



PDF Download
3762669.pdf
30 March 2026
Total Citations: 2
Total Downloads: 891

Latest updates: <https://dl.acm.org/doi/10.1145/3762669>

RESEARCH-ARTICLE

RAG-LoE: IoT context-aware information retrieval with Large Language Models in Industry 5.0

MARCO ARAZZI, University of Pavia, Pavia, PV, Italy

MONICA MARCONI SCIARRONI, Marche Polytechnic University, Ancona, AN, Italy

ANTONINO NOCERA, University of Pavia, Pavia, PV, Italy

EMANUELE STORTI, Marche Polytechnic University, Ancona, AN, Italy

Open Access Support provided by:

University of Pavia

Marche Polytechnic University

Published: 18 November 2025

Online AM: 29 August 2025

Accepted: 06 August 2025

Revised: 31 May 2025

Received: 28 February 2025

[Citation in BibTeX format](#)

RAG-IoE: IoT context-aware information retrieval with Large Language Models in Industry 5.0

MARCO ARAZZI, Department of Electrical, Computer and Biomedical Engineering, Università di Pavia, Pavia, Italy

MONICA MARCONI SCIARRONI, Department of Information Engineering, Università Politecnica delle Marche, Ancona, Italy

ANTONINO NOCERA, Department of Electrical, Computer and Biomedical Engineering, Università di Pavia, Pavia, Italy

EMANUELE STORTI, Department of Information Engineering, Università Politecnica delle Marche, Ancona, Italy

Human-centric design, intelligence, and seamless interconnectivity are key pillars of the Industry 5.0. A critical challenge in these scenarios is the efficient retrieval of relevant, context-aware information for workers within Internet of Everything (IoE) networks. Traditional information retrieval techniques struggle with the heterogeneous, dynamic data generated in industrial settings. To address this, we define a context-aware data model for IoE scenarios, on top of which we propose RAG-IoE, a novel Retrieval-Augmented Generation (RAG) solution to enable adaptive, scalable, and context-based information retrieval from both structured and unstructured data sources. Our approach organizes IoE data within a semantic framework, integrating hybrid retrieval methods. It combines structured search on a Knowledge Graph with unstructured data retrieval using embeddings stored in a vector database, followed by LLM-driven reasoning to refine results. This architecture enhances decision-making, reduces cognitive overload, and ensures precise guidance for industrial operators. We validate the efficiency and effectiveness of RAG-IoE using a novel dataset through both a user study and quantitative analysis, demonstrating its potential to optimize human-machine collaboration in Industry 5.0 environments.

CCS Concepts: • **Information systems** → **Information retrieval query processing**; *Language models*; *Sensor networks*; *Graph-based database models*; • **Computing methodologies** → *Natural language generation*;

Additional Key Words and Phrases: Context-aware, knowledge graph, large language model, retrieval-augmented generation, IoT, IoE, Industry 5.0

This work has been partially supported by the PRIN 2022 project “HOMEY: a Human-centric IoE-based Framework for Supporting the Transition Towards Industry 5.0”, funded by the European Union - Next Generation EU, Mission 4 Component 1 (code: 2022NX7WKE, CUP: F53D23004340006).



Authors' Contact Information: Marco Arazzi, Department of Electrical, Computer and Biomedical Engineering, Università di Pavia, Pavia, Italy; e-mail: marco.arazzi01@universitadipavia.it; Monica Marconi Sciarroni, Department of Information Engineering, Università Politecnica delle Marche, Ancona, Italy; e-mail: monica.marconi@staff.univpm.it; Antonino Nocera, Department of Electrical, Computer and Biomedical Engineering, Università di Pavia, Pavia, Italy; e-mail: antonino.nocera@unipv.it; Emanuele Storti, Department of Information Engineering, Università Politecnica delle Marche, Ancona, Italy; e-mail: e.storti@univpm.it.



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

© 2025 Copyright held by the owner/author(s).

ACM 2577-6207/2025/11-ART23

<https://doi.org/10.1145/3762669>

ACM Reference Format:

Marco Arazzi, Monica Marconi Sciarroni, Antonino Nocera, and Emanuele Storti. 2025. RAG-IoE: IoT context-aware information retrieval with Large Language Models in Industry 5.0. *ACM Trans. Internet Things* 6, 4, Article 23 (November 2025), 31 pages. <https://doi.org/10.1145/3762669>

1 Introduction

Industry 5.0 represents a transformative shift from conventional industrial practices toward smart, sustainable and human-centric manufacturing systems [41, 62]. A key aspect is enhancing the social value of workers by emphasizing their interaction with digital technologies across various organizational levels and processes. This approach can ultimately contribute to increasing industries' resilience and sustainability, as also pointed out by the European Commission [13]. While its predecessor, Industry 4.0, was primarily focused on automation and interconnectivity, following a technology-driven trajectory, Industry 5.0 seeks to harmonize human intelligence with digital, intelligent platforms. This can be done leveraging advanced technologies such as the **Internet of Everything (IoE)** [35], which builds upon the foundational principles of Internet of Things, emphasizing the seamless integration of people, data, and devices into a unified ecosystem [50] and enabling context-aware decision-making within industrial environments. In particular, providing workers with the right contextual information at the right time is critical for achieving improved efficiency, reducing their cognitive overload and enabling them to be focused on the activity at hand.

The notion of "context" is however inherently dynamic and multidimensional, making its management complex. On the one hand, it is time-dependent, as workers may change their locations in the organization and the activities they have to perform. On the other hand, workers can collaborate with others to accomplish an activity, or delegate it to a colleague, potentially temporarily altering their role within the organization, and consequently the devices they are authorized to access. To make an example, suppose a maintenance technician is asked by a junior colleague to provide support in diagnosing an issue in the smart ventilation system of a factory room. As the collaboration is established, the access authorizations are properly transferred. As she enters the room, she needs the contextually-relevant information to accomplish the task, e.g., what sensors and devices are co-located in the room, on which she can operate according to the privileges, their technical specifications, the specific details of the activity to perform.

In order to support these scenarios, efficient retrieval and utilization of context-aware information is a critical challenge, but traditional techniques struggle to cope with the complexity and heterogeneity of data streams produced in Industry 5.0 environments. Here, diverse and heterogeneous entities such as smart sensors, autonomous systems, and human operators generate vast amounts of structured and unstructured data. In particular, unstructured data is harder to process, categorize, and integrate with other structured data, requiring the adoption of innovative retrieval mechanisms capable of dynamically adapting to evolving industrial scenarios, ensuring the privacy and security of information. In these scenarios, the integration of **Large Language Models (LLMs)** with their advanced **natural language processing (NLP)** capabilities, can play a crucial role in efficiently processing and analyzing data generated within IoE environments. Incorporating LLMs into context-aware IoE systems can significantly enhance how workers interact with digital technologies, streamlining the retrieval of relevant information and enabling more informed decision-making. Lightweight models are particularly valuable as they can be deployed on the edge of the IoE network.

To address these challenges, this article presents RAG-IoE, a novel approach based on **Retrieval-Augmented Generation (RAG)** [67] for context-based information retrieval in IoE networks.

The approach starts from a user context, and aims to retrieve those devices in the IoE that the worker can exploit to achieve a specific task. The framework integrates declarative knowledge representation, in the shape of a **Knowledge Graph (KG)** containing structured information of the IoE, and employs lightweight LLM-driven reasoning to create an adaptive and scalable retrieval mechanism. The search is performed in a hybrid fashion, by combining a search of structured and unstructured information in proper storage systems, and then exploiting an LLM for refining the results. The use of RAG is particularly effective in exploiting LLMs for vertical domains, as it incorporates domain-specific knowledge and focuses the model's attention on relevant, real-world data. Additionally, by filtering and contextualizing information, RAG reduces the likelihood of hallucinations, thereby ensuring more reliable and precise guidance for workers [63].

The contributions of this work are multifold:

- We propose a data model for an IoE network that not only captures static information about an agent, such as a worker or smart object, but also incorporates the concept of context. This context is inherently time-dependent and includes the dynamic information relevant to the agent's state and activities.
- We propose a RAG-based method to select the most suitable device for an agent based on user context. Structured data in a KG is used for pre-filtering through contextual reasoning. For unstructured data like technical specs, the system searches a vector database using embedded user context. The final prompt, combining context and search results, is sent to a lightweight LLM to identify the best device and provide usage guidance.
- We create a novel dataset tailored to Industry 5.0 scenarios, combining real-world sensor information and synthetic descriptions in natural language generated via LLMs, guided by manually curated examples to ensure realism and consistency. This dataset serves for testing and validating the proposed approach.
- We performed experiments aimed to validate efficiency and effectiveness of the approach. In order to evaluate the quality of the generated responses, we conducted a user study and a quantitative analysis. The latter is based on the RAGAs suite, which employs a dedicated LLM as an external evaluator. In both evaluations, an appropriate set of metrics was applied to provide a comprehensive assessment of the system's performance.

The rest of this article is structured as follows: Section 2 surveys related work on context models, semantic models for IoE scenarios and RAG for LLMs. In Section 3 we introduce the methodological principles of the approach. The data model of the framework, along with the notion of context, is presented in Section 4. Section 5 is devoted to discuss the context-aware RAG-based solution, while an evaluation in terms of efficiency and effectiveness is provided in Section 6. Finally, Section 7 concludes the work and outlines future work.

2 Related Work

In this section, we review related work relevant to our study, focusing on existing models for context representation, semantic model for Industry 5.0 and IoE, and the application of RAG in LLMs, highlighting its role in enhancing contextual understanding and improving knowledge retrieval for semantic reasoning.

2.1 Context Models

One of the first definitions of context is provided in [53] where authors mention three crucial dimensions: location of an agent, its social relations and nearby resources. Being focused mostly on location-based elements, context models were often represented through taxonomies. The necessity for a broader and more inclusive understanding of context was acknowledged later, such

as in [15], where context is articulated as any information pertinent to the interaction between a user and an application.

Research efforts have also focused on developing models and languages for designing contexts. Knowledge representation techniques seek to decouple context from specific applications, introducing specialized frameworks that treat context as a core component. Context representation models have been widely explored in the literature (see survey in [1]) and are often structured as sets of attribute-value pairs, e.g., [30]. Additionally, they can be implemented using markup languages [33] and formalized as context ontologies, which facilitate interoperability, reasoning, and dynamic adaptation, making context-aware computing more robust and scalable. An example is [45], in which the **Context Aware Sensor Configuration Model (CASCOM)** is proposed to simplify the process of configuring IoT middleware platforms, easing the retrieval process for data consumers, especially non-technical personnel. The SOCAM ontology [26] exploits a general-purpose upper ontology and application-specific lower ontologies representing entities such as person, location, computational entity and activity. In the IoT domain, [59] introduced CONON, a layered OWL ontology designed for context modeling. Its upper layer defines general context features based on objects and concepts, while the lower layers address domain-specific details. Similarly, [65] proposed a five-layered model for describing context-aware systems, where the lowest layer gathers raw sensor data, and the higher layers facilitate interoperability and integration of diverse sensors. Additionally, they developed a Web-Based Context Model Ontology to support semantic reasoning and context sharing across heterogeneous systems.

Most solutions do not consider context values at different granularity levels. Among the exceptions, in the Context Spaces [43] a context is represented as a point or region in a multidimensional space. The Context Dimension Tree hierarchical model, proposed in [54] for wireless sensor network applications, encodes simple values of the dimensional as leaf nodes, while structured values define a tree structure. In [18], a general model is proposed to represent context components at different granularities, enabling the definition of a multi-resolution context-based algorithm for querying the IoT, according to given preferences that can be tightened or relaxed depending on the given application goal.

2.2 Semantic Models for IoE

In the literature, a great effort has been devoted to defining semantic models to represent device characteristics and their relationships in the Internet of Things. A major problem faced by the research community in this context is related to the heterogeneity of IoT devices and their continuous evolution over time. One of the earliest approaches in this direction is described in [52], in which a general-purpose ontology called OntoSensor was proposed, starting from the Web Ontology Language and the **Suggested Upper Merged Ontology (SUMO)** [44]. Building once again from the SUMO ontology, the authors of [21] designed an initial ontology to retrieve all-and-only relevant sensor data, following an evolving prototype life-cycle. The **Semantic Sensor Network (SSN)** ontology [11] is recognized as one of the most prominent ontologies for IoT and domotics. Several work built lightweight semantic models on top of the SSN ontology, e.g., [6, 32]. In particular, the IoT-Lite ontology [6] builds a core model containing only the main concepts to support the most standard queries for IoT solutions. Originally proposed by the W3C SSN Incubator group, the SSN ontology has also been revised by [32] in the **Sensor, Observation, Sample, and Actuator (SOSA)** ontology. This proposal aims at a lightweight vocabulary including broader concepts with respect to the SSN ontology, with the idea to provide a core model that can be integrated and aligned to other specifications. Researchers also focused on modeling the environments in which sensors are deployed, which are important aspects for Industry 5.0. Examples include DogONT [7] for home environments, the **Building Topology Ontology (BOT)** [49] to support the exchange of

information related to building life-cycles, or the **Organization Ontology (ORG)** [12] modeling organizational structures and related information through the concepts of organizations, their actors, activities and roles.

