Article

Leveraging Large Language Models to Enhance an Intelligent Agent with Multifaceted Capabilities

Sree Ganesh Thottempudi * and Sagar Borra

Posted Date: 19 September 2024

doi: 10.20944/preprints202409.1446.v1

Keywords: large language models; retrieval augmented generation

Article

Leveraging Large Language Models to Enhance an Intelligent Agent with Multifaceted Capabilities

Sree Ganesh Thottempudi * and Sagar Borra

Berlin School Of Technology, SRH University Berlin, Germnay
* Correspondence: sreeganesh.thottempudi@srh.de

Abstract: This project aims to create a virtual assistant with AI integration to improve Siemens Energy's internal processes. Using cloud-based technologies, microservice architecture, and large language models (LLMs), the project seeks to create a reliable, effective, and user-friendly assistant customized to Siemens Energy's requirements. The first significant business difficulty identified by the study was the time engineers had to spend looking for information in large volumes of company papers. The proposed virtual assistant responds with precision and context awareness to optimize productivity. The assistant uses a microservice architecture to guarantee scalability, flexibility, and integration for various use scenarios. Tasks like document retrieval, translation, summarization, and comparison can now be handled effectively. Utilizing Amazon Web Services (AWS) for cost-effectiveness and scalability, the backend is cloud-deployed, backed by a frontend created for natural user interaction. To increase precision and relevance, the system uses cutting-edge AI, such as vector databases and Retrieval Augmented Generation (RAG). The assistant expedites document management procedures, improves data accessibility, and reduces search time. The results highlight how it may enhance workflow efficiency for Siemens Energy engineers and how flexible it can be for future AI-driven applications.

Keywords: large language models; retrieval augmented generation

1. Introduction

Thanks to their ability to automate tedious operations, manage information, and optimize workflows, intelligent assistants have become essential components of modern organizations. Siemens Energy, a global corporation renowned for its advancements in traditional and sustainable energy solutions, confronts formidable obstacles concerning data accessibility and administration. The amount of data generated daily by operations spanning 90 nations is daunting, especially for engineers who frequently have to spend significant time looking for specific information across multiple papers. To overcome these obstacles, this thesis aims to create an AI-based intelligent assistant that will drastically cut down on search times and boost output.

Large Language Models (LLMs), such as GPT -4 and Claude, have demonstrated great promise in natural language creation and understanding. These models are perfect for creating an intelligent assistant customized to meet a business's unique requirements since they can comprehend context, handle sophisticated questions, and provide nuanced answers. The suggested approach focuses on integrating LLMs into a microservice architecture to offer Siemens Energy a robust, scalable, and adaptable platform that can meet its changing and varied needs.

The first step in the study process was thoroughly examining the issue. Because of the volume and variety of data that Siemens Energy stores inside the company, information retrieval becomes a bottleneck for the engineers. Currently in use, traditional AI systems are restricted to particular tasks and need the contextual awareness required to tackle more intricate, multiple questions. Furthermore, the current solutions cannot meet the organization's growing needs across many departments and use cases because they are not scalable enough. Therefore, the necessity for an AI solution that can smoothly combine several functionalities into a single, cohesive system is urgent.

This study suggests a virtual assistant built on generative AI and sophisticated LLMs to solve these problems. The assistant will be constructed with a microservice architecture, which guarantees flexibility and scalability and enables it to process complex queries and various kinds of documents. The AI-powered assistant will be a flexible tool for engineers who require fast and precise information to make decisions because it can retrieve documents, summarize, translate, and compare data. It will also support many languages, letting the global staff of Siemens Energy communicate and work together.

The architecture of the assistant is highly flexible, allowing it to incorporate new features and functionalities as the demands of the company change. Amazon Web Services (AWS) will be used for the backend deployment, guaranteeing reliable performance, scalability, and affordability. Retrieval Augmented Generation (RAG), other advanced retrieval techniques, and vector databases like Qdrant will significantly improve the system's capacity to deliver accurate and pertinent responses. Together, these elements provide a complete solution that resolves Siemens Energy's present issues and lays the groundwork for upcoming AI-driven improvements.

The suggested solution seeks to improve the efficiency and speed of information retrieval by streamlining the management of Siemens Energy's extensive knowledge base. Using a microservice-based methodology and LLMs, the assistant will be an effective tool that improves decision-making and efficiency throughout the company. This study advances the topic of artificial intelligence (AI) in enterprise solutions and shows how cutting-edge AI technology can be integrated to address practical business problems.

2.Related Work:

2.1. Understanding Large Language Models

The paper “Talking about Large Language Models” by Shanahan, M. highlights the significance of comprehending large language models' underlying mechanics as it addresses the possibilities and limitations of LLMs. It draws attention to the notable gains in performance that are shown as training data and model parameters are increased, with LLMs demonstrating surprisingly good results in next-token prediction tasks. It does, however, issue a warning against anthropomorphizing these models as this may give rise to false beliefs about their capacity. LLMs fundamentally differ from human cognition and interaction, even though they can carry out many tasks that need human intelligence. To ensure that decisions about the deployment and utilization of LLMs are well-informed, the document suggests that developers and users concentrate on the mathematical and technical concepts underlying LLMs rather than imputing human-like characteristics (Shanahan, 2024).

The “Challenges and applications of large language models" by Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., & McHardy, R. lists the main difficulties and areas in which large language models (LLMs) can be used. Unfathomable datasets, high inference latency tokenizer reliance, hallucinations, indistinguishability between generated and human-written text, limited context length, high pre-training costs, prompt brittleness, misaligned behavior, tasks not solvable by scale, outdated knowledge, fine-tuning overhead, lack of reproducibility, brittle evaluations, and lack of experimental designs are just a few of the challenges that fall into the design, behavioural, and scientific categories. Applications in chatbots, computer programming, computational biology, legal studies, medical research, reasoning, robotics, social sciences, and synthetic data production were all investigated. The paper emphasizes how these difficulties limit the applications and the need for more study to overcome these constraints and improve the effectiveness of LLMs in various fields (Kaddour et al., 2023).

The paper “Leveraging Generative AI and Large Language Models: A Comprehensive Roadmap for Healthcare Integration” by Yu, P., Xu, H., Hu, X., & Deng, C. study findings emphasizes the possibilities and difficulties of integrating LLMs, such as ChatGPT, in the healthcare industry. Two primary concept clusters emerged from the scoping review of sixty-three papers: one was centered on ChatGPT, models, patients, and research, while the other was centered on answers, doctors, and queries. The technological approaches to generative AI applications, training strategies, model evaluations, and contemporary applications in healthcare are among the key findings, along with ethical and legal challenges and future research paths needed to solve problems like hallucinations and develop responsible AI frameworks, benefits like improved diagnostic support and automated documentation were recognized (Yu et al., 2023).

2.2. Retrieval Augmented Generation

Retrieval-augmented generation (RAG) is examined in the paper "Retrieval-Augmented Generation for Large Language Models: A Survey" by Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., & Wang, H. as a potential solution to Large Language Models (LLMs) drawbacks such as hallucinations, out-of-date information, and untraceable reasoning. The article examines the development of RAG, outlining the three primary paradigms: Modular RAG, Advanced RAG, and Naive RAG. Every paradigm improves the cooperation of the augmentation, generation, and retrieval processes. While modular RAG adds flexibility with specific modules for various activities, naive RAG concentrates on basic retrieval and generation, while advanced RAG enhances retrieval quality and context integration. The survey addresses the assessment of RAG systems, identifies the most advanced technologies in each paradigm, and makes recommendations for future research to address pressing issues. RAG improves LLMs' performance in real-world applications by continuously updating knowledge and integrating domain-specific information (Gao et al., 2023).

The work presented in the article "Benchmarking large language models in Retrieval-Augmented Generation" by Chen, J., Lin, H., Han, X., & Sun, L. methodically investigates how well RAG improves LLMs. Four primary competencies are the subject of this study: counterfactual robustness, information integration, hostile rejection, and noise robustness. The researchers created the Retrieval-Augmented Generation Benchmark (RGB), a unique corpus of examples in both English and Chinese, to make this evaluation easier. Six representative LLMs were evaluated using RGB, and the results showed that although these models exhibit some resilience to noise, they still have severe problems with correctly integrating information, rejecting irrelevant information, and handling counterfactuals. The results highlight the need for ongoing improvement to successfully use RAG in LLMs and guarantee precise and trustworthy model answers (Chen et al., 2023).

This article, "Hallucination reduction in large language models with Retrieval-Augmented Generation using Wikipedia knowledge," by Kirchenbauer, J., & Barns, C., presents research on the use of RAG with Wikipedia as an external knowledge source to mitigate hallucinations in LLM. The study draws attention to the ongoing problem of hallucinations, which undermines the validity of models by producing false or distorted data. Integrating dynamically collected, pertinent Wikipedia content allowed the Mistral model to demonstrate notable gains in response quality, recall, and precision. This study shows that RAG can significantly lower hallucinations, offering a solid paradigm for improving the veracity and reliability of LLM outputs, particularly in vital fields like law, healthcare, and education. The results highlight RAG's potential to promote using trustworthy AI systems in various applications (Kirchenbauer & Barns, 2024).

2.3. Applications of LLMs

The paper "Large Language Models for Information Retrieval: A Survey" by Zhu, Y., Yuan, H., Wang, S., Liu, J., Liu, W., Deng, C., Dou, Z., & Wen, J. presents research findings that demonstrate how LLMs have a revolutionary effect on information retrieval (IR) systems. The paper explains how IR has developed from term-based approaches to sophisticated neural models, highlighting how LLMs with more extraordinary language creation, interpretation, and reasoning skills, such as ChatGPT and GPT-4, have entirely changed the discipline. It discusses essential topics, including retrieval, reranking, reading, and query rewriting, and demonstrates how LLMs improve these elements by yielding more precise and contextually appropriate results. Notwithstanding its benefits, problems like interpretability and data scarcity still exist. The survey also looks into how LLMs could be used as search agents to handle specific information retrieval tasks and make the user's experience more efficient. The thorough overview and insights are intended to guide future research in this quickly developing topic and to unify techniques (Zhu et al., 2023).

