

Secondary Publication



Bahr, Lukas; Wehner, Christoph; Wewerka, Judith; u. a.

Knowledge graph enhanced retrieval-augmented generation for failure mode and effects analysis

Date of secondary publication: 31.10.2025

Version of Record (Published Version), Article

Persistent identifier: urn:nbn:de:bvb:473-irb-111079x

Primary publication

Bahr, Lukas; Wehner, Christoph; Wewerka, Judith; u. a. (2025): Knowledge graph enhanced retrieval-augmented generation for failure mode and effects analysis, in: Journal of Industrial Information Integration, Amsterdam: Elsevier BV, Vol. 45, Nr. 100807, pp. 1–9, doi: 10.1016/j.jii.2025.100807.

Legal Notice

This work is protected by copyright and/or the indication of a licence. You are free to use this work in any way permitted by the copyright and/or the licence that applies to your usage. For other uses, you must obtain permission from the rights-holders.

This document is made available under a Creative Commons license.



The license information is available online:

<https://creativecommons.org/licenses/by-sa/4.0/legalcode>



Contents lists available at ScienceDirect

Journal of Industrial Information Integration

journal homepage: www.elsevier.com/locate/jii

Full length article



Knowledge graph enhanced retrieval-augmented generation for failure mode and effects analysis

Lukas Bahr ^{a,c},* , Christoph Wehner ^b, Judith Wewerka ^c, José Bittencourt ^c, Ute Schmid ^b, Rüdiger Daub ^{d,e}

^a TUM School of Engineering and Design, Technical University of Munich, Germany

^b Cognitive Systems Group, University of Bamberg, Germany

^c Digitization Department for Battery Production, BMW Group, Germany

^d Institute for Machine Tools and Industrial Management, Technical University of Munich, Germany

^e Fraunhofer Institute for Casting Composite and Processing Technology IGCV, Fraunhofer, Germany

ARTICLE INFO

Keywords:

FMEA
Risk assessment
Knowledge graph
Retrieval-augmented generation
Large language models

ABSTRACT

Failure mode and effects analysis (FMEA) is an essential tool for mitigating potential failures, particularly during the ramp-up phases of new products. However, its effectiveness is often limited by the reasoning capabilities of the FMEA tools, which are usually tabular structured. Meanwhile, large language models (LLMs) offer novel prospects for advanced natural language processing tasks. However, LLMs face challenges in tasks that require factual knowledge, a gap that retrieval-augmented generation (RAG) approaches aim to fill. RAG retrieves information from a non-parametric data store and uses a language model to generate responses. Building on this concept, we propose to enhance the non-parametric data store with a knowledge graph (KG). By integrating a KG into the RAG framework, we aim to leverage analytical and semantic question-answering capabilities for FMEA data. This paper contributes by presenting set-theoretic standardization and a schema for FMEA data, an algorithm for creating vector embeddings from the FMEA-KG, and a KG-enhanced RAG framework. Our approach is validated through a user experience design study, and we measure the precision and performance of the context retrieval recall.

1. Introduction

A global and close-to-simultaneous start of production for a new product challenges multiple and interdisciplinary teams along the vertical and horizontal value chain. Quality assurance teams are particularly concerned during the start of production with poorly controlled processes [1]. Furthermore, processes are often complex, and cause-and-effect relationships are difficult to identify, for example, due to various operations, coordination, or communication barriers in a global production network [2,3]. For this reason, experience gained in pre-series with regard to possible problems during the ramp-up and operation of production systems must be transferred to the series plants in a global production network using user-centered approaches [4].

Failure mode and effects analysis (FMEA) is one tool to avoid potential failures during a ramp-up phase. FMEA is a risk analysis tool that focuses on systematically identifying failures and preventing defects, for example, in the process chain or the product's design [5, 6]. Typically, many stakeholders are involved in complex products, leading to different and incoherent FMEA approaches that make it

difficult to reason over the FMEA analysis between the different development units [7]. Furthermore, FMEA documents become quickly inextricable as actors maintain inconsistent data entries, leading to documentation that is difficult to evaluate with unclear completeness and integrity [8]. Lastly, FMEA results are usually tabular structured and lack reasoning capabilities, such as finding generalizations in error patterns, summarizing, or analytic guidance for error prevention.