Through the availability of domain specific ontologies, researchers and industries have started to adopt more and more solutions based on KGs to realize flexible and homogeneously integrated systems [29]. In the Internet of Things domain, the intrinsic heterogeneity of devices and standards requires the construction of suitable integration solutions to allow for a flexible and fruitful exploitation of the produced data and services. For instance, the work in [39] focuses on an Cyber-Physical Production System equipped with an **Industrial IoT (IIoT)** including also production and business processing data, while [17] proposes a Process-aware IIoT KG, where raw sensor data are enriched with information about process activities and the physical production environment.

Despite many studies have focused on the IoT and industry domains, the advent of the Industry 5.0 and the IoE paradigms introduce new challenges that only few approaches have addressed so far. In a previous work of ours, [3] we proposed SemIoE, an OWL2 ontology for the representation of information entities within an IoE ecosystem. The ontology aims to provide a structured and standardized way to describe smart objects, locations, agents and their roles, access rights and preferences, fostering interoperability and enhancing the semantic understanding of IoE environments. The ontology is used in the context of the HOMEY project to support data gathering from highly heterogeneous data sources, stream monitoring [55], data querying, along with entrusting role-based security mechanisms.

2.3 RAG

The recent breakthroughs in LLMs have brought about a transformative shift to the field of artificial intelligence [23]. These sophisticated models, trained on vast datasets and equipped with billions of parameters, exhibit remarkable capabilities in comprehending and generating human-like text. Their advancements have significantly enhanced various NLP tasks, including text generation and information retrieval [56]. By fine-tuning LLMs on domain-specific datasets, they can be effectively adapted to specialized applications, improving their performance in targeted tasks. In general, existing LLM architectures can be categorized into three primary types: Encoder-only, Decoder-only, and Encoder-Decoder models.

Encoder-only models, such as BERT [14] (Bidirectional Encoder Representations from Transformers), excel in tasks requiring deep language understanding, e.g., text classification, sentiment analysis, and semantic search. Built on the transformer encoder, they leverage bidirectional self-attention to analyze words in context, generating rich semantic representations. Decoder-only models, such as GPT [47] (Generative Pre-trained Transformer), are designed for generative tasks, utilizing the decoder component of the transformer architecture. They process input sequences unidirectionally, generating outputs one token at a time, with each prediction conditioned on previous tokens. This autoregressive approach enables coherent and contextually relevant text generation, making them ideal for applications like language modeling, dialogue generation, and summarization. While highly effective for open-ended generation, their lack of bidirectional attention limits their ability to fully understand input context, making them less suitable for tasks like semantic similarity or classification. Encoder-Decoder models, such as T5 [48] (Text-to-Text Transfer Transformer) and BART [36] (Bidirectional and Auto-Regressive Transformers), are highly versatile architectures designed for tasks requiring input transformation rather than simple continuation, e.g., machine translation, summarization, and text-to-text conversions, which are, however, computationally demanding. They combine an encoder, which processes input into a contextualized representation, with a decoder that generates output autoregressively.

Usually, LLMs are composed of a massive number of parameters, an example of this is GPT-4 which allegedly counts more than a trillion parameters. This imposes a challenge to run these models, but most importantly to fine-tune them to build task-specific chatbots. To overcome this it is preferred to use LLMs as few-shot learners [8, 68]. So, instead of fine-tuning or retraining it is more efficient to provide a context to the desired task in the prompt to allow the model to generate a more task-oriented answer. This type of strategy is also referred to as RAG [37], which uses an external source of knowledge to provide additional context in the LLM prompt.

In the recent research literature, a great effort has been devoted towards the definition of solutions based on RAGs in disparate application domains [10, 19, 28, 60]. For example, [19] exploits a RAG-based system to generate traffic scenarios to develop autonomous vehicles solutions. Unlike classical approaches that can only generate existing scenarios, this strategy would allow a more refined testing of autonomous vehicles against novel and complex traffic situations. Another example is presented in [10], in which the authors propose a Retrieval-Augmented Text-to-Image Generator (Re-Imagen) to generate high-quality images. In [28], the authors propose an interesting approach leveraging a RAG-based solution to manage textual graphs, supporting users to navigate the content of textual graphs through a “chatbot”.

Another important research direction in the context of RAGs focuses on exploiting RAG technology in conjunction with KGs. In this case, KGs can be used to retrieve information to enhance the context of LLMs and provide a natural answer to the operator that has to fulfill a task. For instance, the authors of [31] propose an explainable behavior prediction system for autonomous driving by integrating KGs and LLMs using RAG. Unlike traditional deep learning methods that rely solely on kinematic data, this approach leverages **Knowledge Graph Embeddings (KGE)** and Bayesian inference to enhance reasoning with real-time sensor inputs. Tested on pedestrian crossing and lane change prediction, it outperforms state-of-the-art methods in anticipation and F1 score, demonstrating its potential for safer, more interpretable autonomous systems. In [4], instead, the authors propose a system that enhances **Failure Mode and Effects Analysis (FMEA)** by integrating KGs with RAG to improve reasoning and factual accuracy. Traditional FMEA tools lack analytical depth, while LLMs struggle with factual knowledge. The proposed KG-enhanced RAG framework bridges this gap by enabling semantic question-answering on FMEA data, demonstrating improved retrieval recall and precision.

This strategy has also been applied to querying the KG itself. In [2], the authors present SparqLLM, a framework that enhances KG querying using RAG and LLMs. While KGs improve data integration and reasoning, their complexity limits usability for non-experts, especially in Industry 5.0. SparqLLM automates KG construction through an ETL pipeline and enables natural language queries by generating SPARQL queries with LLMs. By integrating template-based retrieval, it enhances accuracy and reduces semantic errors. A dynamic visualization dashboard further improves usability. Experimental evaluations show high query accuracy, robustness, and scalability, making SparqLLM an effective solution for semantic data access.

3 Methodology

In this section, we introduce the general description of RAG-IoE along with its main characteristics and challenges. In our scenario, we consider an Industry 5.0 where entities are interconnected through an IoE. As will be clearer in the next sections, our model revolves around the main concept of an *agent* of the industry, which can be either a human employee or an autonomous smart object, while legacy IoT devices (which are often also included in smart objects) provide the basic services to monitor and manage the factory. An IoT device can be either a sensor (i.e., a device capable of measuring some properties of interest, such as temperature, humidity, light) or an actuator (a device performing a naïve action, such as “open”, “close”, “turn on”).

The IoE provides the substrate to derive any information related to the factory activities and data; however, according to the Industry 5.0 manifesto, the factory must adapt itself to its agents' needs, thus implementing a user-centric view of the whole infrastructure. Indeed, the main idea behind our solution is that the IoE must be able to project a portion of the network based on the agent that is accessing its services. In practice, the user-centric view of the Industry 5.0 becomes an "ego-network" of the underlying IoE, which is the enabling infrastructure of the factory itself. For instance, at a given time during the working day, an agent can be located in the "Factory Floor 0" with the role "Quality Control Inspector" and is engaged in the activity "Inspecting fan motor" of a target unit located in that specific zone. Altogether, this information describes a specific situation in which the agent is involved in. Accordingly, the data access need of the agents may vary during its interaction with the network due to such specific situations, such as their physical location inside the factory boundaries, the activities, their role, and other information related to the industry's business model. For this reason, we define the concept of "context" of an agent, which describes the peculiar situation (location, activity, and so forth) in which an agent is involved at a precise timestamp.

The dynamic nature of the context and the complexity of the domain under investigation require the definition of an advanced information retrieval system, supporting the construction of the "ego-network" tailored to the specific context in which the agent operates. To build this system, we adopt a solution based on a KG modeling the concepts and relationships included in the considered scenario and allowing for the exact retrieval of information, locations, activity and systems. However, the context of an agent also includes the concept of the *current* activity, thus implicitly adding an additional constraint to the retrieval task: among all the systems returned by the KG, only those actually useful to complete the target activity should be considered. To fully address the complete retrieval task, a more sophisticated support semantic layer is needed, capable of mapping each system in the IoE to the various activities it can support. As will be clearer in the next sections, implementing such an exact mapping is not feasible in a complex scenario such as that of Industry 5.0. Therefore, we propose a strategy based on a combination of the aforementioned KG and a Retrieved-Augmented Generator capable of inferring the mappings between systems and activities by leveraging the knowledge of an LLM. We focus on lightweight models that can be deployed on the network edge, offering a cost-effective solution while also mitigating privacy concerns related to cloud-based APIs for LLMs.

4 Data Model

This section is devoted to introduce the model for agents in an Industry 5.0 and their contexts. In particular, in our solution, we focus on the following main entities:

- **Agent**: the main actor of the system. An agent can be either a human employee or a smart object capable of carrying out autonomous activities and with possible mobility capabilities.
- **Site**: a physical location in the building where the agents carry out their activities.
- **Activity**: a working task, included in a production process, that is assigned to some agent.
- **System**: an IoT component of the Industry; it can be either an IoT sensor or an actuator, and can belong to a smart object.

In our model, we denote by A the set of agents, L the set of sites, W the set of activities, and S the set of systems. An agent $a_i \in A$ is defined as a triple $a_i = \langle L_{a_i}, W_{a_i}, S_{a_i} \rangle$, where $L_{a_i} \subseteq L$ is the set of physical sites the agent is allowed to access inside the industry, $W_{a_i} \subseteq W$ is the set of activities the agent can perform, and $S_{a_i} \subseteq S$ is the set of systems that the agent can access. The set S_{a_i} may depend on different aspects based on the nature of the agent. In case of a human agent, it may be determined by the agent's role within a role-based organization or by delegation and collaboration

mechanisms that dynamically alter duties and responsibilities. Conversely, if the agent is a smart object, the set is defined by the object's capabilities.

We also introduce a notion of time t , which dictates the specific state in which the agent can be at a given moment. In particular, at each time t , the agent is associated with:

- A single location $l_{a_i}^t \in L_{a_i}$, representing the specific site where the agent is situated.
- A single (the current) activity $w_{a_i}^t \in W_{a_i}$, indicating the task the agent is performing at that time.
- A set of accessible systems $\bar{S}_{a_i} \subseteq S_{a_i}$, determined by the context of the agent defined later in the model.

A site $l_i \in L$ is a specific physical location of an industry. It contains information about the spatial coordinates and the shape of a marked zone. Additionally, sites have a hierarchical containment relationship, meaning that a site can contain other sites. In particular, we define a partial order relation \leq_L over L , where given two sites $l_i, l_j \in L$, $l_i \leq_L l_j$ specifies that l_i is physically *contained* inside l_j . In the following, we will use the symbol \leq_L to denote the direct order relation, and the notation \leq_L^* to explicit a transitivity closure of the partial order, i.e., $l_i \leq_L^* l_j$ if $\exists \{l'_1, \dots, l'_s\} \subseteq L : l_i \leq_L l'_1 \dots \leq_L l'_s \leq_L l_j$, for $s \geq 1$. In practice, this relation allows us to establish a spatial lattice, according to which larger sites (such as buildings or floors) contains smaller ones (e.g., rooms). For instance, given l_1 ="Manufacturing Building" (largest site), l_2 ="Building Floor 0" (inside the manufacturing building), l_3 ="Assembly Room" (inside Floor 0) and l_4 ="Room 101" (also inside Floor 0), then $l_2 \leq_L l_1$, $l_3 \leq_L l_2$ and $l_4 \leq_L l_2$. Therefore, by transitivity, $l_3 \leq_L^* l_1$, $l_4 \leq_L^* l_1$.

An activity $w_i \in W$ is the representation of an operational task inside a process that is assigned to an agent. Similarly to the sites, a partial order relation \leq_W is defined on W , to represent specialization between activities. Given two activities $w_i, w_j \in W$, the relation $w_i \leq_W w_j$ holds if w_j represents a more general or abstract activity, while w_i is a refined or more specific version of w_j . The structure induced by this partial order forms a lattice (W, \leq_W) . In the following, we denote by \leq_W a direct order relation, while we refer to \leq_W^* as the transitive closure of the partial order on W . We denote by $W_0 \subseteq W$ those activities that constitute the leaves of the lattice, i.e., $W_0 = \{w' \in W : \nexists w_k \in W, w_k \leq_W w'\}$. To give an example, if w_1 ="Perform maintenance on the ventilation system", w_2 ="Inspecting fan motor" and w_3 ="Measure fan motor RPM", then $w_3 \leq_W w_2 \leq_W w_1$ since w_1 is the most general, and w_2 is more general than w_3 . In this case, w_3 is a leaf in the lattice, as there are no more specialized activities defined. As said above, the assignment of an activity to an agent is a temporal notion and, as such, is related to the time t . Formally, let $f : T \rightarrow W$ be a function mapping each time instant to a specific activity the agent is carrying out, then $f(a_i, t) = w_{a_i}^t \in W_{a_i}$.