The study "Towards Hybrid Architectures: Integrating Large Language Models in Informative Chatbots" by Von Straussenburg, A. F. A., & Wolters, A. (n.d.). investigates how to improve the performance and dependability of conventional chatbot technologies by integrating LLMs to deliver precise and interesting user interactions. The researchers point out that whereas conventional chatbots are great at retrieving organized data and guaranteeing accuracy, they frequently have trouble producing responses that seem human. On the other hand, LLMs like the GPT-4 show better aptitude for comprehending and producing natural language. Still, they might also generate answers that must be supported by verifiable information. The suggested system combines the advantages of both approaches through inter-agent communication in a hybrid architecture. The goal of this hybrid approach is to increase the chatbot's overall efficacy by producing responses that are accurate and human-like. Prototyping this conception and assessing its performance are the next steps, and the model will be iteratively improved based on continuous evaluations (Von Straussenburg & Wolters, n.d.).

The study discussed in this article, "Enhancing large language model performance to answer questions and extract information more accurately" by Zhang, L., Jijo, K., Setty, S., Chung, E., Javid, F., Vidra, N., & Clifford, T. aims to improve LLMs' functionality and accuracy in information extraction and question answering. The authors improved the models through fine-tuning that involved examples and feedback. Using LLMs like GPT-3.5, LLaMA2, GPT4ALL, and Claude, they evaluated the models using metrics like Rouge-L and cosine similarity scores. By evaluating these models against financial information, the study showed that refined models can outperform zero-shot LLMs in accuracy. Combining fine-tuning with RAG is one noteworthy strategy that the research highlights, and it worked well to improve response accuracy. This technique enhances the generated replies' relevance and dependability by obtaining pertinent papers to offer context. The study emphasizes how critical it is to address problems like hallucinations in LLMs, especially in high-stakes fields like banking, where accuracy is essential (Zhang et al., 2024).

Language Model Programming (LMP) is discussed in the paper "Prompting is programming: a query language for large language models" by Beurer-Kellner, L., Fischer, M., & Vechev, M. LMP improves the flexibility and usefulness of large language models (LLMs) by adding output limitations and lightweight scripting to classic natural language prompting. The Language Model Query Language (LMQL), which integrates declarative and imperative programming components to simplify communication with LLMs, is introduced by the authors to enable LMP. By streamlining the inference process, LMQL can cut latency and processing expenses by up to 80% without sacrificing job accuracy. This innovative method simplifies the challenges associated with vendor-specific libraries and model-specific implementations, offering a more user-friendly and effective means of utilizing LLM capabilities for a range of downstream activities. The assessment shows how well LMQL works to improve and streamline sophisticated prompting methods (Beurer-Kellner et al., 2022).

The research paper "Personal LLM Agents: Insights and Survey about the Capability, Efficiency and Security" study examines the development and possibilities of intelligent personal assistants (IPAs), mainly when LLMs are used. The versatility and scalability of existing IPAs, such as Siri and Google Assistant, are constrained; they are mostly capable of carrying out activities inside designated domains and necessitate explicit programming for additional operations. The study shows how LLMs significantly improve the field, especially in comprehending user intent, reasoning, and autonomous problem-solving. This implies that these models have the potential to increase the functionality and usefulness of IPAs dramatically. Future IPAs could handle more complicated tasks autonomously and quickly by utilizing LLMs' strong semantic and reasoning capabilities, revolutionizing user interaction and functionality (Y. Li et al., 2024).

The research described by Guan, Y., Wang, D., Chu, Z., Wang, S., Ni, F., Song, R., Li, L., Gu, J., & Zhuang, C.'s paper "Intelligent Virtual Assistants with LLM-based Process Automation" describes the creation and assessment of a brand-new intelligent virtual assistant system that is powered by LLMs and was created especially for automating mobile apps. This system, known as LLMPA, uses an executor and environmental context to complete multi-step tasks using natural language instructions automatically. The system outperformed baseline models in large-scale testing on the Alipay platform, demonstrating notable success in comprehending user goals and preparing and

doi:10.20944/preprints202409.1446.v

carrying out complex tasks. Its strong performance depended on essential elements like Previous Action Descriptions and Instruction Chains. While noting the need for additional advancements in contextual processing, reasoning capabilities, and optimized on-device deployment to fully realize virtual assistants' potential in daily use, the research emphasizes the system's potential for widespread adoption in mobile applications (Guan et al., 2023).

As a means of examining divergent chains of thought, the research findings provided in the paper “Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate” by Liang, T., He, Z., Jiao, W., Wang, X., Wang, Y., Wang, R., Yang, Y., Tu, Z., & Shi, S. identify and suggest the "Degeneration-of-Thought" (DoT) problem in self-reflection and present the Multi-Agent Debate (MAD) framework. The MAD framework proved its worth in two complex tasks, demonstrating that GPT-3.5-Turbo with MAD can outperform GPT-4 on the Common MT dataset. According to the study, optimal results require adaptive break techniques and a moderate amount of "tit for tat" interactions. Using agents with identical backbone LLMs also improves performance. Furthermore, if various models are employed for agents, the research implies that LLMs might not be unbiased judges, suggesting a potential bias in multi-agent interactions. Future topics for study include using multi-agent systems in board games, increasing the number of agents participating in debates, and using AI feedback to align models (Liang et al., 2023).

The study results presented in the paper “Enhancing Cloud-Based Large Language Model Processing with Elasticsearch and Transformer Models” by Ni, C., Wu, J., Wang, H., Lu, W., & Zhang, C. highlight several essential points. Based on Yelp's dataset, the study used sophisticated Transformer-based models like BERT, DistilBERT, and RoBERTa, along with traditional machine learning models to predict emotions. It was discovered that RoBERTa achieves the best accuracy improvement by 0.62%, while BERT and RoBERTa show greater classification accuracy. DistilBERT is proven to be beneficial in terms of training time. The research highlights the capabilities of Transformer pre-training models for handling practical text and proposes further research into multimodal models to improve performance despite existing computing limitations. It also goes into how Elasticsearch can help with cloud systems' scalability and efficiency issues, improving LLM processing capabilities (Ni et al., 2024).

With a particular emphasis on the application of LLMs in wireless networks, the paper “Wireless Multi-Agent Generative AI: From Connected Intelligence to Collective Intelligence” by Zou, H., Zhao, Q., Bariah, L., Bennis, M., & Debbah, M. describes the creation and integration of multi-agent generative AI networks. It highlights how on-device LLMs can work together to solve challenging problems using logic and planning. Reinforcement learning and multi-agent LLM games are essential enabling technologies that help achieve optimal cooperation behaviors. The study highlights how crucial semantic communication is to efficient knowledge transfer in networks. Intent-driven network automation is one of the potential uses for 6G networks that are examined, along with a use case showing how multi-agent LLMs might improve user transmission rates and network energy efficiency. Future research directions in system 2, machine learning, human-agent interaction, and the broader application of generative AI in wireless networks are highlighted in the paper's conclusion (Zou et al., 2023).

The research findings in the article “Creating large language model applications Utilizing LangChain: A primer on developing LLM apps fast” by Topsakal, O., & Akinci, T. C. demonstrate how LLMs, especially OpenAI's ChatGPT, have revolutionized the field of AI. The paper highlights the novel function of the open-source library LangChain, which offers modular architecture and adaptable pipelines to make creating LLM applications easier. One significant development that simplifies the development of complex AI applications is LangChain's ability to interact with several data sources and apps. The results point to the great promise that LangChain's adaptability and efficiency have for future AI advancements, which will stimulate more research and development in the LLM space (Topsakal & Akinci, 2023).

3. Proposed Method:

Before getting into the system's actual functionality, let's examine the Generative AI framework, which is the base for this assistant.

3.1. Framework: LlamaIndex

LlamaIndex is a data framework created to help LLMs integrate and interact with outside data sources. It serves as a layer of intermediate that makes it easier for LLMs to retrieve, analyze, and use data from various sources, increasing task efficiency and effectiveness. When enormous amounts of data must be maintained and accessible by AI models to produce intelligent insights, automated reactions, or improved decision-making, LlamaIndex is very helpful.

Figure 1 depicts the visual representation of the LlamaIndex framework. It illustrates the essential components and procedures that makeup LlamaIndex.

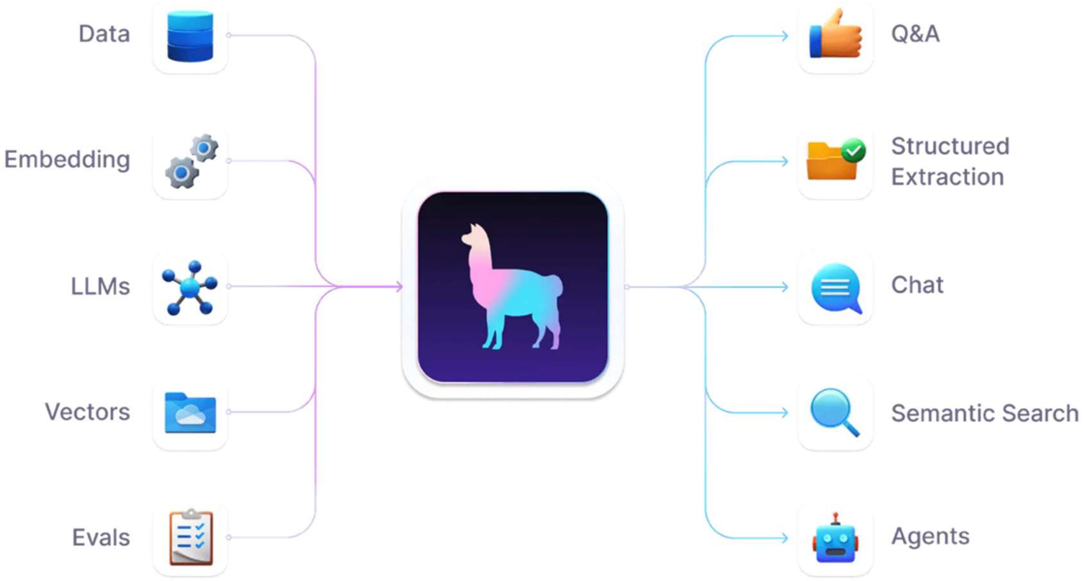


Figure 1. LlamaIndex Framework.