Recent advances in large language models (LLMs) enable fine-tuning of custom data sets. This can be used for information retrieval and interactions with facts of the FMEA. LLMs are commonly trained on large amounts of text and show great potential, among others, in tasks such as language comprehension or answering questions [9]. However, LLMs often fail when factual knowledge is required [10]. Designing LLMs for tasks that require factual knowledge remains an open research question. In [11], the authors show that retrieval-augmented generation (RAG) outperforms language models for question-answering (QA) tasks that require factual precision. The RAG architecture stores the information in a non-parametric store and utilizes a language model to

* Corresponding author.

E-mail address: lukas.bahr@tum.de (L. Bahr).

<https://doi.org/10.1016/j.jii.2025.100807>

Received 27 August 2024; Received in revised form 17 February 2025; Accepted 20 February 2025

Available online 6 March 2025

2452-414X/© 2025 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

generate the answer. A non-parametric store is a flexible and schema-less database solution for storing and retrieving data. However, current RAG approaches face difficulties extracting analytical data due to the loss of symbolic meaning attached to numerical values. This limits drawing conclusions from the numerical information of the FMEA. For example, “*What failure cause has the highest risk priority number for the process step X?*” cannot be correctly inferred from the non-parametric data store.

One possible solution is to embed the non-parametric data stored in a knowledge graph (KG). KGs store symbolic facts by representing data as nodes and relations in a directed, labeled multi-graph [12]. The symbolic structure of the KG contextualizes the data and allows interpretable reasoners to derive novel insights from the data [13]. Prominent tools such as search engines (Google, Bing, etc.) or query answering services (Apple Siri, Amazon Alexa, etc.) have recognized the benefits of semantic technologies [14,15].

This paper contributes by integrating KGs within a RAG framework for user-friendly and analytic QA on FMEA data. This paper formalizes a set-theoretic standardization of the FMEA, enabling any FMEA to be mapped into a KG, regardless of the tools used for the FMEA’s creation or storage. To transpose this set-theoretic standardization into a KG, the paper proposes a schema for any FMEA. Subsequently, the paper suggests an algorithm for traversing the FMEA-KG to create vector embeddings. Finally, the paper demonstrates how to integrate the non-parametric data store into a KG, resulting in a KG-enhanced RAG (KG-RAG) framework. The KG-RAG framework enables retrieving human-understandable information while allowing analytical capabilities through the KG graph query. A user experience design study provides information on the effectiveness of the approach. We further measure the performance of context recall and precision.

2. Related work

The following briefly introduces the influential work on FMEA, KG, LLM, and RAG relevant to this paper. The methods defined in ISO/IEC 31010 [5] dominate the risk assessment in manufacturing, inter alia: the five whys method [16], fault tree analysis [17], and failure mode and effect analysis (FMEA) [6,18]. FMEA is a systematic methodology to analyze potential failure modes within a system to classify the severity and likelihood of failures [19]. This structured approach aims to identify, prioritize, and mitigate risks associated with system failures [20].

However, challenges arise when dealing with convoluted systems, such as in manufacturing [21]. In particular, incoherent FMEA approaches remain due to the intersection between different domains [7], difficulties in identifying and managing errors [22], and large and poorly structured FMEA documents [8]. Some methods aim to facilitate the accessibility of FMEA documents, such as integrating FMEA with quality function deployment [23]. Recent research has integrated game theory into FMEA to improve prioritization methods during its establishment. The approach in [24] employs a zero-sum game strategy to rank failure modes and dynamically refine risk assessments. The best-worst method utilizes Pythagorean fuzzy uncertain linguistic variables and Nash equilibrium principles [25].