Finally, a system $s_i \in S$ represents a single IoT component of the more general IoE that interconnects the whole Industry 5.0. An IoT component can be either a sensor (a device with sensing capabilities) or an actuator (a device carrying out single and simple actions, such as, "open/close the lock", "start/stop the conveyor belt", "increase/decrease temperature", and so on). It is defined by a set of system capabilities, i.e., $s_i = \{c_1, \dots, c_m\}$, where $c_i \in W_0$. In other terms, a system capability is equivalent to a leaf in the lattice (W, \leq_W) , meaning that it provides basic support to the achievement of higher level activities, e.g., "Monitor fan motor RPM". In our scenario, a system can either be independent or included inside a smart object, i.e., an agent of the industry. For instance, an RPM sensor can be embedded within a smart ventilation smart object. In particular, we define a relation $\sigma \subseteq S \times A$ such that $(s_j, a_i) \in \sigma$ indicates that the system s_j is *embedded* within the agent a_i and, thus, the following condition holds:

$$(s_j, a_i) \in \sigma \Rightarrow \begin{cases} \forall t, l_{s_j}^t = l_{a_i}^t, & \text{i.e., the system shares the same location as the agent} \\ s_j \in S_{a_i}, & \text{i.e., the agent has access to the system it includes} \end{cases}$$

The context for an agent is defined by the information that changes over time, namely, the site where it is located and the activity that is being performed.

Definition 1. (Context of an agent) Given an agent $a_i \in A$, its context at time t is denoted by $\gamma_{a_i}^t = \langle l_{a_i}^t, w_{a_i}^t \rangle$, where $l_{a_i}^t \in L_{a_i}$ is the location of the agent, $w_{a_i}^t \in W_{a_i}$ is the activity being performed at time t .

On the other hand, the context of a system is defined by its current location.

Definition 2. (Context of a system) Given a system $s_j \in S$, its context at the time t is denoted by $\gamma_{s_j}^t = \langle l_{s_j}^t \rangle$, where $l_{s_j}^t \in L$ is the location of the system.

The context of a system is compatible with the context of an agent in case the agent's site is contained within the system's site and its current activity is compatible with the system's capabilities.

Definition 3. (Compatibility of contexts) Given an agent $a_i \in A$ with context $\gamma_{a_i}^t = \langle l_{a_i}^t, w_{a_i}^t \rangle$, and a system $s_j = \{c_1, \dots, c_m\} \in S$ with context $\gamma_{s_j}^t = \langle l_{s_j}^t \rangle$ at time t , $\gamma_{s_j}^t$ is compatible with $\gamma_{a_i}^t$ if and only if the following conditions hold:

- accessibility: $s_j \in S_{a_i}$, meaning that the system is accessible by the agent
- location compatibility: $l_{s_j}^t \preceq_L^* l_{a_i}^t$, meaning that the system's location is contained within the agent's location at time t ;
- activity compatibility: $\exists c_k \in \{c_1, \dots, c_m\}$ such that $c_k \preceq_W^* w_{a_i}^t$, meaning at least one system capability is a specialization of the agent's current activity.

For example, consider an agent a_i , a maintenance technician, located in $l_{a_i}^t = \text{"Factory Floor 0"}$ at time t , performing the activity $w_{a_i}^t = \text{"Inspecting fan motor"}$. Now, assume a vibration monitoring system s_j is located in "Room 101" and has the capability $c = \text{"Measure fan motor RPM"}$. Since "Room 101" is within "Factory Floor 0", we have $l_{s_j}^t \preceq_L^* l_{a_i}^t$. Furthermore, "Measure fan motor RPM" is a specialization of "Inspecting fan motor", meaning $c_k \preceq_W^* w_{a_i}^t$. Since both conditions are met, the system's context is compatible with the agent's context at time t .

5 Context-aware RAG

This section is devoted to discuss the context-aware RAG solution for retrieval of information from an IoE network. Based on the defined data model, we formulate the problem statement as follows: given an agent $a_i \in A$ with context $\gamma_{a_i}^t$ at time t , find the most relevant set of systems $\bar{S}_{a_i} \subseteq S$ whose contexts are compatible with $\gamma_{a_i}^t$.

To address this problem, we rely on enterprise information systems, which serve as a primary source of data. These systems maintain various types of information that contribute to defining both static properties and dynamic aspects of the context. On the one hand, the hierarchical structuring of locations is derived from the topology of the enterprise, which remains relatively stable over time, ensuring that spatial containment relationships between sites are well-defined. Similarly, the technical capabilities of systems can be extracted from technical documentation, which is often stored in **Product Lifecycle Management (PLM)** systems or asset management platforms. While some specifications are structured (e.g., metadata attributes such as supported parameters, resolution, and measurement range), others are provided in unstructured text (e.g., maintenance manuals and operational guidelines). On the other hand, contextual information can be obtained directly from real-time monitoring systems. For instance, the location of human agents can be inferred from access control systems (e.g., badge logs or indoor positioning systems), while current activities can be retrieved from workforce scheduling and task management software.

However, some information, such as the complete enumeration of possible activities and their hierarchical relationships (i.e., the partial order relation \leq_W), are constantly evolving and difficult to capture from traditional information systems. These relationships may be only partially structured and tracked, and may emerge from multi-fold sources, such as operational workflows, human decision-making, internal best practices or, ex-post, from historical activity logs. Exhaustive approaches have been explored in the literature, e.g., in [16] where an explicit definition of goals, subgoals, and tasks within an Ambient Assisted Living scenario is provided. In that framework, the relationships between a system and the tasks it supports are fully elicited, resulting in a detailed and comprehensive model. While such a level of completeness is feasible in small-scale, controlled environments, it becomes impractical in large-scale scenarios such as Industry 5.0, where hundreds of agents, systems, activities interact dynamically, and the number of possible relationships grows exponentially. Being a complete, predefined enumeration of all dependencies unmanageable, a more scalable and adaptable approach is proposed. In the following, we discuss the platform architecture (Section 5.1), detailing how the KG (Section 5.2), the vector store and the RAG solution (Section 5.3) interact to provide a context-aware solution for the problem at hand.

5.1 Platform Architecture

Given the complexity and diversity of the information involved, a hybrid solution is required to effectively handle both structured and unstructured data, as well as dynamic and evolving information. *Structured data*, such as agent details and location, assigned activities, technical structured system specifications and locations, can be encoded using data and knowledge representation models. In contrast, *unstructured* or *evolving data*, such as technical documentation of systems, often written in natural language, require a different approach, as a purely declarative model struggles to accommodate the specificity of these formats. In particular, we leverage:

- A KG as a flexible model for capturing semantical relationships and enhancing interoperability among heterogeneous definitions. Relationships between locations, agents, activities and systems can be easily captured and maintained in this structured format, ensuring efficient storage, querying and access to data.
- A vector database that uses embeddings to represent unstructured descriptions. As such, by converting textual descriptions into high-dimensional vector representations, we can perform semantic retrieval in a manner that aligns with the conceptual meaning of the descriptions rather than relying solely on keyword matching. This enables the system to understand and match activities to systems based on their semantic similarity, rather than strictly predefined terms.

Based on the KG and the vector database, we devised a context-aware, RAG-based solution, an overview of which is presented in Figure 1. The architecture includes the following modules:

- *Context definition*: The process begins with the identification of the agent’s context, according to the model defined in Section 4 (step 1).
- *Retrieval*: Given the context, the module is in charge of retrieving the information needed to construct the agent’s “ego-network”, through a hybrid approach. This operation is performed in two steps:
 - *Pre-filtering*: An exact retrieval is aimed to pre-filter the possible results. It is performed by querying the KG using some of the contextual information, namely the location, to extract the list of systems accessible by the agent (step 2).
 - *Embedding*: The activity assigned to the agent is extracted, once again, from the context and an approximate retrieval is conducted with the purpose to obtain a short-list of candidate solutions. This step is performed by comparing the embedding of the activity description

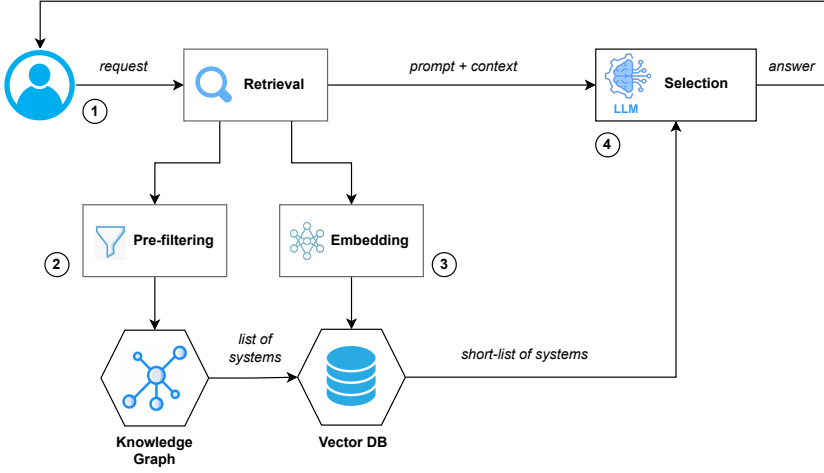


Fig. 1. Overview of the context-aware RAG solution for IoE networks.

ALGORITHM 1: Context-aware RAG solution

Input: agent a_i , **Output:** tuple $\langle \bar{S}_{a_i}, report \rangle$

- 1: $S_{a_i} \leftarrow \text{pre-filter}(a_i, l_{a_i}^t)$
 - 2: $q_{w_{a_i}^t} \leftarrow \text{formulate_query}(w_{a_i}^t)$
 - 3: $E_w \leftarrow \text{embed}(q_{w_{a_i}^t})$
 - 4: $scores \leftarrow []$
 - 5: **for each** $s \in S_{a_i}$ **do**
 - 6: $E_s \leftarrow \text{extract_embedding}(s)$
 - 7: $scores.append(E_s \cdot E_w)$
 - 8: **end for**
 - 9: $S_{a_i}^k \leftarrow \text{get_top_k}(S_{a_i}, scores)$
 - 10: $docs \leftarrow \text{retrieve_docs}(S_{a_i}^k)$
 - 11: $P \leftarrow \text{write_prompt}(S_{a_i}^k, docs, S_{a_i}, q_{w_{a_i}^t})$
 - 12: $\langle \bar{S}_{a_i}, report \rangle \leftarrow \text{submit_LLM}(P)$
 - 13: **return** $\langle \bar{S}_{a_i}, report \rangle$
-

with the embeddings of the systems descriptions stored in a vector database. The systems obtaining the highest relevance scores are selected (step 3).

- *Selection:* Finally, the textual descriptions of the previously identified most relevant systems are passed to an LLM together with the activity description, with the goal to identify the correct mapping between the target activity and the most adequate systems that the agent can leverage to complete it (step 4).

The following subsections detail the retrieval phase of the RAG solution, which are formalized in Algorithm 1. In Section 5.2, we describe the exact retrieval approach aimed at pre-filtering the solutions based on structured information (Figure 1, step 2), while Section 5.3 explains how embeddings are applied to unstructured data for vector database search and how this information is used for the final selection (Figure 1, steps 3–4).

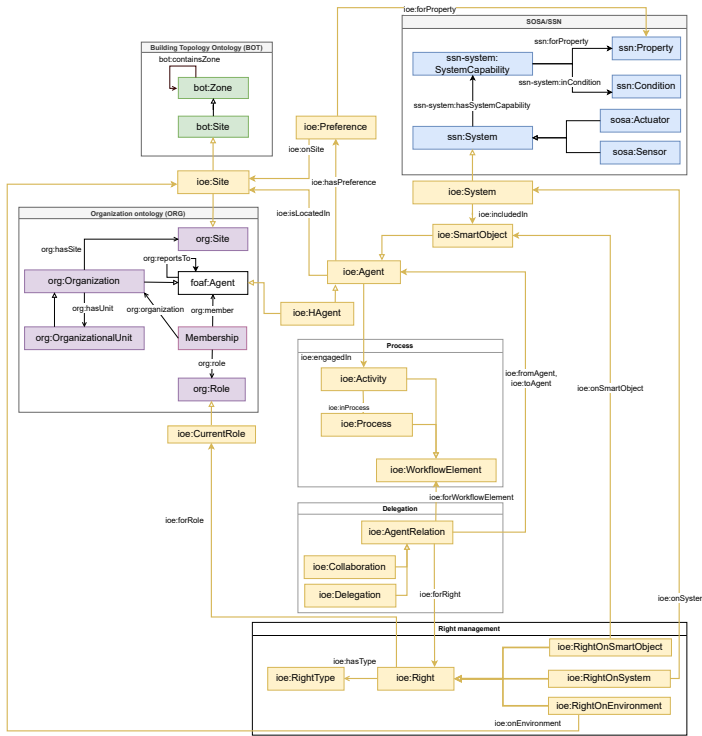


Fig. 2. An overview of the SemIoE ontology diagram.