Let’s break down this figure.

1. Data Sources (Left side):
   - Data: Denotes the unprocessed information in files, databases, or other data repositories.
   - Preprocessing: This stage entails cleaning, converting, and arranging the data to prepare it for use by LLMs.
   - LLMs: They use the processed data for a variety of purposes.
   - Indexes: These are organized data representations that make effective searching and retrieval possible.
2. LlamaIndex (Center):
   - The central icon represents the heart of the LlamaIndex framework. This gradient-colored llama unifies all the parts and facilitates communication between data sources and language models.
3. Applications (Right side):
   - Q&A: The LLMs can respond precisely to user inquiries because of the processed data and indexes.
   - Document retrieval: Enables pertinent documents to be found within a sizable corpus in response to user requests or specifications.
   - Chat: Enables conversational interfaces with intelligence that can access and use outside data.
   - Semantic search: Improves search performance by deciphering query context and semantics.

- Agents: Automated agents with decision-making and task-performing capabilities based on integrated data.

A robust architecture like LlamaIndex helps to connect big language models with outside data sources. It makes preprocessing, indexing, and data integration more efficient, allowing LLMs to work more productively on various activities.

Creating the AI-integrated virtual assistant includes several essential elements: data processing, workflow procedures, system integration, and architecture design.

3.2. Architecture

A microservice architecture underpins the system, providing fail-safe operations, scalability, and adaptability. Because each service runs separately from the others, adding new features is simple and doesn't interfere with the system as a whole. The backend is set up on AWS, using services like Lambda, S3, and Elastic Compute Cloud (EC2) for effective processing, retrieval, and storage. This cloud-based architecture ensures scalability and firm performance, making it ideal for managing extensive company activities.

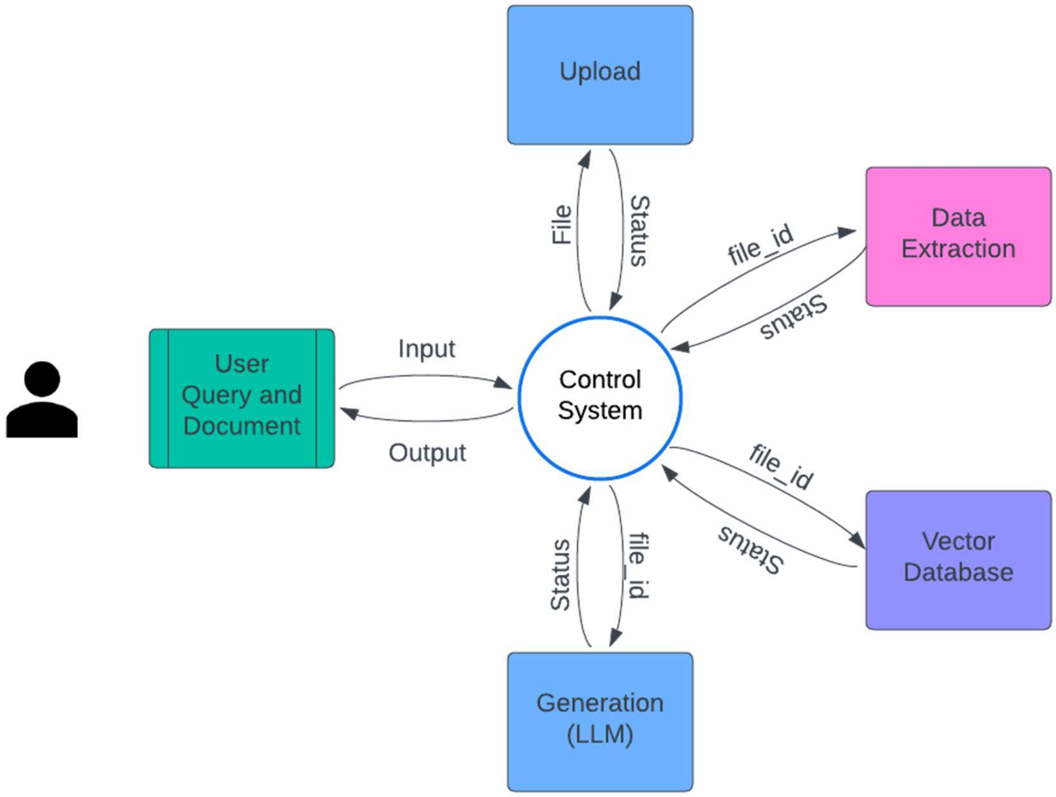


Figure 2. System Architecture.

3.2. Workflow of Different Processes

3.2.1. Upload, Extraction, Chunking, and Indexing

The workflow diagram below shows the three primary steps of the process: uploading, extracting, chunking, and indexing. During the uploading stage, an employee logs into a Mendix application and uploads a file. The next step is initiated by the Mendix application, which calls for an Apprunner service, which saves the uploaded file in an S3 bucket.

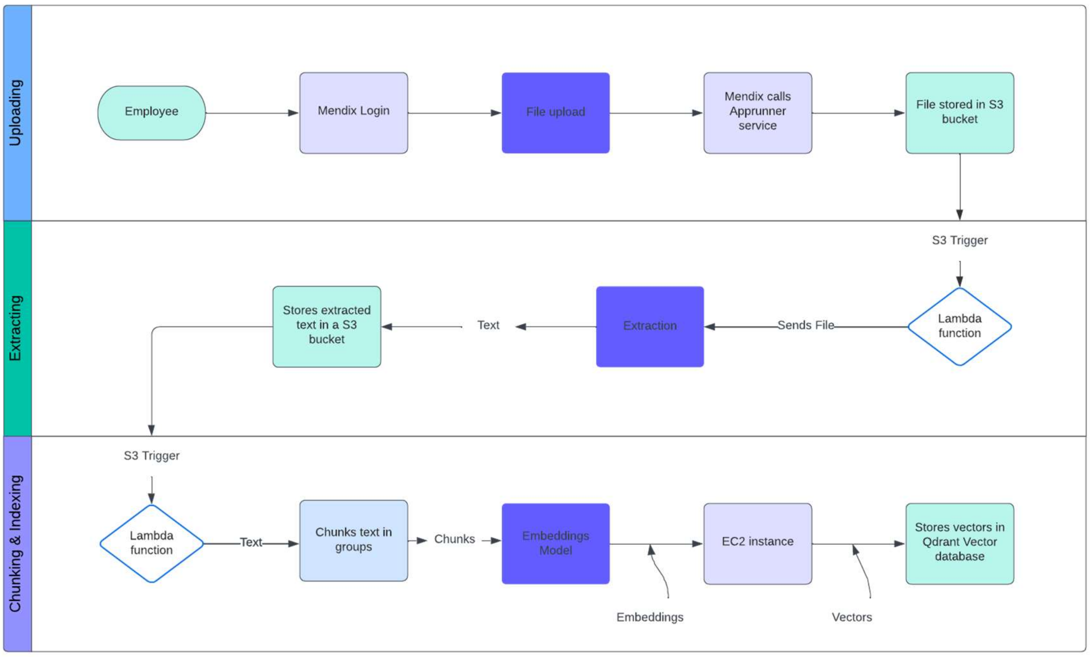


Figure 3. Workflow of uploading, extraction, chunking, and indexing.

An S3 trigger triggers a Lambda function and transfers the file to an extraction service during extraction. The text content of the file is extracted by the extraction service, which then stores the text back into an S3 bucket. Another S3 trigger initiates the third stage, Chunking & Indexing, by triggering a Lambda function that obtains the extracted text. After that, this function divides the text into smaller chunks that an EC2 instance's embedding model can process. These text chunks are transformed into vectors by the embedding model and kept in a Qdrant vector database for effective indexing and retrieval.

3.2.2. Retrieval and Generation

The workflow diagram in Figure 4 shows the method for responding to user inquiries in a Mendix chat interface. It consists of two main phases: retrieval and generation. During the Retrieval phase, a user asks the Apprunner Service a question over Mendix Chat. The Apprunner Service searches an EC2 instance to retrieve a vector index representing the query in a high-dimensional vector space. This index uses vector similarity search to find pertinent data points and retrieves similar vectors from the EC2 instance.

