, A., Cardoso, F., Gomes, R. & Torné, I. G. (2025). From RAG to Multi-Agent Systems: A Survey of Modern Approaches in LLM Development, MDPI AG. DOI: https://doi.org/10.20944/preprints202502.0406.v1.

Arslan, M., Ghanem, H., Munawar, S. & Cruz, C. (2024). A Survey on RAG with LLMs, Procedia Computer Science, 246, 3781–3790. DOI: https://doi.org/10.1016/j.procs.2024.09.178.

Bergmann, D. (2025). What Is Instruction Tuning? | IBM. https://www.ibm.com/think/topics/instruction-tuning (Accessed on July 27, 2025).

Brehme, L., Ströhle, T. & Breu, R. (2025). Can LLMs Be Trusted for Evaluating RAG Systems? A Survey of Methods and Datasets. DOI: https://doi.org/10.48550/arXiv.2504.20119.

Béchard, P. & Ayala, O. M. (2025). Multi-task retriever fine-tuning for domain-specific and efficient RAG. DOI: https://doi.org/10.48550/arXiv.2501.04652.

Chen, L.-C., Pardeshi, M. S., Liao, Y.-X. & Pai, K.-C. (2025). Application of retrieval-augmented generation for interactive industrial knowledge management via a large language model, Computer Standards & Interfaces, 94. DOI: https://doi.org/10.1016/j.csi.2025.103995.

Choi, Y., Kim, S., Bassole, Y. C. F. & Sung, Y. (2025). Enhanced Retrieval-Augmented Generation Using Low-Rank Adaptation, Applied Sciences, 15(8), 4425. DOI: https://doi.org/10.3390/app15084425.

Chondamrongkul, N., Hristov, G. & Temdee, P. (2025). Addressing Technical Challenges in Large Language Model-Driven Educational Software System, IEEE Access, 13, 12846–12858. DOI: https://doi.org/10.1109/ACCESS.2025.3531380.

El Hassani, I., Masrour, T., Kourouma, N. & Tavčar, J. (2025). AI-driven FMEA: integration of large language models for faster and more accurate risk analysis, Design Science, 11. DOI: http://doi.org/10.1017/dsj.2025.7.

Es, S., James, J., Espinosa-Anke, L. & Schockaert, S. (2023). RAGAS: Automated Evaluation of Retrieval Augmented Generation. ArXiv: https://arxiv.org/abs/2309.15217.

Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S. & Li, Q. (2024). A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models, 6491–6501. ArXiv: https://arxiv.org/abs/2405.06211.

Feng, K., Luo, L., Xia, Y., Luo, B., He, X., Li, K., Zha, Z., Xu, B. & Peng, K. (2024). Optimizing Microservice Deployment in Edge Computing with LLMs: Integrating RAG and Chain-of-Thought Techniques, Symmetry, 16(11), 1470. DOI: https://doi.org/10.3390/sym16111470.

Gadiraju, S. S., Liao, D., Kudupudi, A., Kasula, S. & Chalasani, C. (2024). InfoTech Assistant: A Multimodal Conversational Agent for InfoTechnology Web Portal Queries, IEEE BigData 2024, 3264–3272. ArXiv: https://arxiv.org/abs/2412.16412.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M. & Wang, H. (2023). Retrieval-Augmented Generation for Large Language Models: A Survey. ArXiv: https://arxiv.org/abs/2312.10997.

Godbole, A., George, J. G. & Shandilya, S. (2024). Leveraging Long-Context LLMs for Multi-Document Understanding and Summarization in Enterprise Applications. ArXiv: https://arxiv.org/abs/2409.18454.

Hou, X., Zhao, Y., Wang, S. & Wang, H. (2025). Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions. ArXiv: https://arxiv.org/abs/2503.23278.

Jung, J., Jeong, H. & Huh, E.-N. (2025). Federated Learning and RAG Integration: A Scalable Approach for Medical LLMs, ICAIIC 2025, 968–973, IEEE. ArXiv: https://arxiv.org/abs/2412.13720.

Khan, M. Z., Ge, Y., Mollel, M., McCann, J., Abbasi, Q. H. & Imran, M. (2025). RFSensingGPT: A Multi-Modal RAG-Enhanced Framework for Integrated Sensing and Communications Intelligence in 6G Networks, IEEE TCCN. DOI: https://doi.org/10.1109/TCCN.2025.3558069.

Klesel, M. & Wittmann, H. F. (2025). Retrieval-Augmented Generation (RAG), Business & Information Systems Engineering. DOI: https://doi.org/10.1007/s12599-025-00945-3.