5.2 Exact Retrieval

Structured information is stored in a KG, which relies on the SemIoE ontology [3] as its schema. SemIoE is a lightweight OWL2 ontology designed to provide a structured and standardized framework to describe entities and their relationships for an Industry 5.0 scenario. The ontology offers a semantic layer representing agents, systems, environments, processes, rights and preferences and defining their interconnections within an IoE network. This helps to enhance the semantic understanding of IoE environments, fostering interoperability across heterogeneous IoE components. As shown in the diagram in Figure 2, the ontology reuses and integrates several external modules that cover specialized aspects of the IoE landscape. Notably, SemIoE incorporates the W3C SSN ontology [11], which serves as the foundation for defining technical characteristics and functionalities of sensors and actuators (prefix *ssn* in figure).

We summarize the classes most relevant for the purpose of this work in the following (with prefix *ioe* in figure), highlighting their relation with the data model definitions provided in Section 4:

- *Site* represents locations within the organization boundaries and corresponds to the set L in the model. The class extends *bot:Site* from the Building Topology Ontology [49] (prefix *bot* in figure), which in turn is a specialization of *bot:Zone* and inherits the relation *bot:containsZone*. This last corresponds to the relation \leq_L in the model, as it enables to define a partial order between specific sites.
- *Activity* represents an activity, e.g., “maintenance of the ventilation system”. As such, it corresponds to the set W in the model.

- *Agent* represents either the class of *HAgents*, i.e., human agents, or *Smart Objects*. In the model it is denoted by *A*. It is located in a *Site* and can be involved in an *Activity*.
- A Smart Object is composed by one or more *Systems*, e.g., a humidity sensor or a damper actuator for ventilation, which correspond to the set *S* in the model. The inclusion relation σ is represented through the property *includedIn*. According to the SSN ontology, each system is characterized by a set of technical *Properties*, which correspond to the system capabilities.
- An Agent can access a number of systems, based on its *Role*. This is represented through a set of rights enabling read/right access to specific systems, or to an entire smart object, and systems within, or to all systems within an environment. As such, it allows to represent the set of accessible systems S_{a_i} for an agent a_i .

The SemIoE ontology includes further classes and relations, e.g., to represent relations among agents or preferences, which are not fully reported here. The full specification of the ontology is available at <https://w3id.org/semioe>.

The KG is serialized as RDF triples, which are stored in a Graph DBMS ensuring efficient management and querying of information. Additionally, we exploit an ontology reasoner to infer implicit knowledge based on logical axioms defined in the ontology. Specifically, the reasoner allows us to compute the logical consequences of transitive object properties, such as *bot:containsZone*, which encodes the partial order relation \preceq_L defined over sites. To give an example, consider three sites l_1, l_2, l_3 such that $l_3 \preceq_L l_2 \preceq_L l_1$. In the KG this hierarchy would be represented by two triples: $l_1 \text{ bot:containsZone } l_2$, and $l_2 \text{ bot:containsZone } l_3$. By exploiting the transitivity of the object property, the reasoner can infer the additional relationship $l_3 \preceq_L^* l_1$, and materialize a new triple $l_1 \text{ bot:containsZone } l_3$ in the graph. This materialization extends the graph with previously hidden knowledge, reducing the need for recursive query and making query answering more efficient.

5.2.1 Pre-filtering. Information retrieval from the KG can be done through the SPARQL language [25], a query language for RDF graphs acknowledged by W3C as an official recommendation. Starting from the location of an agent, as an instance of class *Site*, the SPARQL query reported in Listing 1 aims to retrieve the set of systems which are located in the specified site and that the agent can access. This is checked by extracting the agent's role and identifying the corresponding rights.

```

PREFIX ioe: <http://www.w3id.org/semioe#>
PREFIX org: <http://www.w3.org/ns/org#>
PREFIX ex: <http://mydomain/example/>

SELECT ?s
WHERE {
  ?s a ioe:System;
     ioe:includedIn ?so.
  ?so a ioe:SmartObject;
     ioe:locatedIn <site>.

  ?m org:member <agent>;
     org:role ?r.
  ?ri a ioe:Right;
     ioe:forRole ?r.

  ?ri ?p ?o
  FILTER ((?p = ioe:onSystem && ?o = ?s) ||
          (?p = ioe:onSmartObject && ?o = ?so) ||
          (?p = ioe:onEnvironment && ?o = <site>))
}

```

Listing 1. SPARQL query for retrieval of systems located in a <site> and accessible by an <agent>

As a result, the sensors satisfying the condition of being co-located with a given agent, and accessible by the agent are retrieved (Algorithm 1, line 1). By exploiting logical reasoning, the approach works also for indirect location relations, e.g., such that a system is located in a site which is not identical to the agent's site, but it is contained within it.

An important novelty introduced by our approach, leveraging the model defined by the SemIoE ontology, is the capability to adapt the context of an agent in response to the specific nature of an activity, whether it originates from the agent's own responsibilities, a collaborative request from another agent, or a delegation due to the unavailability or constraints of another agent. This enables more flexible and context-aware activity management. In particular, according to the SemIoE model, the set of systems accessible to an agent by the role can be dynamically extended through temporary privileges granted via delegation/collaboration strategies defined by the *ioe:AgentRelation* concept. This mechanism enables agents to access additional resources when acting on behalf of others under specific relational contexts. To simplify the presentation, in the following, we will refer to all such relations collectively as delegations.

In this context, the set of systems accessible to an agent S_{a_i} , can be defined as the union of two disjoint subsets:

$$S_{a_i} = S_{a_i}^o \cup S_{a_i}^d$$

where $S_{a_i}^o$ represents the set of systems the agent has direct access to based on their role in the industry; whilst, $S_{a_i}^d$ denotes the set of systems that the agent can temporarily access through delegation mechanisms, as defined by the *ioe:AgentRelation* concept in the SemIoE ontology. In particular, $S_{a_i}^d$ is built starting from active delegation relations that involve a_i . More formally, a delegation can be modeled as a tuple $\langle a_x, a_i \rangle$ in which a_x is the *requesting* agent of the delegation/collaboration and a_i is the agent *receiving* it. Now, let $D_{a_i} = \{\langle a_x, a_i \rangle \mid a_x \in A\}$ be the set of active delegation received by a_i , and let $P(\langle a_x, a_i \rangle)$ be the set of privileges granted to a_i by one of these delegations. $S_{a_i}^d$ can, hence, be obtained as follows:

$$S_{a_i}^d = \bigcup_{\langle a_x, a_i \rangle \in D_{a_i}} P(\langle a_x, a_i \rangle)$$

5.3 Approximate Retrieval and Selection

As discussed in the previous section, through a declarative model, we could efficiently store and retrieve relationships between agents and their activities, locations, and systems. However, a fundamental piece of information for constructing an agent's "ego-network", projected from the underlying IoE based on the agent's context (see Section 3), is the association between the agent's activities and the most suitable systems for executing them. Given the complexity of the Industry 5.0's scenario, building the exact mapping between system capabilities and activities requirements remains impractical. Therefore, in the following, we describe an approximate retrieval mechanism that leverages the power of LLMs to generate these mappings dynamically. Hence, while the exact search enables to obtain a set of accessible systems that can possibly span the whole IoE, the role of this step is to filter-out those systems that are not good candidates to carry out the target activity.

Recently, RAG has been successfully employed to build efficient retrieval mechanisms from rich and sparse collections of textual documents [4, 37]. The main idea behind these solutions is that user queries should be answered by the LLM model by using chunks of information, suitably retrieved from a vast collection of available textual documents, as source of knowledge. The LLM must infer the link between a user query and the knowledge included within textual chunks, through its capability of modeling and elaborating the natural language.

In our scenario, systems' capabilities can be described through textual documents; think, for instance, of the unstructured texts (maintenance manuals and guidelines), which are typically associated with each system's model. Such unstructured texts represent, in our case, the chunks of information written in natural language. The user query, on the other hand, is derived from the description of the activities in which the agent is involved. Recall that, given the context of an agent a_i at a given time t , it is possible to identify a single activity, say $w_{a_i}^t$, the agent is involved in.

In our solution, we assume that a detailed description of $w_{a_i}^t$ is available in the system (e.g., through task management software). With these premises, we can define the user query $q_{w_{a_i}^t}$ starting from the description of $w_{a_i}^t$ (Algorithm 1, line 2); for instance, if its description is “Inspecting fan motor”, then we can formulate the user query for the RAG as “Which are the most adequate systems to carry out the activity of inspecting fan motor?”. To answer this query, we have to perform the following steps, starting from the list of candidate systems obtained from the Graph:

- Extract the systems’ descriptions from a suitable repository.
- Narrow down the list of systems to a controlled number such that we include only the ones with the highest probability of being useful to complete the target activity.
- Build a prompt for the underlying LLM including the filtered list of systems, the original query, and a “chain of thought” it has to follow to maximize the correctness of the produced answer. The knowledge encoded in the LLM (thanks to the huge datasets exploited for its training) enables it to accurately identify the correct mappings between systems and the activities they support.

A critical design aspect to discuss in our proposed solution involves the selection of LLMs and the corresponding computational architecture. Unlike what is typically done in the related literature [24], due to the peculiarity of our application scenario, we adopt a hybrid centralized/decentralized approach based on edge computing. In our solution, we attempt to deploy smaller LLMs at the network edge rather than relying on a centralized infrastructure, which would allow the deployment of larger LLM models. This architectural choice is motivated by several factors. First, a centralized deployment that executes the approximate retrieval for all involved agents would require substantial computational resources, including high-performance GPUs and large-scale data centers, which may be impractical in many (smaller) industrial settings. Second, edge computing enables localized inference elaborating a limited amount of data only referring to a target agent. Third, and most importantly in the context of Industry 5.0, a local deployment enhances privacy and data sovereignty by ensuring that sensitive operational data related to an individual (human employee) remain local to the agent, enabling subsequent secure interactions between the agent and the LLM during the activity execution. This reduces the risks of privacy leakage typically associated with data aggregation and centralized processing. It is important to underline that this strategy aligns with the human-centric principles of Industry 5.0, emphasizing autonomy, transparency, and ethical AI deployment. As a final remark, we observe that the recent literature is beginning to consider decentralized approaches to build collaborative and more scalable RAGs [46, 61]. Moreover, some authors suggest that, potentially, small LLMs suitably supported by a knowledge base and a RAG mechanism can return performance similar, if not better, to large LLM [40]. Due to the peculiarity of the considered scenario, we then adopt a hybrid strategy, according to which the knowledge base (exact retrieval, see Section 5.2) is centrally stored in the industry main computing unit, while the deployment of the LLM-based approximate retrieval component is at the edge. This approach allows us to inherit the advantages of classical centralized knowledge bases while also addressing privacy and scalability concerns.

5.3.1 Embedding. The first step is performed by a common component of a RAG, which is a “vector database”, denoted by VS , containing the embeddings of the text chunks related to the technical documentation describing the capabilities of each system model available in the IoE. These embeddings are dense vector representations of the text chunks, capturing their semantic meaning, generated using a pre-trained language model and stored in VS . They can be built through any lightweight language model, such as NV-Embed-v2 [34] or bge-en-icl [9]. Let $E(s_i)$ be the embedding of the technical description of the capabilities of the system $s_i \in S$ and let $E(q_{w_{a_i}^t})$ be the

embedding of the user query (built using the same strategy adopted for $E(s_i)$), we can apply the following strategy to narrow down the list of systems with the highest probability of being useful for the activity $w_{a_i}^t$. For each system $s_i \in S_{a_i}$, we extract the corresponding description in VS and compute the following score (Algorithm 1, lines 3–8):

$$score_{s_i} = E(s_i) \cdot E(q_{w_{a_i}^t})$$

where (\cdot) represents the dot product between the two vectors. Observe that, if the embeddings are normalized (i.e., the magnitude is 1), then this metric also coincides with the cosine similarity of the two vectors. In general, any other similarity metric can be adopted in our scenario. The only requirement is that it must be capable of capturing the alignment of the two vectors in the embedding space. In this way, a high score indicates that the two embeddings represent “near” semantic concepts.

After obtaining the scores for each system, S_{a_i} can be sorted accordingly (Algorithm 1, line 9). We impose that only the k -top systems $S_{a_i}^k$ are selected as candidates for the final elaboration done by the LLM, which will return the set \overline{S}_{a_i} .