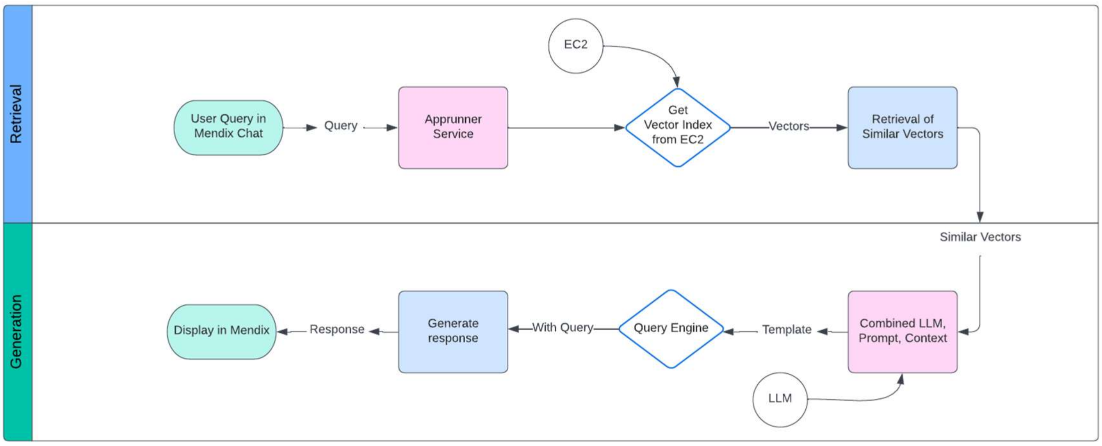


Figure 4. Workflow of retrieval and generation.

The Generation step gets underway with similar vectors retrieved from the retrieval procedure. The original query and these vectors are sent to a query engine. After that, the Query Engine works with an LLM, combining the query with pertinent background information to create a template that produces a logical result. The final response is generated by combining the contextual, prompt, and LLM data, and it is presented to the user via the Mendix interface. This process effectively combines sophisticated language models for response creation with vector search for information retrieval to deliver precise and contextually appropriate responses to user inquiries.

3.3. Data

Data is the fundamental element that powers the AI-integrated virtual assistant created for Siemens Energy in terms of both functionality and efficacy. The primary resource in this system that is processed, examined, and retrieved to offer insightful answers to users' inquiries is data. A wide variety of data formats, including text documents, PDFs, spreadsheets, presentations, and more, are handled by the virtual assistant. These documents include project paperwork, engineering manuals, technical reports, safety standards, and standard operating procedures, among other structured and unstructured data pertinent to Siemens Energy's day-to-day operations.

3.3.1. Data Types and Sources

The system is flexible and has various data-handling capabilities. It can process a variety of data formats, including DOCX, PDF, TXT, MD, PPT, and CSV. This adaptability is essential for Siemens Energy since it handles copious amounts of data generated from many sources, such as internal databases, document management systems, and cloud storage platforms. Technical specs, engineering reports, maintenance records, and regulatory compliance documentation are a few examples of the data sources.

3.3. How Exactly Does It Work?

The AI-powered virtual assistant for Siemens Energy uses a complex architecture combining cloud-based solutions, microservice design, and large language models (LLMs). The system uses sophisticated data processing, retrieval, and natural language understanding algorithms to give users accurate and context-aware answers to their questions.

3.4.1. System Design

The virtual assistant's microservice design enables easy maintenance, scalability, and flexibility. Each microservice handles tasks such as data extraction, indexing, retrieval, and natural language processing. Thanks to this decoupling of services, the system will continue to be reliable and flexible in response to shifting business needs.

The system's backend is implemented on Amazon Web Services (AWS) using services such as Lambda, Simple Storage Service (S3), and Elastic Compute Cloud (EC2). EC2 offers scalable computing power for diverse processes, and serverless services like workflow initiation in response to specific events (like document uploads) are managed by Lambda functions. Large datasets can be stored on S3, which acts as the storage layer and guarantees excellent availability and durability.

3.4.2. Data Ingestion and Preprocessing

Data ingestion occurs initially when a user uploads a document or collection of documents. The system supports several file formats, including PDFs, DOCX, TXT, PPT, and CSV. It must undergo a particular extraction procedure to turn any file into machine-readable content. For example, PDFs are handled by PDFPlumber, and Python tools such as AWS Textract deal with text extraction from other formats.

After extraction, the data is cleaned and preprocessed. This involves managing null data, fixing formatting errors, and eliminating extraneous characters. Preprocessing is essential to guarantee that the data is consistent and noise-free, which could compromise the accuracy of the AI-generated answers.

3.4.3. Chunking and Indexing

Depending on the document's structure and the type of content, the data is chunked into more manageable, meaningful pieces, like paragraphs or sentences, after preprocessing. Chunking is crucial for two primary reasons: it allows for more contextually appropriate responses and improves retrieval speed by dividing huge documents into smaller chunks. After being chunked, a vector database such as Qdrant indexes the data. High-dimensional vectors encode the semantic implications of the chunks and store them in vector databases. These vectors are produced using pre-trained embeddings from models such as BERT or other LLMs. Effective similarity searches are made possible using vector databases, in which user queries are compared to the indexed data to identify the most pertinent sections.

3.4.4. Retrieval Augmented Generation (RAG)

The Retrieval Augmented Generation (RAG) technique, which combines LLMs' generative and information retrieval capabilities, is the retrieval mechanism's foundation. The system first transforms a user-submitted query into a vector representation. After that, it searches the vector database for similarity to find the most pertinent informational fragments.

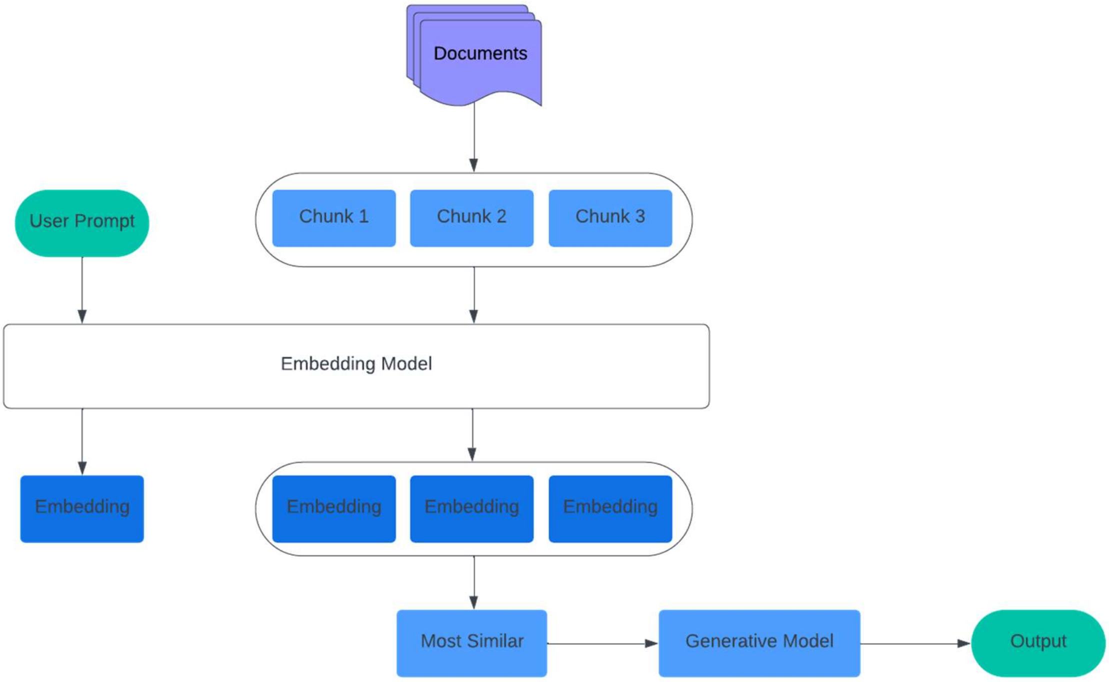


Figure 5. Naive RAG.

Following the retrieval of pertinent data, the LLM receives it and uses its prior knowledge to combine the data with its retrieval to produce a response. This combination guarantees that the response covers every facet of the user question and is contextually relevant.

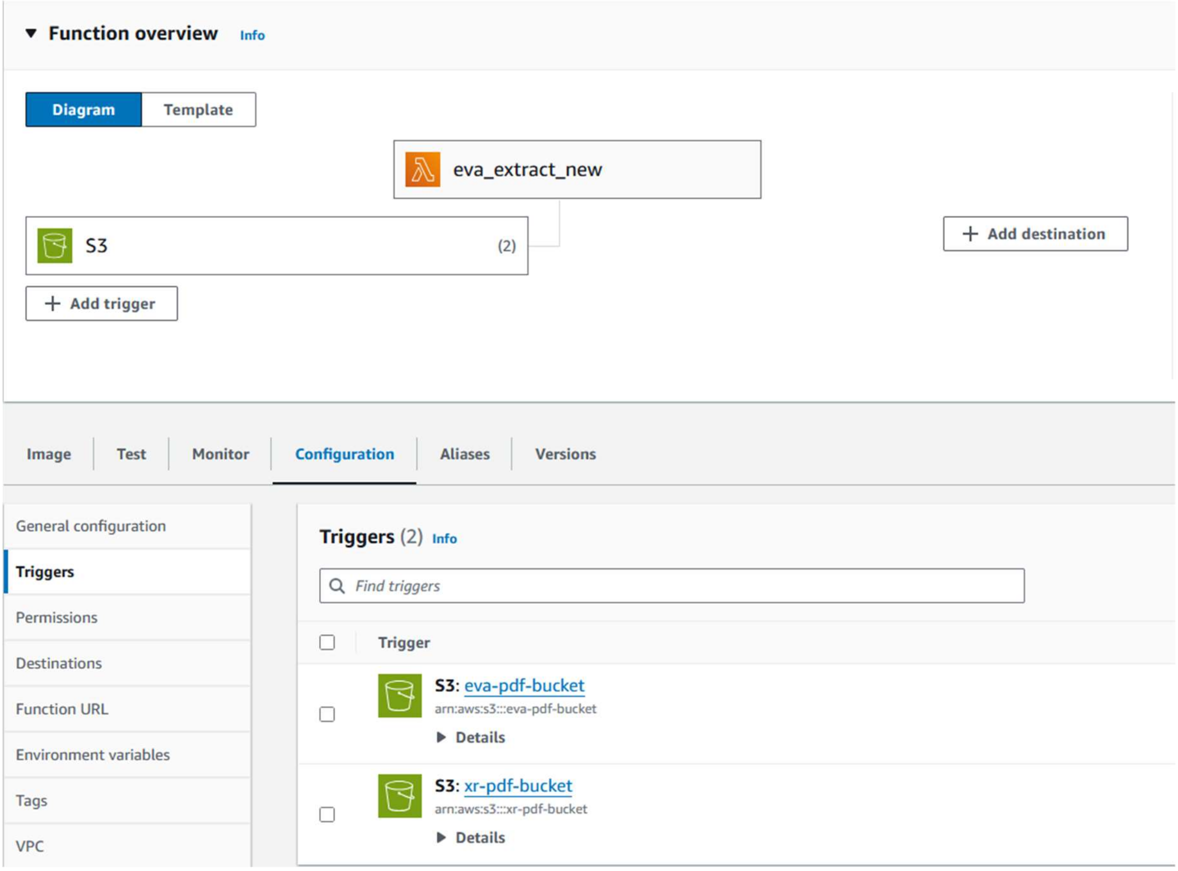
3.4.5. Natural Language Understanding and Response Generation

Advanced LLMs such as GPT-4 enable the virtual assistant to comprehend natural language and provide relevant responses. These models are refined using domain-specific data to understand the subtleties of Siemens Energy's processes, jargon, and technical terminology. After interpreting the purpose of the query and determining the essential entities and relationships, the LLM creates a logical response that fits the context that the received data provides.