Instead of storing the FMEA data in a classical relational (or non-relational) database, KGs offer versatile access to information. KGs are a way to store and retrieve facts [12], making them a crucial technology to answer questions based on factual knowledge [11]. Novel KG tools allow for the storage of vector embeddings [26]. In [27], KGs are initially conceived as semantic networks and have diverse applications, including ventures into human language through projects such as WordNet [28]. Subsequently, a multitude of private companies and academic institutions have embraced KGs for a wide array of applications [12], such as DBpedia [29], Freebase [30], or Google KG [31]. The default method of interacting with formalized knowledge in KGs is a structured query language (e.g., Cypher, SPARQL) [12,32],

which requires domain knowledge of the KG schema for effective data extraction, making interactions cumbersome for non-domain experts.

Advances in LLMs allow for fine-tuning custom data sets, enabling a retrieval approach for FMEA observations. The LLM’s transformer architecture enables training with its self-attention mechanism for multiple sequences that simplify, for example, the optimization process and minimize the risk of vanishing gradients [9,33]. LLMs are trained on large amounts of data and tackle various tasks in the field of natural language processing (NLP), such as language comprehension or QA, and have also found adoption in the manufacturing domain [9,34]. These models have shown significant advantages over non-transformer architectures, such as recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, in terms of scalability, improved contextual embeddings, and overall performance [33]. OpenAI’s GPT (Generative pre-trained transformer) models, such as GPT-4 [35], are noteworthy for their application of large language models (LLMs) in dialogues, providing human-like conversational capabilities [36]. However, LLMs struggle in tasks requiring factual knowledge, making LLMs unreliable in expert domains critical to manufacturing [10]. Furthermore, LLMs tend to hallucinate, leading to misinformation and making them unreliable in tasks where accurate knowledge recall is crucial [37, 38]. Additionally, LLMs are a black-box architecture that results in a lack of interpretability and complicates understanding the reasoning of the model [39,40].

Rather than fine-tuning a black-box model on FMEA data, LLMs allow retrieving vector embedding of its input, permitting architectures for better explainable methods such as RAG [41]. RAG approaches save information chunks as vector embeddings in a non-parametric store and utilize natural language generation methods to generate the answer from the store [42]. This approach allows for the composition of responses that minimize hallucinations, making the responses of chat-based LLMs more fact-based. However, since the information is represented as a vector, the literal and symbolic meaning of the numerical values is lost, which remains a challenge for the current RAG approaches [43]. For example, retrieving the highest-risk priority number of the FMEA likely outputs the wrong information, as the symbolic meaning of the number is embedded in a vector.

Vector embeddings represent the strings in a high-dimensional vector space. In [44], the authors initially propose embedding words into a meaningful representation by taking advantage of the vector dimensionalities. Novel advanced methods, such as BERT [45], adopt transformer-based self-supervised language models, resulting in richer representations of the language structure. In [35,46], it is shown that pre-training on a sufficiently large batch size can lead to high-quality vector representations.

3. QA for FMEA

In this paper, we introduce the KG-enhanced RAG (KG-RAG) framework for analytical and semantic QA on FMEA data. Unlike fine-tuning an LLM, the KG-RAG framework allows dynamic data updating without relearning and basic numerical analytics, like identifying the failure mode with the highest risk priority number. In Section 3.1, we formalize FMEA set-theoretically, enabling transposition into the FMEA knowledge graph described in Section 3.2. We also outline an algorithm for computing vector embeddings from the FMEA-KG. In Section 3.3, we introduce the KG-RAG framework, which retrieves the FMEA context through vector search, enhanced with KG graph queries, for customized user query results.

3.1. Set-theoretic formalization of FMEA

In the context of manufacturing systems, any FMEA can be formally defined using a set-theoretic framework. An FMEA consists of a set F where each element represents a distinct failure mode [47]. A failure mode is the specific manner in which a process, product, or system could potentially fail to perform its intended function. Formally, let

$F = \{f_1, f_2, \dots, f_n\}$ be the set of all identified failure modes. Each failure mode $f \in F$ is associated with the following elements [47]:

- **Failure effect** $E(f)$: The effect or impact of the failure mode on the system.
- **Failure cause** $C(f)$: The underlying cause or reason for the failure mode.
- **Failure measure** $M(C(f))$: The measure used to evaluate the failure mode.