5.3.2 Selection. As a final step, the filtered set of systems is used to build a prompt to an LLM (Algorithm 1, lines 10–11). In the design of the prompt, we adopt a Chain-of-thought prompting strategy to guide the LLM in the elaboration of $S_{a_i}^k$ to answer the query $q_{w_{a_i}^t}$ [66]. In practice, we guide the LLM to add a reasoning step, thus altering the sequence $LLM(q_{w_{a_i}^t}) \rightarrow O$ to $LLM(q_{w_{a_i}^t}) \rightarrow R \rightarrow O$, where O is the expected answer and R represents some intermediary reasoning steps. The reasoning steps are typically introduced by explicitly adding in the query to the LLM a sentence such as “Let’s think step by step” [66] and enlisting the basic required reasoning steps (zero-shots); moreover they can be further improved by including suitable controlled examples (few-shot). In our approach, we engineered the LLM prompt by adopting the following structure:

- **Define the LLM role.** In this step, we specify the objective of the LLM by asking it to pretend to act as an intelligent chatbot designed to assist operators in a smart factory.
- **Define the input.** We describe the available input organized in three components:
 - Activity description in natural language;
 - list of available systems. These represents all the systems the operator has access to;
 - list of relevant systems for the task (based on the embedding phase).
- **Define the output.** We describe how the output must be organized and returned in three information pieces:
 - identify key systems for the input task ($O1$);
 - explain how to process and integrate data from the selected systems ($O2$);
 - provide actionable steps for the operator ($O3$).

Formally, the output $O(O1, O2, O3)$ is built as a sequence of the three steps above obtained upon the starting input. More details about the adopted prompting strategy are reported in Section 6.3. Finally, the prompt is submitted to the LLM and the output is returned to the user (Algorithm 1, lines 12–13).

6 Evaluation

In this section, we demonstrate the effectiveness of our solution through an experimental campaign. Datasets and code are available on the project repository.¹

¹<https://github.com/Homey-Prin22/RAG-IoE/>

6.1 Dataset Generation

As explained in Section 5.3, the proposed framework utilizes a KG and a vector store respectively populated with structured and natural language descriptions of sensors. Therefore, we require a dataset that provides both the sensors and their corresponding descriptions. To the best of our knowledge, no existing dataset fully meets our requirements. Therefore, we have created and proposed a new dataset suitable for activities related to Industry 5.0 and NLP. To build our dataset, we adopted a hybrid approach combining real-world data sources and synthetic data generation. Initially, we collected a set of existing and manually curated sensor devices, approximately 50 unique devices, from different manufacturers and categories, to ensure a diverse representation of sensor types commonly used in industrial systems. In particular, we included Environmental, Proximity and Object Detection, Pressure, Temperature, Vibration, Flow, Gas, Position, Energy, and Power sensors. For a subset of these devices, we manually collected and verified descriptions from authoritative sources, such as technical datasheets, product manuals, and official manufacturer websites. These data served as examples to generate contextually consistent descriptions for the remaining sensors. We leveraged a LLM, specifically *LLaMa2-7B* [57], providing the manually curated descriptions as examples to produce realistic datasheet-like descriptions, consistent with the verified ones. Here is an example of a generated description:

The Inductive Proximity Sensor detects metal objects within its range by measuring changes in the magnetic field caused by the object's presence. It features a highly sensitive coil that can detect even small changes in the magnetic field, making it ideal for detecting thin or distant metallic objects. The sensor has an adjustable sensitivity setting that allows users to customize the detection range based on their specific application requirements. Additionally, the sensor has a fast response time, making it suitable for high-speed applications where rapid detection is crucial.

Ranges of sensitivity and sensitivity:

- * Sensitivity range: 0-5 mm (adjustable)
- * Detection range: 0-100 mm (fixed)
- * Resolution: 0.1 mm (typical)
- * Response time: <1 ms (typical)

In summary, the Inductive Proximity Sensor provides accurate and reliable metal object detection with adjustable sensitivity settings and fast response times. Its highly sensitive coil and fixed detection range make it suitable for a wide range of industrial applications, while its compact size and easy integration make it a convenient choice for many users.

Thanks to this approach, we were able to create coherent descriptions even for those sensors that do not exist in commercial catalogs. This allowed us to build a richer and more diverse dataset, which would have been difficult to obtain relying solely on manual collection. Structured information was stored in a KG consisting of 141,042 RDF triples, representing sites, agents, smart objects and included systems, activities, roles and rights. The following key statistics summarize its composition:

- (1) 50 distinct *ioe:Site* instances.
- (2) 100 unique human *ioe:Agent* entities;
- (3) 1,000 *ioe:SmartObjects* belonging to 50 types. Each smart object includes 5 systems, resulting in 5,000 systems;
- (4) on average, each *ioe:Site* contains approximately 20 smart objects.

These statistics provide a quantitative overview of the factory's digital representation, offering insight into the scale and structure of the modeled environment.

Regarding the assessment, the evaluation dataset was manually curated to include three distinct sets of activities, categorized according to their complexity. Specifically, the three categories are defined as follows:

- (1) **Easy:** it involves a single machine and a set of sensors of the same type (such as temperature sensors), e.g.,:

“Continuously measure and control the temperature inside the system responsible for melting iron for casting. Adjust heating elements based on real-time sensor readings to maintain optimal conditions.”
- (2) **Medium:** the activity involves a single machine and sensors of different types, e.g.,:

“Verify the correct alignment of automated shelving systems in the warehouse to ensure smooth operation and prevent mechanical failures. A high-precision displacement sensor installed on the shelving tracks detects misalignments or structural deviations. If any irregularities are found, corrective actions can be taken to avoid disruptions in the storage and retrieval process.”
- (3) **Hard:** the activity involves at least two machines and multiple sensors of different types, e.g.,:

“Continuously monitor fire hazards in a chemical plant by detecting flammable gases, abnormal temperature fluctuations, and critical pressure levels. The fire safety supervisor analyzes real-time sensor data to assess potential risks and determine necessary actions.”

In particular, the activities dataset is composed as follows:

- (1) 16 different activities divided in the 3 categories in different contexts: 6 Easy Activities, 5 Medium Activities and 5 Hard Activities.
- (2) We defined 16 contexts, each including a random location and one of the 16 activities. We configured the KG to always return 5 relevant smart objects from the pre-filtering, each equipped with 5 sensor instances. The results returned by the Graph deliberately include both relevant smart objects aligned with the activity requirements and random selections, in order to evaluate the model’s ability to correctly identify the appropriate devices. Sensors may possess identical functionalities or feature multiple instances of the same sensor model. This setup is also employed to assess the model’s capacity to avoid generating false positive answers.

All the datasets involved in this study are included in the GitHub repository linked above.

6.2 Experimental Setup

In order to conduct the experiments, we developed our system utilizing the following technologies. For the KG, we relied on GraphDB 10.2.2. The RAG solution incorporates Chroma as the vector store candidate, utilizing *cosine similarity* as the standard similarity metric and *all-MiniLM-L6-v2* as the model to generate the stored embeddings along with the device-model identifiers of the sensors. We intentionally selected a widely used, general-purpose embedding model to demonstrate that our approach is not dependent on a specific embedding model, thereby highlighting its generalizability. To generate the final response, we utilized various LLMs to evaluate the quality of the responses generated under various conditions and the model’s dimensions to assess also the scalability and the minimum required hardware to run the solution. The candidate LLMs used in our experiments are: *DeepSeek-R1-Distill-Qwen-1.5B*, *DeepSeek-R1-Distill-LLaMa-8B*, and *DeepSeek-R1-Distill-Qwen-14B* [27] to evaluate various inference-specialized model sizes, each demanding distinct computational resources. To validate the generalizability of our method, we also included two general-purpose LLMs aligned with the hardware constraints of our setup *LLaMa-3.2-1B* [20] and *Qwen2.5-3B* [5]. The selected models have a relatively limited number of parameters because the proposed system is designed to assist operators directly on-site. Running the LLM on edge devices ensures real-time responsiveness, minimizing latency and enabling seamless interaction in

industrial environments where immediate decision-making is crucial. Approximately, the included models require from 2.5 GB to a maximum of 25 GB of system memory.

To evaluate our system we leveraged the generated dataset described in Section 6.1. In particular, we used the generated sensors' description to populate the vector store, and we used the set of activities to prompt the system.

To assess the quality of the answers generated by the proposed system, we employed a user-study to present the main results combined with additional metrics from the RAGAs [22] suite. This suite provides a set of metrics specifically designed for evaluating RAGs, utilizing an additional LLM as a judge (*Zephyr-7B* [58]).

In the user-study we defined the following metrics:

- (1) *Hallucinations*: This is a boolean metric used to indicate whether the response is classified as containing a hallucination (true) or free from hallucinations (false). A hallucination occurs when the model generates information that is factually incorrect or unsupported by the available data.
- (2) *Accuracy*: This metric quantifies the correctness of the generated responses. A high accuracy score means the information is reliable and free of errors.
- (3) *Relevance*: This metric looks at how well a response addresses the specific query and stays on topic. A highly relevant response directly answers the question, incorporates the necessary context, and avoids unnecessary information.
- (4) *Usefulness*: This metric considers the practical value of the response because even if it is correct, it may still be unhelpful if it lacks actionable insights. It assesses whether the information provided can be applied effectively in a real-world scenario.
- (5) *General Subject Score*: This metric is an evaluation of overall performance.

In particular, from the RAGAs suite, we employed: Faithfulness, Answer Relevancy, Answer Correctness, Context Precision, and Context Recall.

- (1) *Faithfulness*: The Faithfulness metric evaluates how factually aligned a response is with the given context. The metric varies from 0 to 1, where higher values denote greater consistency.

$$\text{Faithfulness Score} = \frac{\text{Number of claims in the response supported by the retrieved context}}{\text{Total number of claims in the response}}. \quad (1)$$

- (2) *Answer Relevancy*: The Answer Relevancy metric evaluates how well a response aligns with the user's query. A higher score reflects greater correspondence with the user input, whereas a lower score indicates responses that are either incomplete or contain superfluous details. To calculate this metric the suite generates a set of possible questions from the obtained responses and compares them with the original prompts.

$$\text{Answer Relevancy} = \frac{1}{N} \sum_{g=1}^N \frac{E_g \cdot E_u}{\|E_g\| \|E_u\|}, \quad (2)$$

where N is the total of the generated questions, E_u is the user input and E_g is a generated question.

- (3) *Answer Correctness*: Evaluating Answer Correctness determines how accurately a generated response aligns with the ground truth. This process involves comparing the response against the reference, with resulting scores between 0 and 1. A score closer to 1 suggests a stronger correspondence between the generated reply and the ground truth, indicating enhanced correctness. Answer correctness covers two essential dimensions: the semantic resemblance between the generated answer and the ground truth, in addition to factual consistency.

- (4) *Context Precision*: It quantifies the ratio of pertinent chunks found in the retrieved contexts. It is determined by averaging the precision@k across each chunk within the context. Precision@k expresses the proportion of relevant chunks at rank k compared to the total number of chunks at that same rank.

$$\text{Precision@k} = \frac{\text{true positives@k}}{\text{true positives@k} + \text{false positives@k}}, \quad (3)$$

$$\text{Context Precision@K} = \frac{\sum_{k=1}^K (\text{Precision@k} \times v_k)}{\text{Total number of relevant items in the top K results}}, \quad (4)$$

where K is the total number of chunks in retrieved contexts and $v_k = \{0, 1\}$ is the relevance indicator at rank k .

- (5) *Context Recall*: It quantifies the proportion of relevant documents (or pieces of information) that have been successfully retrieved. It emphasizes minimizing missed important results. A higher recall indicates that fewer relevant documents were overlooked. In essence, recall ensures that nothing important is left out. Since it focuses on completeness, calculating context recall always requires a reference for comparison.

$$\text{Context Recall} = \frac{\text{Number of claims in the answer supported by the retrieved context}}{\text{Total number of claims in the answer}}. \quad (5)$$

The implementation for the evaluation consists of two machines: the KG on GraphDB was running on a workstation with Intel Xeon Gold 6252N CPU @ 2.30 GHz, 4 cores, 16 GB RAM, operated by Rocky Linux 9.4. The vector DB and the LLM were running on a workstation equipped with an I9 14900KF CPU, 2xRTX 4090 GPUs and 128 GB of RAM. The software has been developed using Python and the PyTorch Framework, version 2.5.1.