3.4.6. Workflow Orchestration

The system uses workflow orchestration solutions like AWS Step Functions or Apache Airflow to manage the series of actions involved in data extraction, indexing, retrieval, and response generation. These tools ensure that dependencies between various activities are efficiently managed and each microservice is adequately activated.

For example, the Lambda function initiates the data extraction service upon document upload. Once the extraction is complete, the chunking and indexing services are triggered, followed by the preprocessing service. This orchestration allows tasks to be processed in parallel, boosting the system's effectiveness and responsiveness.



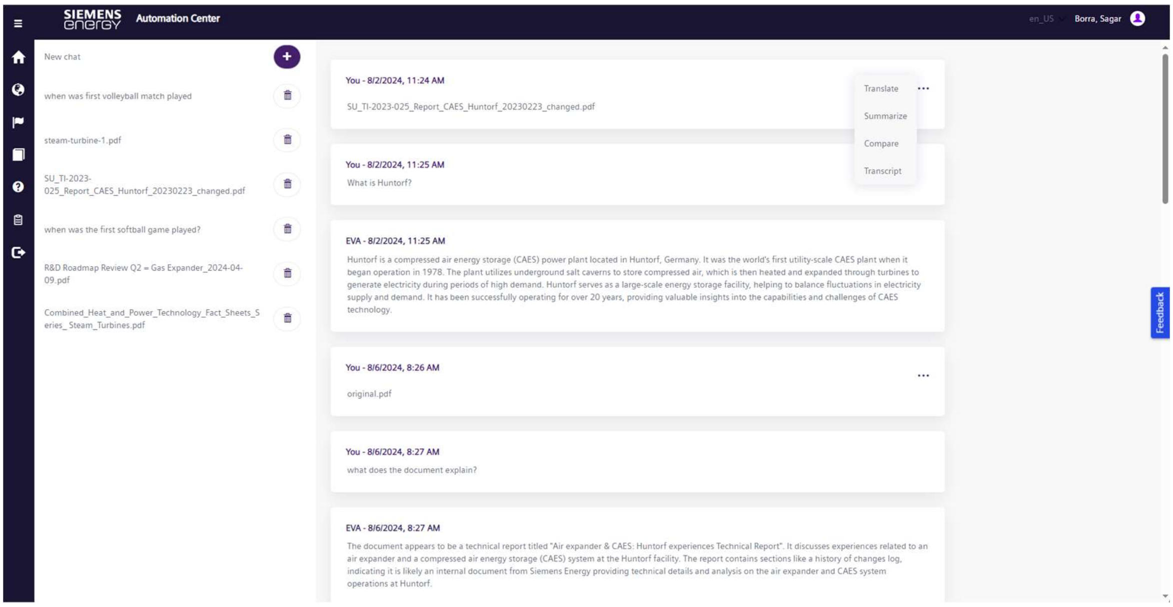
S3: eva-pdf-bucket
arn:aws:s3:::eva-pdf-bucket

S3: xr-pdf-bucket
arn:aws:s3:::xr-pdf-bucket

Figure 6. S3 trigger for a Lambda function.

3.4.7. Real-Time Query Handling

When a user submits a query via the assistant's frontend interface, the system starts a real-time query-handling workflow. The natural language understanding (NLU) component analyzes the query first, dissecting it into its component elements and determining the entities, intents, and any particular requirements.



when was first volleyball match played

steam-turbine-1.pdf

SU_TI-2023-025_Report_CAES_Huntorf_20230223_changed.pdf

when was the first softball game played?

R&D Roadmap Review Q2 = Gas Expander_2024-04-09.pdf

Combined_Heat_and_Power_Technology_Fact_Sheets_Series_ Steam_Turbines.pdf

You - 8/2/2024, 11:24 AM

SU_TI-2023-025_Report_CAES_Huntorf_20230223_changed.pdf

You - 8/2/2024, 11:25 AM

What is Huntorf?

EVA - 8/2/2024, 11:25 AM

Huntorf is a compressed air energy storage (CAES) power plant located in Huntorf, Germany. It was the world's first utility-scale CAES plant when it began operation in 1978. The plant utilizes underground salt caverns to store compressed air, which is then heated and expanded through turbines to generate electricity during periods of high demand. Huntorf serves as a large-scale energy storage facility, helping to balance fluctuations in electricity supply and demand. It has been successfully operating for over 20 years, providing valuable insights into the capabilities and challenges of CAES technology.

You - 8/6/2024, 8:26 AM

original.pdf

You - 8/6/2024, 8:27 AM

what does the document explain?

EVA - 8/6/2024, 8:27 AM

The document appears to be a technical report titled "Air expander & CAES: Huntorf experiences Technical Report". It discusses experiences related to an air expander and a compressed air energy storage (CAES) system at the Huntorf facility. The report contains sections like a history of changes log, indicating it is likely an internal document from Siemens Energy providing technical details and analysis on the air expander and CAES system operations at Huntorf.

Figure 7. Functionalities of EVA.

After that, the query is converted to a vector format and sent to the retrieval module, which uses the vector database for a similarity search. The LLM synthesizes the final response after retrieving the most pertinent pieces. The user is then shown this response via the frontend interface, designed for simple user interaction, ensuring a comfortable and intuitive experience.

3.4.8. Support for Multiple Languages and Document Formats

The assistant's multilingual and multiformat support improves its usability for Siemens Energy's international workforce. Language-specific LLMs and translation services provide multilingual support. Additionally, the assistant can handle various document types, so users can submit papers in any supported format and get correct results regardless of the language or format of the original document.

3.4.9. Continuous Learning and Adaptation

One of the main characteristics of a virtual assistant is its capacity to learn and change over time continuously. The system monitors user interactions, feedback, and response accuracy to increase its comprehension of user demands and modify its models. Using reinforcement learning techniques, the model weights are modified in response to user feedback, making the assistant more accurate and dependable as it is used more frequently.

3.4.10. Integration with Siemens Energy's Digital Ecosystem

ERP, CRM, and project management systems are just a few digital tools and platforms the virtual assistant works with smoothly. Through this connectivity, the assistant may access more data sources and offer users more thorough support, improving workflow efficiency and decision-making. Siemens Energy's AI-integrated virtual assistant functions combine sophisticated data processing, retrieval, and natural language synthesis methods. Utilizing cutting-edge AI models, cloud-based deployment, and microservice architecture, the system offers an enterprise-level scalable, adaptable, and reliable application. This assistant will facilitate Siemens Energy's data-driven decision-making, streamlined information retrieval, and markedly increased productivity, setting the stage for more AI-driven advancements.

3.3. Summary of Methodology

The suggested solution creates a dynamic, responsive, and effective virtual assistant by combining a microservice architecture, cloud deployment, advanced retrieval algorithms, and LLMs. With this all-encompassing strategy, Siemens Energy can be guaranteed that the assistant will fulfill its present and future requirements while offering an adaptive and scalable solution for enterprise-level AI integration.

4. Findings and Discussion

In this chapter, we will compare LLMs and vector databases. We will also examine the system's performance and analytics generated from the AWS console. The evaluation is carried out across different LLMs used to develop the system and highlights critical differences across vector databases used to create the production-ready system. Various factors are considered when testing LLMs, and there are vital considerations when speaking about vector databases.

4.1. LLM Results

Table 1 presents a comparative comparison of the different AI models' performance on various tasks. Every row denotes a certain task or assessment measure, and every column represents a distinct AI model. This is an explanation of the table's composition and how it highlights Claude 3 Sonnet's superiority:

Table 1. LLMs Comparison.