Furthermore, there is a mapping $A : F(f) \rightarrow P$ from the failure mode $F(f)$ to its appearance within the manufacturing line, where P denotes the occurrence in the manufacturing process, such as process step or station. The FMEA employs further mappings that evaluate the severity, occurrence probability, and detection probability with a rating $R = \{1, \dots, 5\}$, for which a higher rating indicates a higher criticality for the failure mode [20]. The risk priority number (RPN) gives the overall risk level rating of the failure mode. We follow the mappings:

- **Severity** S : A mapping $S : E(f) \rightarrow R$, which takes a failure effect $E(f)$ and returns the severity rating.
- **Occurrence probability** O : A mapping $O : C(f) \rightarrow R$, which takes a failure cause $C(f)$ and returns the probability rating of the event.
- **Detection probability** D : A mapping $D : M(C(f)) \rightarrow R$, which takes a failure measure $M(C(f))$ and returns the detection rating probability.
- **Risk priority number** RPN : A mapping $RPN : (S, O, D) \rightarrow U$, which takes S , O , and D and returns the overall risk level rating $U = S \cdot O \cdot D$ [6,18].

Defining the core elements and mappings of an FMEA in this manner allows any specific FMEA to be integrated into the set-theoretic formalization. This approach ensures consistency and facilitates systematic analysis across different FMEAs. In addition, it enables the transposition of any FMEA into a KG for querying and vector search.

3.2. Modeling FMEA data within a KG

We propose a two-step process to store and embed FMEA data in a KG. Typically, FMEA data are structured in a tabular format within a database. First, the tabular data structure is mapped to the previously proposed set-theoretic FMEA formalization. The standardized set helps to transpose the FMEA data into a KG structure for which a schema is defined. Second, the FMEA data must be aggregated, embedded, and attached to the KG to enable the graph structure for vector search.

Transposing the FMEA data into a KG

Consider a KG \mathcal{G} to be a directed graph defined by

$$\mathcal{G} = (N, R, T), \quad (1)$$

where $n \in N$ is the set of nodes and $r \in R$ are the relations of the graph [12,27]. The set T consists of facts (triples) in the form $T = (n_h, r, n_t)$, where n_h is the head node, r is the relation, and n_t is the tail node [12]. KGs enhanced with literals $l \in L$ add descriptive features to the graph. These literals provide additional information about the nodes and their relations, transforming \mathcal{G} into a knowledge graph with literals. In this context, a node is represented as a tuple $n = (s_n, l_1, \dots, l_k)$, where s_n denotes the symbol of the node (such as its name), and l_1, \dots, l_k are literals providing descriptive information [12]. Literals can be descriptive strings, numeric values, or any other information about the node. Similarly, a relationship is indicated as $r = (s_r, l_1, \dots, l_m)$, where s_r is the symbol that identifies the relationship r . The variables k and m denote the number of literals associated with the nodes and the relations, respectively.

KG can be structured using a schema that provides a set of classes and rules specific to the domain. This includes establishing a class

FMEA Chatbot

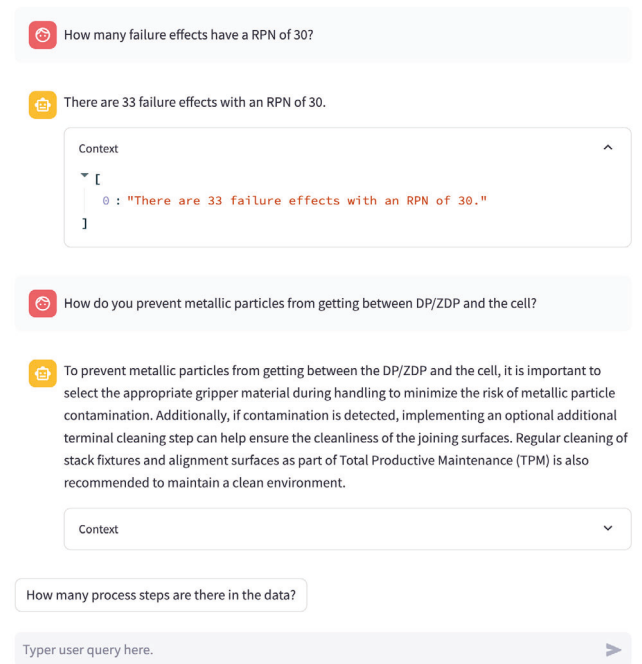


Fig. 1. Screen capture of the FMEA chatbot. The user interface is kept simple. Clicking on context reveals the information retrieved from the FMEA-KG.