6.3 Prompting Strategy

As we already stated in Section 5.3, the prompting strategy is fundamental for the generation of useful answers for the operator. The use of a context is critical to instruct operators in the right way, instead of just relying only on the knowledge possessed by the LLM. In addition to providing a context from an external and verified source, the way the model is prompted also affects the final outcome. Building on this, we employ a Chain-of-thought approach, guiding the LLM through an example activity and outlining the necessary steps to derive possible guidelines for its successful completion. An example of the prompt is presented as follows:

```

Role:
You are an intelligent chatbot designed to assist operators in solving tasks within a
smart factory. Your goal is to provide clear, actionable guidance based on the
available machinery, sensors, and automation systems.

Operator's Request Format:
Operators will ask for help with specific tasks, such as:

- Task Example: Continuously monitor and regulate the pressure inside the hydraulic
press to ensure optimal forming of metal components.

Available Context:

Factory Environment: {task_example} (List of available machines, sensors, and control
systems)
Relevant Sensors: {sensors_example} (A filtered list of sensors most relevant to the
task)

Response Structure:

```

- 1 Identify Key Machinery & Sensors - Describe the machinery involved and the specific sensors available for monitoring and control.
- 2 Explain the Process & Integration - Detail how the sensors collect data, how it's processed (e.g., via PLC, SCADA, or IoT gateways), and how adjustments are made.
- 3 Provide Actionable Steps for the Operator - Guide the operator on what to check, how to validate readings, and any troubleshooting steps.

Example Answer:

```
``You can achieve precise pressure control using your existing setup. The press_1,
press_2, press_3, press_4, and press_5 sensors provide real-time pressure readings,
processed by the Siemens S7-1200 PLC via Modbus. Data is displayed on SCADA (WinCC
) and HMI, with alerts for deviations. Pressure adjustments are automated using the
control system, with a PID loop ensuring stability. Regularly check press_3 for
proper fluid flow monitoring and press_1 and press_4 for hydraulic pressure
consistency. Ensure sensor calibration, monitor system response, and verify SCADA
thresholds to maintain optimal forming conditions.''
```

Final Task:

```
Based on this task description {task_description} and the factory environment {env}, use
the provided sensor descriptions {docs} to generate a short operator-friendly and
direct step-by-step guide on what to check and how to ensure optimal system
performance.
```

In the prompt, *task_description* represents the activity that the operator needs to complete, *env* denotes the operator's ego-network as provided by the KG, and *docs* refers to the retrieved documents that form the contextual information. We provide the LLM with the operator's entire ego-network to evaluate its resilience in generating meaningful responses, even in extreme cases where no sufficiently similar *docs* are retrieved. Additionally, this ensures the model can produce relevant answers in situations where the ego-network includes similar machinery, but the activity involves a specific one with a distinct role in the given environment (e.g., ovens designed for processing different metals). This is an example of an answer returned by our system for the following activity:

“Detect the presence of metallic objects moving along the automated transport system. Ensure proper sorting by identifying specific materials using proximity sensors, and redirecting items to the correct processing area.”

This is the obtained answer:

```
Ensure that the automated transport system correctly identifies and sorts various types of
materials, particularly metals, using proximity sensors.
```

```
Step 1: Identify Key Machinery and Sensors
Step 2: Explain the Process \& Integration
Step 3: Provide Actionable Steps for the Operator
```

Now, I'll begin the thought process.

```
To solve the task, first, identify the key machinery involved in the automated transport
system. Looking at the factory context, there are Conveyor Belt 1 and Conveyor Belt 2.
These conveyors transport raw materials and finished products respectively. However,
the task specifically mentions ensuring that the transport system correctly identifies
and sorts materials, particularly metals. Therefore, Conveyor Belt 1, which transports
raw materials, is more relevant here since it deals with the initial transportation of
materials that may include metals.
```

```
Next, look at the sensors available for each conveyor belt. For Conveyor Belt 1, the sensors
listed are:
```

- proximity_1 (Xtron E2E)
- proximity_2 (Mechtron BAW)
- photo_1 (LumeTech PZ Series)
- vib_1 (Vibrarion 810)
- air_1 (Techsense HPMA115S0)

```
Among these, the proximity sensors (proximity\_1 and proximity\_2) are directly related to
detecting metal objects because they are designed to sense metallic materials. The
Techsense HPMA115S0 measures particulate matter, which could indicate the presence of
metal particles or dust, but it's less direct than the proximity sensors for large
```

metallic objects. The Vibrarion 810 vibrational sensor might detect issues like imbalanced loads, but it's more about machine health rather than material identification.

So, the key sensors for this task would be proximity_1 and proximity_2, which are both designed to detect metal objects.

Step 2: Explaining the Process \& Integration

The proximity sensors detect the presence of metal objects on the conveyor belt. These sensors send their data to a control system, likely a PLC or SCADA system, which processes the information. The system then determines the type of material based on the sensor readings. Depending on the detected material, the conveyor system can sort the items into different processing areas. For instance, if the system detects metal objects, it can divert them to a specific area designated for metal processing. Adjustments can be made by the control system to optimize the sorting process, perhaps through algorithms that analyze the sensor data over time to improve accuracy.

Step 3: Providing Actionable Steps for the Operator

1. **Regular Calibration:** Periodically calibrate proximity sensors to ensure accurate metal detection.
2. **Monitoring:** Use the control system to observe conveyor operation and verify correct sorting of materials.
3. **Troubleshooting:**
 - Check sensor alignment and adjust if necessary.
 - Review and update control logic to address any discrepancies.
 - Inspect the conveyor belt for obstructions or debris affecting sensor performance.
4. **Maintenance:** Keep conveyor belts clean to avoid interference and ensure smooth operations.
5. **Documentation:** Maintain records of sensor performance and system adjustments for future reference.

By following these steps, the operator ensures optimal performance of the automated transport system, efficiently sorting metals and other materials.

6.4 User Study Results

This section presents the main performance results of our model, derived from a user study. In this study, two annotators evaluated the responses generated for each activity by the three candidate models, assessing them based on the metrics defined in Section 6.2. Evaluations have then been averaged. In a highly complex domain such as the one explored in this article, conducting a thorough human evaluation is essential. This allows us to capture the diverse ways a given activity can be approached and solved, considering the nuances that an automated system might overlook. Furthermore, human assessment is crucial for verifying not only the correctness but also the practical usefulness of the answers generated by our proposed system. By incorporating expert judgment, we can better understand the model's strengths and limitations, ensuring that it provides valuable support in real-world industrial settings. In particular, we aim to evaluate whether the deployment of a relatively small LLM on edge devices can still generate satisfactory and useful responses for an operator in a real-world scenario. Given the constraints of edge computing, it is essential to verify that the model can effectively process queries and provide meaningful assistance despite its limited size.

The hallucination metric plays a crucial role in filtering responses before evaluation. The presence of hallucinated content in a response is determined based on the expertise of the annotators. If a response is marked as containing hallucinations (true), it is excluded from the assessment of the four key metrics: accuracy, relevance, usefulness and general subject score. The results of the user study reveal that 21% of the responses contained hallucinated content. Since these responses were not factually reliable, they were excluded from the overall final performance analysis. In terms of evaluation of metrics, each is rated on a scale from 1 to 5, with higher scores indicating better performance, and using the value 0 for hallucinated content which was not evaluated. We assess not

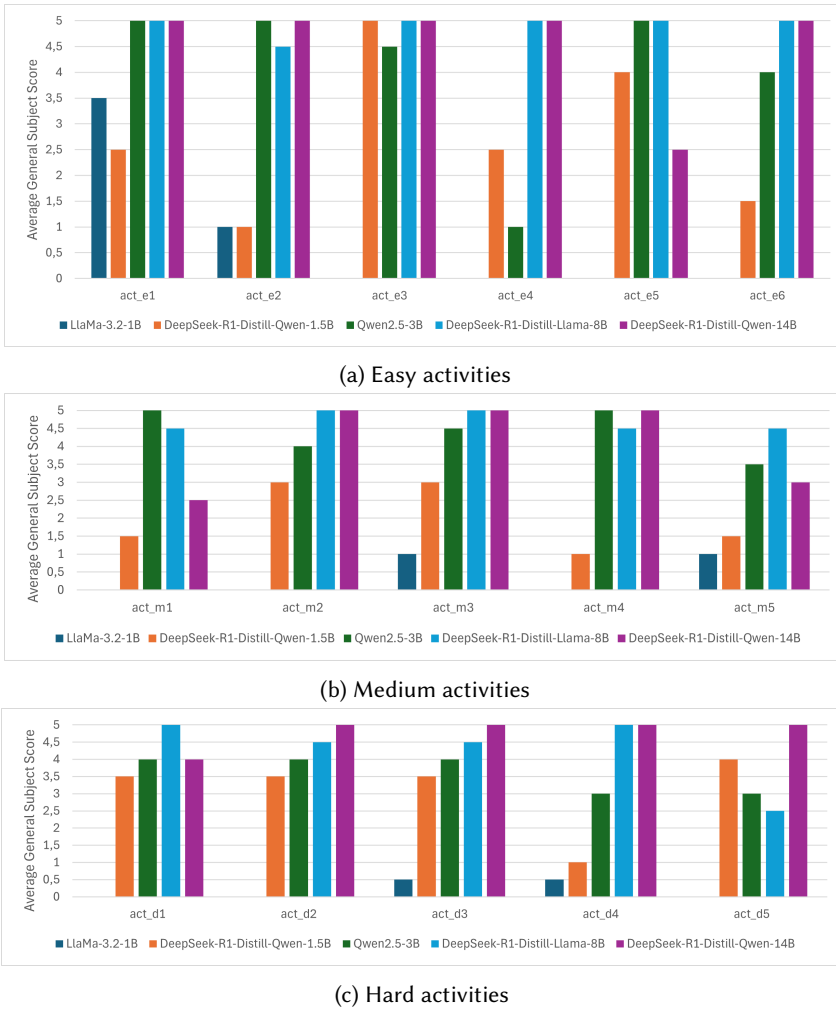


Fig. 3. General subject score evaluation of activities with different difficulty levels.

only the accuracy and relevance of the generated responses but also their coherence and adaptability to the complex and dynamic nature of the industrial environment. Most importantly, we want to test the ability of the model to extract useful insights from the retrieved context, reason over potentially incomplete or ambiguous information, and provide guidance tailored to the operator's activity of particular interest. Additionally, we aim to investigate whether the tradeoff between model size and responsiveness maintains a level of performance that ensures practical usability in real-time decision-making scenarios.

The results about General Subjective Score, which provides an overall assessment of the models' performance, by consolidating the individual scores of accuracy, relevance and usefulness, are reported in Figure 3(a)–(c).

As observed, the complexity of the considered domain and context poses significant challenges for the smallest two of the evaluated models, making it struggle to generate high-quality responses. Most hallucinations are produced by these models and highlighted with a value 0. Especially LLaMa

Table 1. Evaluation Scores on Easy Activities

		act_e1	act_e2	act_e3	act_e4	act_e5	act_e6
LlaMa-3.2-1B	Accuracy	5	1,5	0	0	0	0
	Relevance	2,5	1	0	0	0	0
	Usefulness	3	1	0	0	0	0
DeepSeek-R1-Distill-Qwen-1.5B	Accuracy	3,5	3,5	5	3	4	1,5
	Relevance	2	3	5	2,5	4	1,5
	Usefulness	2	3	5	2,5	4	1,5
Qwen2.5-3B	Accuracy	5	4,5	4,5	0,5	5	3,5
	Relevance	4,5	4,5	4,5	0,5	5	3,5
	Usefulness	5	5	5	1	5	4
DeepSeek-R1-Distill-Llama-8B	Accuracy	5	4,5	5	5	5	5
	Relevance	4,5	5	5	5	5	5
	Usefulness	5	5	5	5	5	5
DeepSeek-R1-Distill-Qwen-14B	Accuracy	5	5	5	5	3	5
	Relevance	5	5	5	5	2,5	5
	Usefulness	5	5	5	5	2,5	5

tends to produce usable results only when handling relatively simple activities, in our case just one of the easy tasks has been answered successfully. Upon closer inspection of its failures, we notice that the model either fails to provide an answer or repeats the examples provided in the Chain-of-thought within the prompt, indicating difficulties in adapting to novel or more intricate scenarios.

Examining the performance of the other two models, we observed consistently strong results across various scenarios, with only a few exceptions. Analyzing these anomalous cases, we found that, particularly for more complex activities, the models tend to generate lengthy and repetitive responses. Instead of providing a concrete solution, they simply list the available machinery and sensors without effectively leveraging the retrieved context. Furthermore, when suggesting possible guided steps of action, they often fail to guide the operator toward the specific machinery and sensors using their unique identifiers, making it more challenging to interpret the response.

Examining the successful cases, we observed that the models provided structured guidelines for solving the activities, often showing a form of reasoning in their responses. In particular, they often first enumerate the available machinery and then justify their selection based on the specific requirements of the activity. Additionally, in some instances, the models went beyond our initial expectations by suggesting the use of supplementary machinery that, while not originally deemed essential in our curated dataset of activities, could still prove beneficial in completing the operation.