Name	**Claude 3 Sonnet**	**GPT-4o**	**Gemini 1.5 Pro**	**Llama-400B (early snapshot)**
Graduate level reasoning	**59.4%** 0-shot CoT	53.6% 0-shot CoT	-	-
Undergraduate level knowledge	**88.7%** 5-shot 88.3% 0-shot CoT	- **88.7%** 0-shot CoT	85.9% 5-shot -	86.1% 5-shot -
Code (Human Eval)	**92.0%** 0-shot	90.2% 0-shot	84.1% 0-shot	84.1% 0-shot
Multilingual math	**91.6%** 0-shot CoT	90.5% 0-shot CoT	87.5% 8-shot	-
Reasoning over text	**87.1** 3-shot	83.4 3-shot	74.9 Variable shots	83.5 3-shot Pre-trained model
Mixed evaluations	**93.1%** 3-shot CoT	-	89.2% 3-shot CoT	85.3% 3-shot CoT Pre-trained model
Math problem-solving	71.1% 0-shot CoT	**76.6%** 0-shot CoT	67.7% 4-shot	57.8% 4-shot CoT
Grade school math	**96.4%** 0-shot CoT	-	90.8% 11-shot	94.1% 8-shot CoT

- Tasks and Metrics: The Graduate Level Reasoning (GQPA, Diamond) assesses the model's capacity for graduate-level reasoning.
- Knowledge at the Undergraduate Level (MMLU): Evaluates the model's comprehension at the undergraduate level.
- Code (HumanEval): Assesses the model's comprehension and writing skills in code.
- Multilingual Math (MGSM): Evaluate the model's multilingual math problem-solving skills.
- The Reasoning Over Text (DROP, F1 Score) assessment gauges the model's capacity to analyze textual content and derive pertinent information.
- Mixed Evaluations (BIG-Bench-Hard): A combination of different challenging tests to assess overall performance.
- Math Problem-Solving (MATH): Evaluates how well the model can solve mathematical puzzles.
- Math for Grade 8 (GSM8K): Evaluate the model's aptitude for solving math problems usually assigned at the grade school level.
- Metrics of Performance:
  - 0-shot CoT (Chain of Thought): The model completes the task without being trained on comparable problems or given any previous instances.
  - 3-shot, 4-shot, 5-shot, and so forth: The model has a few samples (shots) to learn from before completing the task.

4.1.1. The Reason Claude 3 Sonnet Is More Reliable Excellent Results

Claude 3 Sonnet performs admirably on various tasks every time. For example, it scores 96.4% in grade school math (GSM8K) and 92.0% in code evaluation (HumanEval).

- Versatility and Generalization: The model demonstrates substantial versatility by displaying a great capacity to generalize across many task types, such as multilingual math and reasoning over text.
- Effectiveness of Chain of Thought (CoT): Claude 3 Sonnet scores remarkably well on tasks using 0-shot CoT, such as 88.3% in college-level knowledge (MMLU) and 71.1% in math problem-solving (MATH).
- Comprehensive Evaluations: With a 93.1% score in mixed evaluations (BIG-Bench-Hard), the model performs exceptionally well, demonstrating its adaptability to complicated tasks.
- Knowledge and Reasoning: Claude 3 Sonnet demonstrates superior cognitive abilities with high graduate-level reasoning scores (59.4%) and reasoning over text (87.1% in DROP, F1 score).

In conclusion, Claude 3 Sonnet is a better AI model in the provided comparative analysis because of its strong reasoning and problem-solving abilities, good performance on various tasks, and effective generalization.

4.2. Key Differences between Vector Databases

Qdrant offers a production-ready service developed in the safe Rust programming language. The user-friendly API that comes with Qdrant is made to store, find, and manage high-dimensional Points, which are simply vector embeddings enriched with payloads or metadata. These payloads become useful data bits that give users insightful information and increase search efficiency. Payload is comparable to the metadata in other vector databases, such as Chroma, since it includes vector-related information.

Table 2. Vector databases comparison – Part 1.

Name	Free tier	Self-Host	Managed in cloud	Open source	Size of vector dimensions	Metadata filtering	Hybrid search
Qdrant	Yes	Yes	Yes	Yes	Doesn’t have any hard limits	Yes	Yes (Sparse-Dense Vectors)
Weaviate	Yes	Yes	Yes	Yes	65535	Yes	Yes
Pinecone	Yes	No	Yes	No	20000	Yes	No
Faiss	Yes	Yes	No	Yes	50000	No	No
ChromaDB	In memory of server	Yes	Not yet	Yes	30000	Yes	Query

Table 3. Vector databases comparison – Part 2.

Name	Distributed Architecture	Data backup/ Restore	GRPC API	Authentication	Monitoring and Logging
Qdrant	Yes	Yes	Yes	Yes	Yes
Weaviate	Yes	Yes	Yes	Yes	Yes
Pinecone	Yes	Yes	Yes	Yes	Yes
Faiss	No	No	No	No	No
ChromaDB	Yes	Yes	No	No	No

Because Qdrant is written in Rust, it operates quickly and dependably even when faced with large loads. Quadrant sets itself apart from the other databases by the number of client APIs it offers. Currently, Qdrant is compatible with Go, TypeScript/JavaScript, Rust, and Python. It has numerous distance metrics, including Cosine, Dot, and Euclidean, and employs HSNW (Hierarchical Navigable Small World Graph) for vector indexing. It has a recommendation API built right in.

4.2.1. Important Things to Remember when Thinking About Qdrant are as Follows

- Because Qdrant is built in Rust, it is the most reliable and fast option for high-performance vector storage, even with enormous load volumes.
- Qdrant is unique because it supports client APIs and is designed for developers working with Rust, Go, Python, and TypeScript/JavaScript.
- Using the HSNW algorithm, Qdrant provides developers with various distance metrics, such as Dot, Cosine, and Euclidean, allowing them to select the one that best suits their particular use cases.

- With a scalable cloud service, Qdrant quickly moves to the cloud and offers an exploratory free tier. Regardless of the volume of data, its cloud-native architecture guarantees excellent performance.

To sum up, Qdrant is an effective tool for companies to harness the potential of semantic embeddings and transform text search. It provides an easy-to-integrate, dependable, and scalable high-dimensional data management system with superior query performance. Its open-source database permits ongoing enhancements, problem corrections, and development. Qdrant provides excellent performance, metadata filtering, a free self-hosted version, no hard limits on vector dimensions, hybrid search capabilities, and flexible deployment choices (self-hosted or cloud-managed).

4.3. Explanation of AWS Services

AWS has notable metrics for numerous services, which makes it handy for developers to analyze a service. Different metrics for different services are based on the parameters on which the service can be analyzed. As discussed earlier, the system is integrated with services like Apprunner and Lambda, which function as the primary services that the system relies on to run the user requests seamlessly. Below are a few analytics on these services.

4.3.1. EVA-Command: An Apprunner Service

Figure 8 shows the number of 200 responses, which means the number of successful reactions the service generates. The figure analyses the number of 200 responses over one month, and the highest response count was on July 10, with a 200-response count of 38.

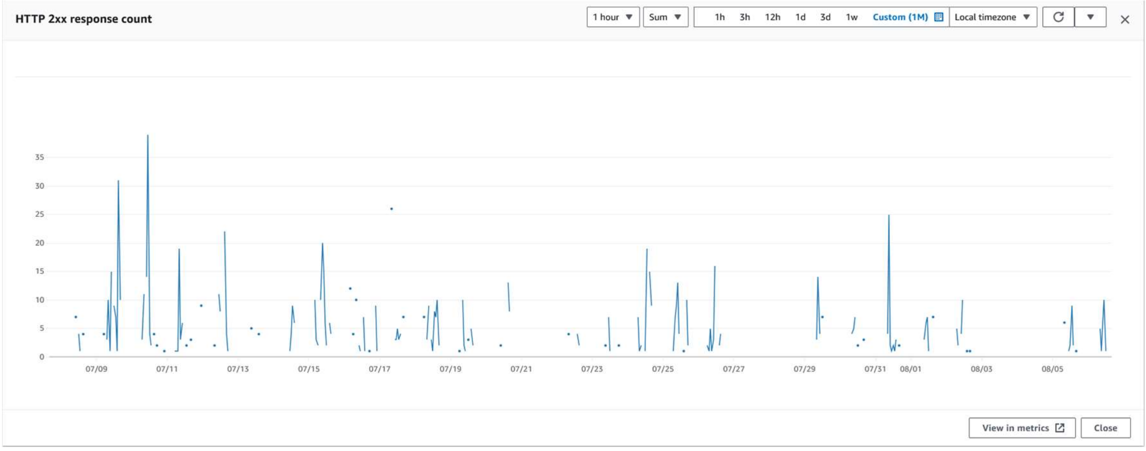


Figure 8. Successful responses of control system (eva_command).

Figure 9 shows how many 400 responses were generated, meaning many of the services generated unsuccessful reactions. The figure analyses the count of 400 responses over one month, and the highest 400 response count was on July 23, with a 400-response count of 3. This shows that there are a good number of errors in the code that shouldn't occur in a production system. These errors are shown in CloudWatch, highlighting which line of code the error occurred and why. This is a constructive way for developers to dive in and directly overcome mistakes.

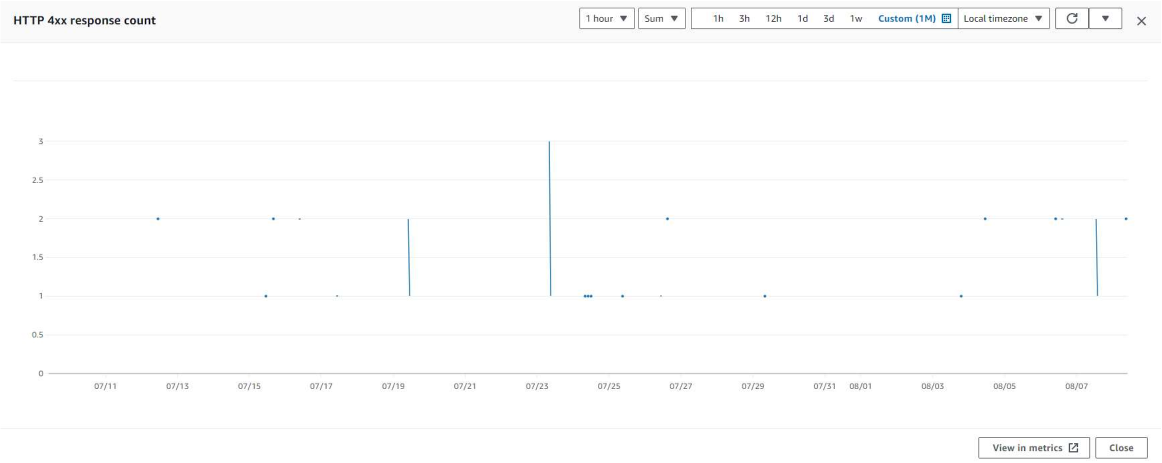


Figure 9. Unsuccessful responses of control system (eva_command).