hierarchy and defining the types of relations, such as their origin and target categories [12]. To formalize an FMEA schema, we map the set-theoretic formalization of an FMEA to the corresponding node s_n and relation s_r symbols of the schema. For $f \in F$, each failure mode corresponds to a subgraph of the FMEA-KG. We define the nodes $F(f)$ as *FailureMode*, $E(f)$ as *FailureEffect*, $C(f)$ as *FailureCause*, $M(C(f))$ as *FailureMeasure* and P as *ProcessStep*. The schema realizes a hierarchy between $F(f)$ and $C(f)$ by the relationship *isDueToFailureCause*, $C(f)$ and $M(C(f))$ by *isImprovedByFailureMeasure*, $F(f)$ and $E(f)$ by *resultsInFailureEffect*. The mapping $A : F(f) \rightarrow P$ corresponds to the relation *occursAtProcessStep*, linking the failure effect to a process step. Other mappings such as RPN , S , O , and D are numerical literals associated with their respective nodes within the KG schema. Fig. 2 shows a visualization of the complete FMEA schema.

The modeling of FMEA as KG allows the usage of graph algorithms, such as reasoning, clustering, or pathfinding, and a query language that facilitates the retrieval of complex joint information [12]. Although there are sophisticated reasoning methods within KGs, e.g., ruled-based or distributed representation-based inference methods, the methods do not scale with the KG's size or have difficulty retrieving deeper compositional information [48]. Particularly poorly formulated queries pose challenges for reasoning methods. To address this issue, we propose to embed information chunks of the FMEA data as vector representations, enabling vector-based reasoning.

Creating vector embeddings of the FMEA-KG

To ensure meaningful information extraction from the vector search, vector embeddings must represent the entire FMEA graph. For this, we build information chunks from the KG that capture the underlying structure of the FMEA. For example, an inquiry such as "What is the consequence of X on Y?" is not explicit and depends on the context X and Y . The results could differ by a slight change with the context of the inquiry. Hence, adopting an efficient approach that encompasses all potential information within the FMEA data is essential.

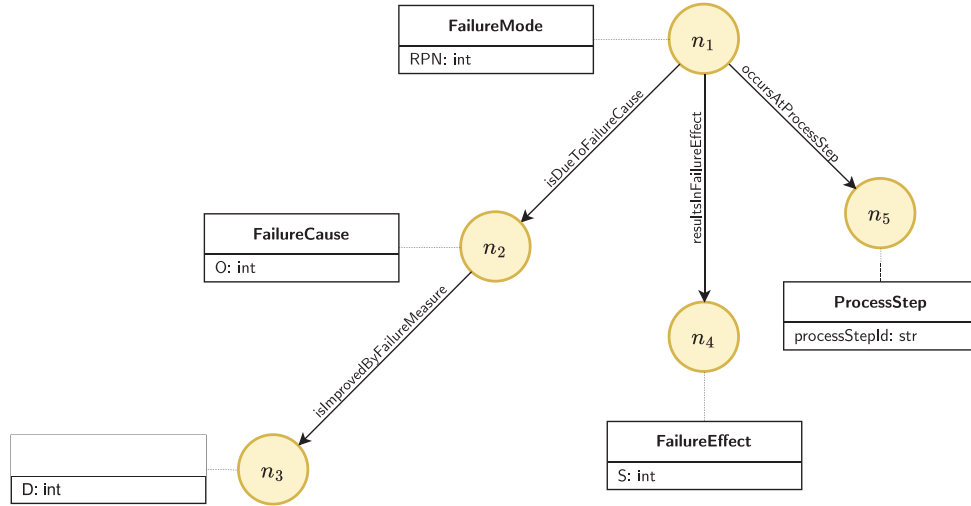


Fig. 2. Overview of the proposed schema for the FMEA, including the symbols, literals, and relations. The nodes ($n_1 - n_5$) represent a failure mode and illustrate how depth-first search is conducted on the tree-like graph structure.