Tables 1–3 show detailed evaluation of the five models, Llama-3.2-1B, DeepSeek-R1-Distill-Qwen-1.5B, Qwen2.5-3B, DeepSeek-R1-Distill-Llama-8B, and DeepSeek-R1-Distill-Qwen-14B, across activities of varying difficulty, highlighting clear differences and performance patterns in accuracy, relevance, and usefulness. Note that accuracy refers to the correctness of the output, relevance indicates the contextual appropriateness, and usefulness is about the practical applicability to the intended purpose.

Regarding the performance in easy activities, DeepSeek-R1-Distill-Llama-8B consistently achieves the highest scores in all metrics, demonstrating strong accuracy (>4.5), high relevance, and excellent usefulness. DeepSeek-R1-Distill-Qwen-14B closely follows, showing excellent performance in most activities but it could have a drop in accuracy and usefulness in easy activities (see act_e5), indicating sensitivity to certain input types despite its larger size. Qwen2.5-3B performs quite well, though it shows some fluctuations in accuracy and relevance, particularly in

Table 2. Evaluation Scores on Medium Activities

		act_m1	act_m2	act_m3	act_m4	act_m5
LlaMa-3.2-1B	Accuracy	0	0	1,5	0	1
	Relevance	0	0	1,5	0	0,5
	Usefulness	0	0	1	0	0,5
DeepSeek-R1-Distill-Qwen-1.5B	Accuracy	1,5	3,5	3	1	1,5
	Relevance	1,5	3	3	1	1,5
	Usefulness	1,5	3	3	1	1,5
Qwen2.5-3B	Accuracy	5	4	5	5	3,5
	Relevance	4,5	3	5	5	3,5
	Usefulness	5	3	4,5	4,5	3
DeepSeek-R1-Distill-Llama-8B	Accuracy	5	5	5	4,5	4,5
	Relevance	4,5	5	5	5	5
	Usefulness	5	5	5	5	4,5
DeepSeek-R1-Distill-Qwen-14B	Accuracy	3	5	5	5	3
	Relevance	3	5	5	5	3
	Usefulness	2,5	5	5	5	3

Table 3. Evaluation Scores on Hard Activities

		act_d1	act_d2	act_d3	act_d4	act_d5
LlaMa-3.2-1B	Accuracy	0	0	0,5	0,5	0
	Relevance	0	0	0,5	0,5	0
	Usefulness	0	0	0,5	0,5	0
DeepSeek-R1-Distill-Qwen-1.5B	Accuracy	3,5	4	3,5	1	4
	Relevance	3,5	4	3,5	1	4
	Usefulness	3,5	3,5	3,5	1	3,5
Qwen2.5-3B	Accuracy	5	4	4,5	4	3
	Relevance	4,5	4	4	3	3,5
	Usefulness	4	4	4,5	2,5	2,5
DeepSeek-R1-Distill-Llama-8B	Accuracy	5	4	4,5	5	3
	Relevance	5	4,5	4	4,5	3
	Usefulness	5	4,5	4	5	2
DeepSeek-R1-Distill-Qwen-14B	Accuracy	4	5	5	5	5
	Relevance	4	5	5	5	5
	Usefulness	4	5	5	5	5

specific cases where accuracy drops, suggesting challenges with certain types of input scenarios. DeepSeek-R1-Distill-Qwen-1.5B displays inconsistent results, scoring well on some tasks but failing to generalize across all examples, reflecting its limitations at lower scale. Llama-3.2-1B struggles significantly, providing non-informative or zero-value responses, failing to provide meaningful outputs.

In the medium-difficulty activities, DeepSeek-R1-Distill-Llama-8B maintains near-optimal performance across all tasks, confirming its ability to handle moderately complex scenarios with consistent accuracy and relevance. DeepSeek-R1-Distill-Qwen-14B is still among the top performers, but it shows inconsistency compared to its 8B version. This is also partially related to the process of training through distillation from different models, LLaMa for 8B and Qwen for the 14B,

which affect the overall behavior of the model. Qwen2.5-3B remains competitive, though with some degradation in accuracy and a slight drop in usefulness, implying that its answers may be correct but not fully address the user's intent. DeepSeek-R1-Distill-Qwen-1.5B struggles more clearly here, while Llama-3.2-1B experiences a severe decline in accuracy, scoring zero in several tasks, with minimal relevance and limited usefulness. This suggests that the model struggles with reasoning beyond simpler prompts.

When it comes to hard activities, Llama-3.2-1B fails to generalize effectively, scoring near-zero in most tasks and only marginal improvements in a few cases; Qwen2.5-3B struggles more in these cases, showing a decrease in both usefulness and relevance, which suggests it has difficulties in generating fully coherent responses under complex constraints. DeepSeek-R1-Distill-LLaMa-8B, on the other hand, still outperforms the other simpler models, but with a noticeable decline in performance for more challenging activities. This is due to the increase in the number of parameters being monitored and the level of reasoning involved. Simple activities typically rely on a single sensor input (e.g., detecting metallic objects with a proximity sensor, monitoring humidity levels in a storage environment or measuring vibration levels during precision cutting of metal parts). In contrast, hard activities require the integration of multiple sensors (e.g., CNC process monitoring involves tracking spindle speed, cutting force, vibration, and temperature simultaneously, or fire hazard detection requires analyzing flammable gases concentrations, temperature fluctuations, and critical pressure levels). This increased complexity makes it more challenging to identify the optimal set of sensors required to complete the activity, which in turn affects the models' performance. In this particular context, instead, DeepSeek-R1-Distill-Qwen-14B shows its advantage in terms of number of parameters, allowing it to be more effective in reasoning on a larger set of machines and sensors that cooperate together in the same scenario, showing almost flawless performance.

These results highlight the robustness of larger models in handling diverse activity complexities, while also emphasizing the limitations of smaller models, particularly when dealing with more challenging reasoning and structured input scenarios. Through this experiment, we have demonstrated that leveraging context from both the vector store and, more importantly, the operator's ego-network enables even relatively small models to generate meaningful and useful responses for factory operators. This approach enhances the model's ability to provide relevant guidance, even with limited computational resources.

6.5 Ragas Metrics Results

As described in Section 6.2, we also conducted an experimental evaluation employing the well-known RAGAs suite, which utilizes an extra LLM model, in our instance *Zephyr-7B*, to serve as a judge assessing the quality of a RAG's answer. As proved in [69], smaller size judges, like the one employed in this experiment, achieve comparable or higher agreement with GPT-4 than human judges on test benches like MT-Bench and AlpacaEval. This creates an opportunity for industries to assess the new integrated solution without risking the exposure of sensitive data, by leveraging models that are only accessible through the major players in the LLMs domain. As noted in various other studies, assessing RAG in highly specialized and intricate domains and activities, such as those discussed in this article, may produce outcomes that appear suboptimal [42, 51]. Despite our awareness of this potential outcome, we opted to present these results as a benchmark for future research. Specifically, we concentrated on the top-performing models from Section 6.4: *DeepSeek-1.5B*, *Qwen2.5-3B*, *DeepSeek-8B*, and *DeepSeek-14B*.

The results of this analysis are reported in Table 4. In addition, we also measure the Contextual Precision and Recall of the retrieved documents, scoring, respectively, 0.5 and 0.63.

As expected, the results obtained from the RAGAs evaluation are not as optimal as those produced by human annotators, who possess a deeper understanding of the complex context under

Table 4. Evaluation with RAGAs Metrics

Metric	DeepSeek-R1-1.5B	Qwen2.5-3B	DeepSeek-R1-8B	DeepSeek-R1-14B
Faithfulness	0.43	0.65	0.71	0.71
Answer Relevancy	0.47	0.61	0.70	0.72
Answer Correctness	0.53	0.60	0.52	0.69

Table 5. Time Analysis on the Components of the Proposed Solution

Component	KG Retrieval	ChromaDB	LLama-3.2-1B	DeepSeek-R1-1.5B	Qwen2.5-3B	DeepSeek-R1-8B	DeepSeek-R1-14B
Time (sec)	0.75	<0.01	1.78	3.2	3.64	6.57	22.59

consideration. Human evaluators can better grasp nuances, implicit relationships, and domain-specific intricacies that the automated system may overlook. However, despite these limitations, the results remain encouraging. They demonstrate the system’s potential and provide a valuable baseline for future advancements in RAG technologies within this domain. These findings highlight areas for improvement while also validating the feasibility of using automated evaluation methods to assess the quality of generated responses.

6.6 Time Analysis on the System

This section focuses on evaluating the time required for our system to generate a response for the operator. In a real-world setting, it is crucial that the operator receives a timely response to avoid delays or disruptions of the activity.

To evaluate query time on the KG, an agent and its corresponding site have been randomly selected. Then, the query reported in Section 5.2 was executed 10 times and the average execution time was recorded. The procedure has been repeated for five different agents and final results were averaged. As for the vector store and the LLMs, a run for each of the 16 considered contexts was executed in terms of search on the vector db and prompt to the LLM. Execution times have then been averaged. Results are reported in Table 5.

As expected, the execution of this model is the most impacting on the overall solution, since it is the most computationally intensive component of the system. This is particularly evident in DeepSeek models, which, in their standard configuration, tend to offer verbose but at the same time more detailed responses as confirmed by past experiments, but they also take longer to produce the answers, showing in the 14B version an upper bound also in terms of time of response that could be considered not feasible for the timely completion of the activity. In recent years, significant advancements have been made in enabling the deployment of AI models of this scale on mobile chips, as highlighted in [38, 64]. Ongoing research in TinyML for LLMs and improvements in processor capabilities are continuously pushing the boundaries, making it increasingly feasible to run these models efficiently on edge and mobile devices.

7 Conclusion

In the context of Industry 5.0, providing human-centric design and access to services and information available within the firm boundaries is one of the defining aspects of this new paradigm. In this article, we deal with a scenario in which Industry 5.0 technological aspects are organized in the form of an IoE, which provides connectivity and interoperability among all the entities involved. In this scenario, the definition of human-centric projections of such a complex ecosystem requires the identification of the context of the target users. For this reason, we started by proposing a theoretical model for representing the information in the IoE and, then, we formalized the notion of user context and compatibility between contexts. After that, we defined a solution to build a human-centric and context-aware view of the industry, at least from the data access stand point. The

proposed solution is based on a hybrid approach combining a declarative model with an advanced information extraction strategy based on an RAG solution. In our approach, through the RAG we generate mappings between activities, assigned to the users, and the systems (smart sensors, actuators, and so forth) available within the industry. This advanced retrieval is based on textual descriptions written in natural language, which are available for both the activities and the systems. The experimental campaign was conducted using a novel manually-curated dataset, combining real-world sensor information and corresponding LLM-generated natural language descriptions. We demonstrate the effectiveness of our solution, which achieves high retrieval quality by leveraging lightweight LLMs, also ensuring relatively short execution times. However, the evaluation does not include direct comparisons with existing RAG-based or graph-based methods. This is primarily due to the unique hybrid nature of our architecture, which integrates a KG as a pre-filtering mechanism rather than as a model for document representation. Adapting standard baselines to our setting, such as removing the KG and instead providing the entire background knowledge directly in the prompt, would require significant architectural changes and may be technically infeasible due to prompt size limitation. Nonetheless, we plan to explore this direction in future work to further highlight the distinct advantages of our framework.

The research described in this article should not be considered an endpoint in this field. In fact, we plan to extend it in different directions. Firstly, once the retrieval of the mapping is complete, we believe that users would benefit from dynamic and adaptive visualization strategies for the information generated by the systems. This additional requirement could also be faced using the power of LLMs. Indeed, recent works in the literature have shown the possibility of exploiting LLMs to generate interesting images from information retrieved from target sources (e.g., Retrieval-Augmented Text-to-Image Generator). We believe that these approaches could be extended to generate suitable analytics dashboards showing data from the IoE. Moreover, we are studying the possibility of not only generating the mappings between activities and systems, but also to identify possible collaborations between agents. In particular, this strategy should also consider information about the expertise of the users, the complexity of the activities, and the physical/mental status of the involved agents (in case of human employees) or their load level (in case of smart objects).