Figure 10 shows the number of active instances at a point in time. This also analyses the number of active cases over one month. The Apprunner instance is configured so that it can have a maximum of 25 instances running simultaneously, and each instance can accommodate ten concurrent service calls. That means that there were at most ten service calls at the same time. Concurrent service calls are handled very well by Apprunner service, considering how low they cost, even if they are used 100 to 200 times a day.

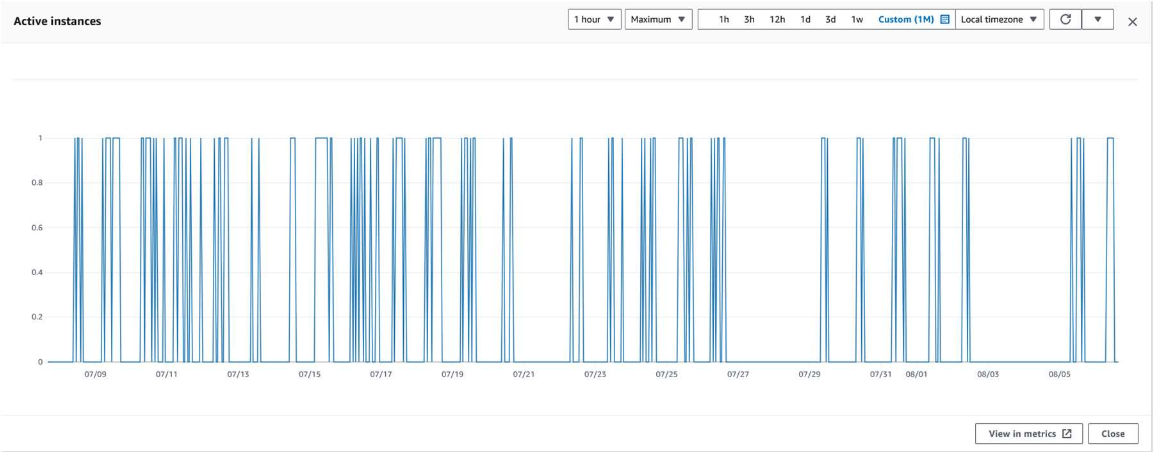


Figure 10. Active instances – EVA Command.

Figure 11 depicts the concurrent service calls simultaneously under one instance, with a maximum of 2 simultaneous calls.

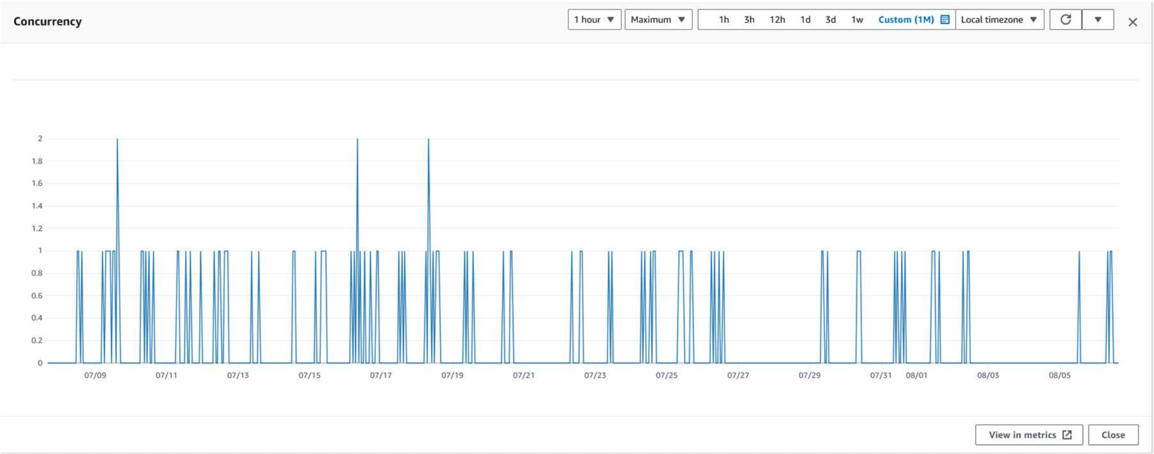
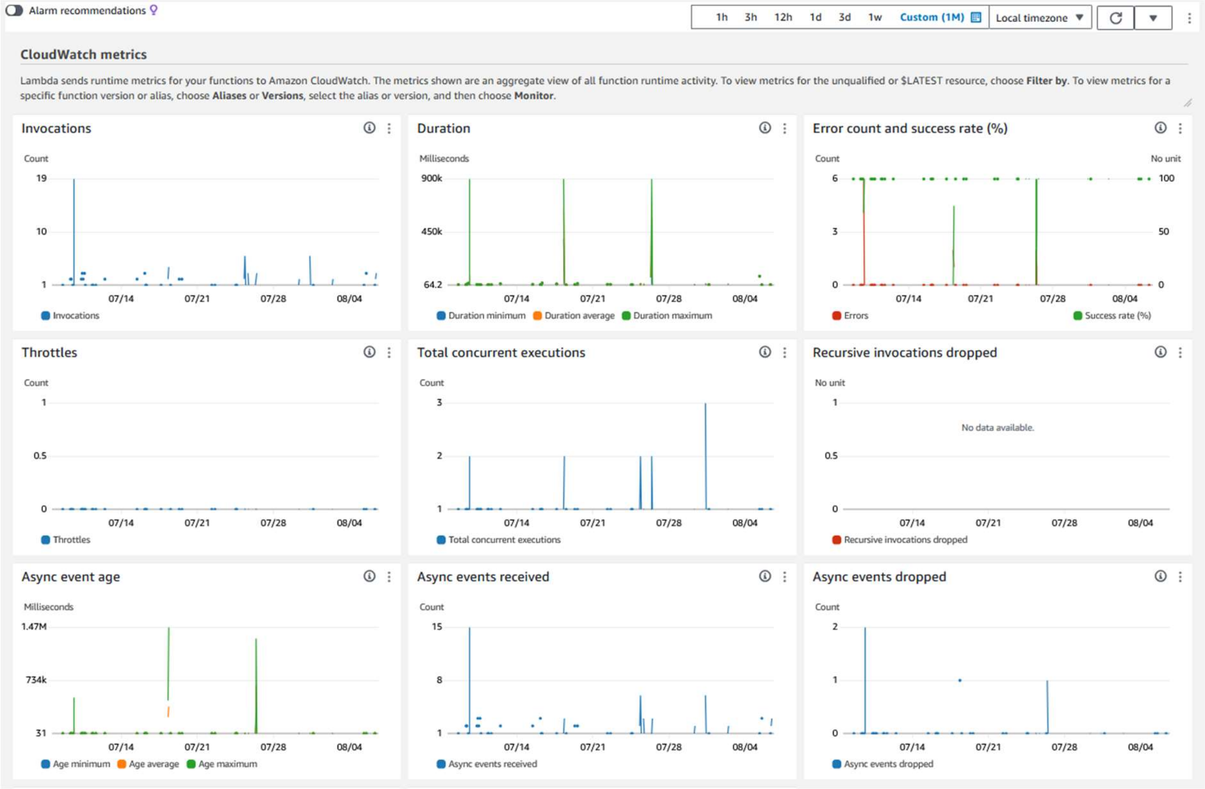


Figure 11. Number of current users for a service.

4.3.2. EVA-Extraction: A Lambda Function

As mentioned earlier, a service is analyzed based on different relevant parameters. Figure 12 shows different metrics, which are different from the metrics that we analyzed earlier with Apprunner services. Let's break down these metrics.


CloudWatch metrics

Lambda sends runtime metrics for your functions to Amazon CloudWatch. The metrics shown are an aggregate view of all function runtime activity. To view metrics for the unqualified or $LATEST resource, choose **Filter by**. To view metrics for a specific function version or alias, choose **Aliases** or **Versions**, select the alias or version, and then choose **Monitor**.