Algorithm 1 Get vector embeddings from FMEA graph \mathcal{G}

```

1: procedure GETVECTOR EMBEDDINGS( $\mathcal{G}$ )
2:    $\triangleright$  vectorEmbeddings : set of vector embeddings
3:   for  $node \in FailureMode$  do
4:     connectedNodes  $\leftarrow$  DFS( $node$ )
5:      $\triangleright$  chunk : string holding properties of connectedNodes
6:     for  $properties \in connectedNodes$  do
7:       chunk  $\leftarrow$  add(properties)
8:     end for
9:     vectorEmbeddings  $\leftarrow$  computeVectorEmbedding(chunk)
10:  end for
11:  return vectorEmbeddings
12: end procedure

```

For this, we propose traversing the FMEA-KG by exploiting the tree-like structure of the FMEA-KG (cf. Fig. 2) to transform the hierarchical structure into a sequential structure. For each *FailureMode* node, we perform a depth-first search (DFS) to obtain the connecting nodes. Next, we compile a singular string representation of the nodes traversed from the root node. With the generated text chunk, we compute the vector embeddings utilizing any word-to-vector encoder. Algorithm 1 fully details the steps for getting the vector embeddings. Subsequently, we create a new node n labeled *VectorEmbedding* that indexes and stores the vector embeddings as literal and links it to the corresponding *FailureMode* node.

3.3. KG-RAG framework

The LLM leverages its extensive training on vast amounts of data to comprehend language nuances and provide contextually accurate output. We define the operation of the LLM in the following way. LLMs process text input x based on a given context or instruction c to generate an output. $h_{process}$ takes x and c as input and generates a combined input. This combined input is fed into the LLM, which computes the inference by $LLM(h_{process}(x, c))$.

Let \mathcal{G} denote the FMEA-KG proposed in Section 3.2. Subsequently, we introduce the KG-enhanced RAG framework for information retrieval from \mathcal{G} . The framework has two components for the retrieval of information: (i) the graph query language of \mathcal{G} and (ii) vector search.

The KG query language enables retrieving specific information for which the object is known or for basic analytics. Let w denote a user inquiry. For example, “What is the cumulative risk priority number for

the process step X ?” An LLM automatically generates a specific query q adhering to the querying language definition. For this, the LLM generates the graph query using the appropriate graph query language, such as SPARQL, by leveraging the context c provided by the graph’s semantics and instruction. The graph query $q = LLM(h_{process}(w, c))$ is then executed on \mathcal{G} . The process can be expressed by $Q : q \rightarrow \mathcal{R}_{query}$, where \mathcal{R}_{query} is the set of retrieved query results from \mathcal{G} .

For queries that do not return results due to unknown semantic information, such as wrong node label conventions in w or failures in the automatic query generation, we provide a vector search based on vector embeddings in \mathcal{G} . Let v_w denote the vector representation of w , and $\{v_i\}_{i=1}^I$ be the set of vector embeddings in the database, where I is the total number of embeddings in \mathcal{G} . We define a similarity function $S : (v_w, \{v_i\}_{i=1}^I) \rightarrow \{v_{i_1}, v_{i_2}, \dots, v_{i_k}\}$, which returns the vectors k most similar to v_w . The results of the most similar vectors are given by \mathcal{R}_{vector} . The similarity function is given by the cosine function $S : (v_w, v_i) \rightarrow \cos(\theta) = (v_w \cdot v_i) / (\|v_w\| \|v_i\|)$, which measures the similarity between v_w and v_i [49]. It ranges from -1 to 1 , with -1 indicating completely opposite vectors and 1 indicating similar vectors.