References

- [1] Unai Alegre, Juan Carlos Augusto, and Tony Clark. 2016. Engineering context-aware systems and applications: A survey. *Journal of Systems and Software* 117 (2016), 55–83.
- [2] Marco Arazzi, Davide Ligari, Serena Nicolazzo, and Antonino Nocera. 2025. Augmented knowledge graph querying leveraging LLMs. arXiv:2502.01298. Retrieved from <https://arxiv.org/abs/2502.01298>
- [3] Marco Arazzi, Antonino Nocera, and Emanuele Storti. 2024. The SemIoE ontology: A semantic model solution for an IoE-Based industry. *IEEE Internet of Things Journal* 11, 24 (2024), 40376–40387. DOI: <https://doi.org/10.1109/JIOT.2024.3452945>
- [4] Lukas Bahr, Christoph Wehner, Judith Wewerka, José Bittencourt, Ute Schmid, and Rüdiger Daub. 2025. Knowledge graph enhanced retrieval-augmented generation for failure mode and effects analysis. *Journal of Industrial Information Integration* 45 (2025), 100807.
- [5] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. arXiv:2309.16609. Retrieved from <https://arxiv.org/abs/2309.16609>
- [6] Maria Bermudez-Edo, Tarek Elsaleh, Payam Barnaghi, and Kerry Taylor. 2016. IoT-Lite: A lightweight semantic model for the Internet of Things. In *Proceedings of the 2016 INTEL IEEE Conferences on Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress*. IEEE, 90–97.
- [7] Dario Bonino and Fulvio Corno. 2008. Dogont-ontology modeling for intelligent domotic environments. In *Proceedings of the International Semantic Web Conference*. Springer, 790–803.
- [8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems* 33 (2020), 1877–1901.

- [9] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE M3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics, 2318–2335.
- [10] Wenhui Chen, Hexiang Hu, Chitwan Saharia, and William W. Cohen. 2022. Re-Imagen: Retrieval-augmented text-to-image generator. In *Proceedings of the 11th International Conference on Learning Representations*.
- [11] Michael Compton, Payam Barnaghi, Luis Bermudez, Raúl García-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, Vincent Huang, Krzysztof Janowicz, W. David Kelsey, Danh Le Phuoc, Laurent Lefort, Myriam Leggieri, Holger Neuhaus, Andriy Nikolov, Kevin Page, Alexandre Passant, Amit Sheth, and Kerry Taylor. 2012. The SSN ontology of the W3C semantic sensor network incubator group. *Journal of Web Semantics* 17 (2012), 25–32.
- [12] World Wide Web Consortium et al. 2014. *The Organization Ontology*. Technical Report. World Wide Web Consortium.
- [13] J. Cotta, M. Breque, L. De Nul, and A. Petridis. 2021. Towards a sustainable, human-centric and resilient European industry. *Luxembourg: Publications Office of the European Union* (2021).
- [14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.
- [15] A. K. Dey. 2001. Understanding and using context. *Personal Ubiquitous Computing* 5, 1 (2001), 4–7.
- [16] Claudia Diamantini, Alessandro Freddi, Sauro Longhi, Domenico Potena, and Emanuele Storti. 2016. A goal-oriented, ontology-based methodology to support the design of AAL environments. *Expert Systems with Applications* 64 (2016), 117–131.
- [17] Claudia Diamantini, Alex Mircoli, Domenico Potena, and Emanuele Storti. 2023. Process-aware IIoT knowledge graph: A semantic model for industrial IoT integration and analytics. *Future Generation Computer Systems* 139 (2023), 224–238.
- [18] Claudia Diamantini, Antonino Nocera, Domenico Potena, Emanuele Storti, and Domenico Ursino. 2020. Querying the IoT using multiresolution contexts. *IEEE Internet of Things Journal* 8, 7 (2020), 6127–6139.
- [19] Wenhao Ding, Yulong Cao, Ding Zhao, Chaowei Xiao, and Marco Pavone. 2024. Realgen: Retrieval augmented generation for controllable traffic scenarios. In *Proceedings of the European Conference on Computer Vision*. Springer, 93–110.
- [20] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. arXiv:2407.21783. Retrieved from <https://arxiv.org/abs/2407.21783>
- [21] Mohamad Eid, Ramiro Liscano, and Abdulmotaleb El Saddik. 2006. A novel ontology for sensor networks data. In *Proceedings of the 2006 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*. IEEE, 75–79.
- [22] Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. 150–158.
- [23] Wenqi Fan, Yujian Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 6491–6501.
- [24] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 1 (2023).
- [25] W3C SPARQL Working Group. 2013. *SPARQL 1.1 Overview*. Technical Report. W3C. Retrieved from <https://www.w3.org/TR/sparql11-overview/>
- [26] Tao Gu, Xiao Hang Wang, Hung Keng Pung, and Da Qing Zhang. 2020. An ontology-based context model in intelligent environments. arXiv:2003.05055. Retrieved from <https://arxiv.org/abs/2003.05055>
- [27] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv:2501.12948. Retrieved from <https://arxiv.org/abs/2501.12948>
- [28] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2025. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems* 37 (2025), 132876–132907.
- [29] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d’Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *Synthesis Lectures on Data, Semantics, and Knowledge* 12, 2 (2021), 1–257.
- [30] C. Hou, H. Hsiao, C. King, and C. Lu. 2005. Context discovery in sensor networks. In *Proceedings of the 3rd International Conference on Information Technology: Research and Education, 2005 ITRE 2005*.

- [31] Mohamed Manzour Hussien, Angie Nataly Melo, Augusto Luis Ballardini, Carlota Salinas Maldonado, Rubén Izquierdo, and Miguel Angel Sotelo. 2025. Rag-based explainable prediction of road users behaviors for automated driving using knowledge graphs and large language models. *Expert Systems with Applications* 265 (2025), 125914.
- [32] Krzysztof Janowicz, Armin Haller, Simon J. D. Cox, Danh Le Phuoc, and Maxime Lefrançois. 2019. SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics* 56 (2019), 1–10.
- [33] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. H. Butler, and L. Tran. 2004. *Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies, Ver. 1.0*. Technical Report. W3C.
- [34] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models. arXiv:2405.17428. Retrieved from <https://arxiv.org/abs/2405.17428>
- [35] Jiewu Leng, Weinan Sha, Baicun Wang, Pai Zheng, Cunbo Zhuang, Qiang Liu, Thorsten Wuest, Dimitris Mourtzis, and Lihui Wang. 2022. Industry 5.0: Prospect and retrospect. *Journal of Manufacturing Systems* 65 (2022), 279–295.
- [36] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (2020), 7871–7880.
- [37] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.
- [38] Xiang Li, Zhenyan Lu, Dongqi Cai, Xiao Ma, and Mengwei Xu. 2024. Large language models on mobile devices: Measurements, analysis, and insights. In *Proceedings of the Workshop on Edge and Mobile Foundation Models*. 1–6.
- [39] Mingfei Liu, Xinyu Li, Jie Li, Yahui Liu, Bin Zhou, and Jinsong Bao. 2022. A knowledge graph-based data representation approach for IIoT-enabled cognitive manufacturing. *Advanced Engineering Informatics* 51 (2022), 101515.
- [40] Suqing Liu, Zezhu Yu, Feiran Huang, Yousef Bulbulia, Andreas Bergen, and Michael Liut. 2024. Can small language models with retrieval-augmented generation replace large language models when learning computer science? In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*. 388–393.
- [41] Praveen Kumar Reddy Maddikunta, Quoc-Viet Pham, B. Prabadevi, Natarajan Deepa, Kapal Dev, Thippa Reddy Gadekallu, Rukhsana Ruby, and Madhusanka Liyanage. 2022. Industry 5.0: A survey on enabling technologies and potential applications. *Journal of Industrial Information Integration* 26 (2022), 100257.
- [42] Ermelinda Oro, Francesco Maria Granata, Antonio Lanza, Amir Bachir, Luca De Grandis, and Massimo Ruffolo. 2024. Evaluating retrieval-augmented generation for question answering with large language models. *CEUR Workshop Proceedings* 3762 (2024), 129–134.
- [43] Amir Padovitz, Seng Wai Loke, and Arkady Zaslavsky. 2004. Towards a theory of context spaces. In *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004*. 38–42.
- [44] Adam Pease, Ian Niles, and John Li. 2002. The suggested upper merged ontology: A large ontology for the semantic web and its applications. In *Proceedings of the Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web*. 7–10.
- [45] C. Perera and A. V. Vasilakos. 2016. A knowledge-based resource discovery for Internet of Things. *Knowledge-Based Systems* 109 (2016), 122–136.
- [46] Ruiyang Qin, Zheyu Yan, Dewen Zeng, Zhenge Jia, Dancheng Liu, Jianbo Liu, Ahmed Abbasi, Zhi Zheng, Ningyuan Cao, Kai Ni, et al. 2024. Robust implementation of retrieval-augmented generation on edge-based computing-in-memory architectures. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*. 1–9.
- [47] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).
- [48] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67.
- [49] Mads Holten Rasmussen, Maxime Lefrançois, Georg Ferdinand Schneider, and Pieter Pauwels. 2021. BOT: The building topology ontology of the W3C linked building data group. *Semantic Web* 12, 1 (2021), 143–161.
- [50] Swarna Priya R. M., Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, Siva Rama Krishnan Somayaji, Kuruva Lakshmana, Rajesh Kaluri, Aseel Hussien, and Thippa Reddy Gadekallu. 2020. Load balancing of energy cloud using wind driven and firefly algorithms in internet of everything. *Journal of Parallel and Distributed Computing* 142 (2020), 16–26.
- [51] Sujay Roychowdhury, Sumit Soman, H. G. Ranjani, Neeraj Gunda, Vansh Chhabra, and Sai Krishna Bala. 2024. Evaluation of RAG metrics for question answering in the telecom domain. In *Proceedings of the ICML 2024 Workshop on Foundation Models in the Wild*. arXiv:2407.12873. Retrieved from <https://arxiv.org/abs/2407.12873>

- [52] David J. Russomanno, Cartik R. Kothari, and Omoju A. Thomas. 2005. Building a sensor ontology: A practical approach leveraging ISO and OGC models. In *Proceedings of the IC-AI*. 637–643.
- [53] B. N. Schilit, N. Adams, and R. Want. 1994. Context-aware computing applications. In *Proceedings of the 1994 1st Workshop on Mobile Computing Systems and Applications*. 85–90.
- [54] F. A. Schreiber, L. Tanca, R. Camplani, and D. Viganò. 2012. Pushing context-awareness down to the core: more flexibility for the PerLa language. In *Proceedings of the Electronic Proc. PersDB 2012 Workshop (Co-located with VLDB 2012)*. 1–6.
- [55] Monica Marconi Sciarroni, Marco Esposito, Paola Pierleoni, and Emanuele Storti. 2024. Monitoring Data Streams in Industry 5.0: A Knowledge Graph Approach. In *Proceedings of the 2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI)*. IEEE, 566–571.
- [56] B. Sindhu, R. P. Prathamesh, M. B. Sameera, and S. KumaraSwamy. 2024. The evolution of large language model: Models, applications and challenges. In *Proceedings of the 2024 International Conference on Current Trends in Advanced Computing (ICCTAC)*. IEEE, 1–8.
- [57] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288. Retrieved from <https://arxiv.org/abs/2307.09288>
- [58] Lewis Tunstall, Edward Emanuel Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of LM alignment. In *Proceedings of the 1st Conference on Language Modeling*.
- [59] Xiao Hang Wang, D Qing Zhang, Tao Gu, and Hung Keng Pung. 2004. Ontology based context modeling and reasoning using OWL. In *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004*. IEEE, 18–22.
- [60] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics ACL 2024*. 6233–6251.
- [61] Chenhao Xu, Longxiang Gao, Yuan Miao, and Xi Zheng. 2025. Distributed retrieval-augmented generation. arXiv:2505.00443. Retrieved from <https://arxiv.org/abs/2505.00443>
- [62] Xun Xu, Yuqian Lu, Birgit Vogel-Heuser, and Lihui Wang. 2021. Industry 4.0 and Industry 5.0–Inception, conception and perception. *Journal of Manufacturing Systems* 61 (2021), 530–535.
- [63] Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2024. Evaluation of retrieval-augmented generation: A survey. In *Proceedings of the CCF Conference on Big Data*. Springer, 102–120.
- [64] Zhongkai Yu, Shengwen Liang, Tianyun Ma, Yunke Cai, Ziyuan Nan, Di Huang, Xinkai Song, Yifan Hao, Jie Zhang, Tian Zhi, et al. 2024. Cambricon-llm: A chiplet-based hybrid architecture for on-device inference of 70b llm. In *Proceedings of the 2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 1474–1488.
- [65] Daqing Zhang, Tao Gu, and Xiaohang Wang. 2005. Enabling context-aware smart home with semantic web technologies. *International Journal of Human-friendly Welfare Robotic Systems* 6, 4 (2005), 12–20.
- [66] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *Proceedings of the 11th International Conference on Learning Representations, ICLR 2023*.
- [67] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, J. Jiang, and B. Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. arXiv:2402.19473. Retrieved from <https://arxiv.org/abs/2402.19473>
- [68] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. <https://arxiv.org/abs/2303.18223v16>
- [69] Lianghui Zhu, Xinggang Wang, and Xinlong Wang. JudgeLM: Fine-tuned large language models are scalable judges. In *Proceedings of the 13th International Conference on Learning Representations*.

Received 28 February 2025; revised 31 May 2025; accepted 6 August 2025