Kotkar, A. D., Mahadik, R. S., More, P. G. & Thorat, S. A. (2024). Comparative Analysis of Transformer-based LLMs for Text Summarization, ACET 2024, IEEE, 1–7. DOI: https://doi.org/10.1109/ACET61898.2024.10730348.

Kumar, P., Haresh, M. & Hayagreevan, V. (2025). Development of Interactive Assistance for Academic Preparation Using Large Models Language, ICCCIT 2025, 265–269. DOI: https://doi.org/10.1109/ICCCIT62592.2025.10928137.

Lambiase, S., Catolino, G., Palomba, F., Ferrucci, F. & Russo, D. (2025). Exploring Individual Factors in the Adoption of LLMs for Specific Software Engineering Tasks. DOI: https://doi.org/10.48550/arXiv.2504.02553.

Liang, X., Wang, H., Wang, Y., Song, S., Yang, J., Niu, S., Hu, J., Liu, D., Yao, S., Xiong, F. & Li, Z. (2024). Controllable Text Generation for Large Language Models: A Survey. DOI: https://doi.org/10.48550/arXiv.2408.12599.

Liu, L., Zhou, Y., Ma, J., Zhang, Y. & He, L. (2025). Domain-Specific Question-Answering Systems: A Case Study of a Carbon Neutrality Knowledge Base, Sustainability, 17(5), 2192. DOI: https://doi.org/10.3390/su17052192.

Ma, L. et al., (2023). A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge. ArXiv: https://arxiv.org/abs/2310.11703.

Mahadevkar, S. V., Patil, S., Kotecha, K., Soong, L. W. & Choudhury, T. (2024). Exploring AI-driven approaches for unstructured document analysis and future horizons, Journal of Big Data, 11(1), 1–54. DOI: https://doi.org/10.1186/s40537-024-00948-z.

Mao, Y., He, J. & Chen, C. (2025). From Prompts to Templates: A Systematic Prompt Template Analysis for Real-world LLMapps, FSE 2025, 75–86. ArXiv: https://arxiv.org/abs/2504.02052.

Newswire, P. R. (2025). Study Finds 72 Percent of Enterprises Plan to Ramp Spending on GenAI in 2025, https://aithority.com/machine-learning/study-finds-72-percent-of-enterprises-plan-to-ramp-spending-on-genai-in-2025/ (Accessed on July 9, 2025).

Nsaif, W. S., Salih, H. M., Saleh, H. H. & Al-Nuaim, B. T. (2024). Conversational Agents: An Exploration into Chatbot Evolution, Architecture, and Important Techniques, The Eurasia Proceedings of STEM, 27, 246–262. DOI: https://doi.org/10.55549/epstem.1518795.

Olawore, K., McTear, M. & Bi, Y. (2025). Development and Evaluation of a University Chatbot Using Deep Learning: A RAG-Based Approach, 15545, 96–111. DOI: https://doi.org/10.1007/978-3-031-88045-2_7.

Oro, E., Granata, F. M. & Ruffolo, M. (2025). A Comprehensive Evaluation of Embedding Models and LLMs for IR and QA Across English and Italian, BDCC, 9(5), 141. DOI: https://doi.org/10.3390/bdcc9050141.

Page, M. J. et al., (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews, The BMJ Volume 372 No. 71. https://doi.org/10.1136/bmj.n71

Pahune, S. & Akhtar, Z. (2025). Transitioning from MLOps to LLMOps: Navigating the Unique Challenges of Large Language Models, Information, 16(2), 87. DOI: https://doi.org/10.3390/info16020087.

Paloniemi, T., Setälä, M. & Mikkonen, T. (2025). Porting an LLM based Application from ChatGPT to an On-Premise Environment, ICSR 2025, IEEE, 78–83. ArXiv: https://arxiv.org/abs/2504.07907.

Prabhugaonkar, S. (2024). Generative AI with Azure OpenAI – Part 2, https://cloud-authority.com/generative-ai-with-azure-openai-part-2 (Accessed on July 27, 2025).

Rani, M., Mishra, B. K., Thakker, D. & Khan, M. N. (2024). To Enhance Graph-Based RAG with Robust Retrieval Techniques, ICOSST 2024, IEEE, 1–6. DOI: https://doi.org/10.1109/ICOSST64562.2024.10871140.

Ruman (2024). Setting Top-K, Top-P and Temperature in LLMs, Medium, 3. April, https://rumn.medium.com/setting-top-k-top-p-and-temperature-in-llms-3da3a8f74832 (Accessed on June 25, 2025).