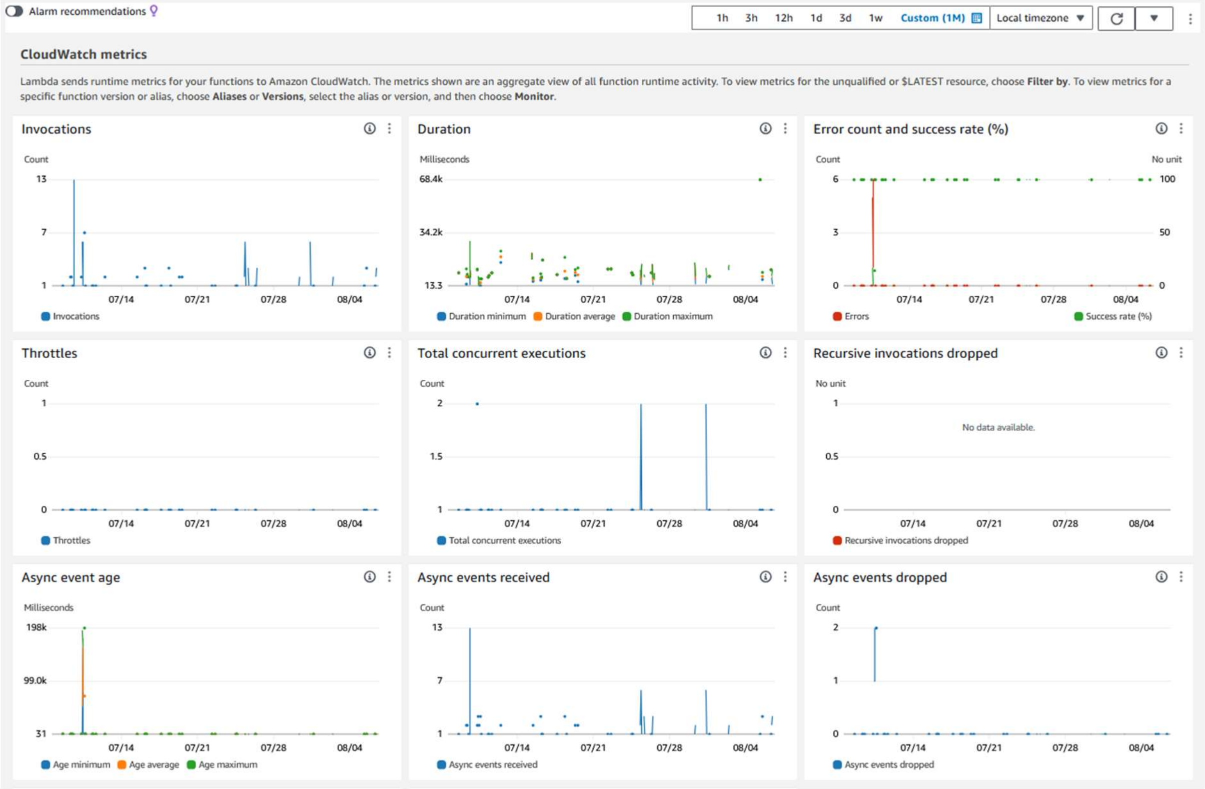
Figure 12. CloudWatch metrics – EVA Extraction.

Firstly, we have several invocations of the lambda function over one month, with a maximum number of invocations of 19. Next, we can see how long it took to complete the extraction process of every uploaded file. The 900000 milliseconds, which translates to 15 minutes, is the longest duration for a PDF file to be extracted. Up next, there is an interesting graph that indicates the count of success rate (in %) and error count. Success rate refers to how well the document was extracted, a 100% extracted file or some records are not extracted because of some content in the file, which led to an extraction of 70 % of the document. The error count is straightforward and occurs only when the extraction fails.

Next, the total number of concurrent executions is depicted with a maximum count of 3 concurrent extraction executions. The extraction time is just so less (5 to 10 seconds), and concurrent execution is only possible when a large document is being extracted. There are other service calls with a similar document length. Moving to the following graph, asynchronous extraction takes place for a document length of 400 to 500 pages. This might take up to 20 minutes, possibly having complex tables, non-readable texts, and more in the document. There is a graph beside the async event age graph for the number of async events received, with a maximum of 15 events. These events occurred in the larger document that we discussed earlier.

4.3.3. EVA-Vector Database: A Lambda Function

The metrics that are used to analyze the lambda function of the vector database (Figure 13) are the same ones that were previously used for the extraction. The significant difference between them is in the duration and the async event age. This is because converting the extracted text into vectors and storing them in the vector database is much faster than extraction. The maximum time even for a large document with 400 to 500 pages is 34.2 seconds, and the maximum async event age is just over 3 minutes. This clearly states the robustness of Qdrant database and how effective the service can be when hosted within AWS as an EC2 instance.



CloudWatch metrics

Lambda sends runtime metrics for your functions to Amazon CloudWatch. The metrics shown are an aggregate view of all function runtime activity. To view metrics for the unqualified or $LATEST resource, choose **Filter by**. To view metrics for a specific function version or alias, choose **Aliases** or **Versions**, select the alias or version, and then choose **Monitor**.

No data available.

Figure 13. CloudWatch metrics – EVA Vector database.

4. Conclusion and Future Work

A significant milestone in enterprise-level AI applications is the creation of a virtual assistant powered by AI for Siemens Energy. This study has shown how well LLMs, microservice architecture, and cloud-based solutions can be combined to build a dependable, scalable, and user-friendly assistant that can be tailored to the unique requirements of the Siemens Energy workforce. The assistant creates new opportunities for raising production and operational efficiency throughout the company and streamlining information management and retrieval procedures.

Implementing a microservice architecture, which lays the groundwork for a highly adaptable and scalable system, is one of the research's most important results. The architecture ensures that the assistant can adjust to changing business needs by seamlessly integrating new functionalities and decoupling various components and services. This is especially crucial for Siemens Energy, a multinational company that needs a flexible solution to manage multiple use cases in different divisions and locations.

The assistant can now deliver more precise and context-aware responses thanks to integrating advanced AI techniques like LLMs and RAG. With these technologies, the assistant can effectively compare papers, translate documents, answer complicated inquiries, and summarize vast amounts of material. Utilizing vector databases guarantees quick and pertinent data retrieval, which reduces engineers' time looking for information. This capability increases productivity and facilitates better decision-making by offering accurate and timely insights.

The backend implementation on AWS offers advantages in robust performance, cost-effectiveness, and scalability. The assistant's activities are supported by a dependable and effective infrastructure provided by AWS services, including EC2, Lambda, and S3. Because of its cloud-based architecture, the system is appropriate for enterprise-level applications and can manage high traffic and significant data volumes.

Future development will enhance the assistant's functionalities to incorporate more sophisticated AI-driven features like automated reporting, anomaly detection, and predictive analytics. These improvements, which offer more proactive decision-making tools and deeper insights, will significantly improve the system's capacity to support Siemens Energy's strategic goals. To better serve the broad and international workforce of Siemens Energy, future versions of the assistant will incorporate more sophisticated natural language understanding capabilities and enhance its multilingual support.

Future research should focus on integrating more advanced machine learning models and algorithms to enhance the performance and versatility of the assistant. The assistant can learn from user interactions and adjust to changing organizational demands by integrating continuous learning techniques. Improving the assistant's responses and features will entail investigating reinforcement learning strategies and considering user input.

Additionally, if the assistant develops further, it can be integrated with additional digital tools and platforms utilized by Siemens Energy. By establishing a more linked digital ecosystem, the assistant can offer extensive support and improve overall process efficiency and data accessibility. Creating APIs and interfaces to enable smooth communication between the assistant and other company systems, such as project management software, ERP, and CRM, will be necessary for this.

To sum up, Siemens Energy's creation of an AI-integrated virtual assistant is a big step toward using AI technologies to address practical business problems. The assistant's capacity to improve decision-making, optimize information retrieval, and accommodate a variety of use scenarios illustrates how AI has the potential to revolutionize business operations. Siemens Energy can fortify its competitive edge and foster innovation worldwide by further expanding upon this basis and investigating novel AI-powered solutions.

References

Beurer-Kellner, L., Fischer, M., & Vechev, M. (2022). Prompting is programming: a query language for large language models. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2212.06094

Chen, J., Lin, H., Han, X., & Sun, L. (2023). Benchmarking large language models in Retrieval-Augmented Generation. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2309.01431

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., & Wang, H. (2023). Retrieval-Augmented Generation for Large Language Models: A survey. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2312.10997

Guan, Y., Wang, D., Chu, Z., Wang, S., Ni, F., Song, R., Li, L., Gu, J., & Zhuang, C. (2023). Intelligent Virtual Assistants with LLM-based Process Automation. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2312.06677

Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., & McHardy, R. (2023). Challenges and applications of large language models. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2307.10169

Kirchenbauer, J., & Barns, C. (2024). Hallucination reduction in large language models with Retrieval-Augmented Generation using Wikipedia knowledge. OSF. https://doi.org/10.31219/osf.io/pv7r5

Li, Y., Wen, H., Wang, W., Li, X., Yuan, Y., Liu, G., Liu, J., Xu, W., Wang, X., Sun, Y., Kong, R., Wang, Y., Geng, H., Luan, J., Jin, X., Ye, Z., Xiong, G., Zhang, F., Li, X., . . . Liu, Y. (2024). Personal LLM Agents: Insights and Survey about the Capability, Efficiency and Security. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2401.05459

Liang, T., He, Z., Jiao, W., Wang, X., Wang, Y., Wang, R., Yang, Y., Tu, Z., & Shi, S. (2023). Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2305.19118

Ni, C., Wu, J., Wang, H., Lu, W., & Zhang, C. (2024). Enhancing Cloud-Based Large Language Model Processing with Elasticsearch and Transformer Models. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2403.00807

Shanahan, M. (2024). Talking about Large Language Models. Communications of the ACM, 67(2), 68–79. https://doi.org/10.1145/3624724

Topsakal, O., & Akinci, T. C. (2023). Creating large language model applications Utilizing LangChain: A primer on developing LLM apps fast. International Conference on Applied Engineering and Natural Sciences, 1(1), 1050–1056. https://doi.org/10.59287/icaens.1127

Von Straussenburg, A. F. A., & Wolters, A. (n.d.). Towards hybrid architectures: integrating large language models in informative chatbots. AIS Electronic Library (AISeL). https://aisel.aisnet.org/wi2023/9

Yu, P., Xu, H., Hu, X., & Deng, C. (2023). Leveraging Generative AI and large language Models: A Comprehensive Roadmap for Healthcare integration. Healthcare, 11(20), 2776. https://doi.org/10.3390/healthcare11202776

Zhang, L., Jijo, K., Setty, S., Chung, E., Javid, F., Vidra, N., & Clifford, T. (2024). Enhancing large language model performance to answer questions and extract information more accurately. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2402.01722

Zhu, Y., Yuan, H., Wang, S., Liu, J., Liu, W., Deng, C., Dou, Z., & Wen, J. (2023). Large Language models for information retrieval: a survey. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2308.07107

Zou, H., Zhao, Q., Bariah, L., Bennis, M., & Debbah, M. (2023). Wireless Multi-Agent Generative AI: From connected intelligence to collective intelligence. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2307.02757

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.