Saad-Falcon, J., Khattab, O., Potts, C. & Zaharia, M. (2023). ARES: An Automated Evaluation Framework for RAG Systems. ArXiv: https://arxiv.org/abs/2311.09476.

Schulhoff, S. et al., (2024). The Prompt Report: A Systematic Survey of Prompt Engineering Techniques, arXiv preprint 2406.06608. ArXiv: https://arxiv.org/abs/2406.06608.

Sefton, B. (2025). How Retrieval-Augmented Generation (RAG) Unlocks the Power of Business Data, Insightful AI, https://insightfulai.co.uk/how-retrieval-augmented-generation-rag-unlocks-the-power-of-business-data/ (Accessed on Juli 09, 2025).

Shah, D. (2024). Top Challenges in Building Enterprise LLM Applications, Coralogix, https://coralogix.com/ai-blog/top-challenges-in-building-enterprise-llm-applications/ (Accessed on August 05, 2025).

Shukla, V. & Parker, G. G. (2024). Building Custom LLMs for Industries: A Comparative Analysis of Fine-Tuning and RAG, ICAAEEI 2024, IEEE, 1–6. DOI: https://doi.org/10.1109/ICAAEEI63658.2024.10899129.

Stoica, I., Zaharia, M., Gonzalez, J., Goldberg, K., Sen, K., Zhang, H., Angelopoulos, A., Patil, S. G., Chen, L., Chiang, W.-L. & Davis, J. Q. (2024). Specifications: The missing link to making the development of LLM systems an engineering discipline. ArXiv: https://arxiv.org/abs/2412.05299.

Taulli, T. & Deshmukh, G. (2025). Developing Agents, in: Building Generative AI Agents: Using LangGraph, AutoGen, and CrewAI, Apress, 81–101. DOI: https://doi.org/10.1007/979-8-8688-1134-0.

Wang, H., Zhang, D., Li, J., Feng, Z. & Zhang, F. (2025). Entropy-Optimized Dynamic Text Segmentation and RAG-Enhanced LLMs for Construction Engineering Knowledge Base, Applied Sciences, 15(6), 3134. DOI: https://doi.org/10.3390/app15063134.

Wang, X., Wang, Z., Gao, X., Zhang, F., Wu, Y., Xu, Z., Shi, T., Wang, Z., Li, S., Qian, Q., Yin, R., Lv, C., Zheng, X. & Huang, X. (2024). Searching for Best Practices in Retrieval-Augmented Generation. ArXiv: https://arxiv.org/abs/2407.01219.

Weber, T., Brandmaier, M., Schmidt, A. & Mayer, S. (2024). Significant Productivity Gains through Programming with Large Language Models, PACM HCI, 8(EICS), 1–29. DOI: https://doi.org/10.1145/3661145.

Yager, K. G. (2024). Towards a Science Exocortex, Digital Discovery, 3(10), 1933–1957. DOI: https://doi.org/10.1039/D4DD00178H.

Yan, Y., Liao, Y., Xu, G., Yao, R., Fan, H., Sun, J., Wang, X., Sprinkle, J., Ziyan, A., Ma, M., Cheng, X., Liu, T., Ke, Z., Zou, B., Barth, M. & Kuo, Y.-H. (2025). Large Language Models for Traffic and Transportation Research: Methodologies, State of the Art, and Future Opportunities. DOI: https://doi.org/10.48550/arXiv.2503.21330.

Yu, H., Gan, A., Zhang, K., Tong, S., Liu, Q. & Liu, Z. (2025). Evaluation of Retrieval-Augmented Generation: A Survey, in: Big Data, Springer Nature Singapore, 102–120. ArXiv: https://arxiv.org/abs/2405.07437.

Zheng, Z., Zhang, J., Vu, T.-A., Diao, S., Tim, Y. H. W. & Yeung, S.-K. (2023). MarineGPT: Unlocking Secrets of Ocean to the Public. DOI: https://doi.org/10.48550/arXiv.2310.13596.

Zhou, X., He, J., Zhou, W., Chen, H., Tang, Z., Zhao, H., Tong, X., Li, G., Chen, Y., Zhou, J., Sun, Z., Hui, B., Wang, S., He, C., Liu, Z., Zhou, J. & Wu, F. (2025). A Survey of LLM × DATA. DOI: https://doi.org/10.48550/arXiv.2505.18458.

Zolfaghari, V., Petrovic, N., Pan, F., Lebioda, K. & Knoll, A. (2024): Adopting RAG for LLM-Aided Future Vehicle Design, FLLM 2024, IEEE, 437–442. ArXiv: https://arxiv.org/abs/2411.09590.