The general KG-RAG framework operates with two pipelines: one for retrieving information from the FMEA-KG given the user inquiry w and another for serving the information to the user. Fig. 3 summarizes the framework. Initially, the user inquiry w is fed into an LLM (cf. Fig. 3 (1)). The LLM generates a graph query q , considering w and the graph’s semantic structure of the FMEA schema (cf. Fig. 3 (2)). If $Q(q)$ yields a result, the information retrieval is complete (cf. Fig. 3 (3)). If $Q(q)$ does not output any results, we perform a vector search $S(v_w, \{v_i\}_{i=1}^I)$ to retrieve the k most similar vectors (cf. Fig. 3 (4)). The results from the retrieval process, either \mathcal{R}_{query} or \mathcal{R}_{vector} are input to the LLM to generate a response addressing the user’s inquiry embedded in the context c . This can be expressed by $LLM(h_{process}(\mathcal{R}, c))$.

Handling of epistemic uncertainty

LLMs are prone to hallucination when faced with epistemic uncertainty when they lack sufficient training data for similar examples [50]. By default, LLMs cannot indicate uncertainty and generate plausible-sounding responses regardless of confidence. The proposed KG-RAG framework approach addresses this by relying on retrieved context rather than model parameters for answer generation. It enables users to assess uncertainty by displaying retrieved contexts. Graph query-based contexts are highly reliable, as they derive from symbolic manipulation and factual data in the FMEA. For vector-based search, the KG-RAG approach provides the cosine distance alongside the retrieved context, signaling semantic proximity to the user query. Greater distances indicate higher epistemic uncertainty.

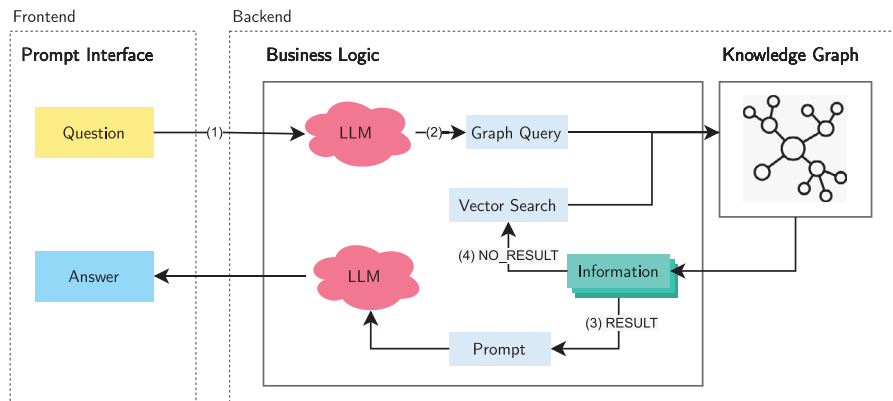


Fig. 3. Sketch of the proposed KG-RAG framework. The information is retrieved either by utilizing the graph query language of the KG or by using vector search. The framework employs an LLM to generate the query and to serve the result.

Table 1

Example data from FMEA used in our experiments. The FMEA contains data to mitigate risks in the production of HVS.

Process Step	Failure Mode	Failure Effect	S	Failure Cause	O	Failure Measure	D	RPN
Cell stacking	Incorrect cell placement	Misalignment of cooling system	7	Improper alignment guides	5	Adjust automated cell placement, visual inspection	3	105
Welding of cell contact system	Weak weld joints	Increased resistance, potential overheating	8	Improper welding parameters, contaminated surfaces	6	Weld quality checks, resistance testing	4	192
	Weld porosity	Reduced mechanical strength, potential failure	9	Contaminated surfaces, improper shielding gas flow	5	Weld X-ray inspection, surface cleaning procedures	3	135

Table 2

Overview of *min*, *max* and *average* number of relationships for each label in the FMEA-KG. The KG has a total of 3,107 nodes and 4,576 relationships.

Label	<i>min</i> (#relationships)	<i>max</i> (#relationships)	<i>avg</i> (#relationships)
<i>FailureMode</i>	2	11	3.45
<i>FailureEffect</i>	1	68	15.7
<i>FailureCause</i>	4	17	4.67
<i>FailureMeasure</i>	1	17	1.39
<i>ProcessStep</i>	2	71	20.30

4. Experiments

We evaluate the KG-RAG framework for FMEA QA, focusing on the capability of context retrieval. To facilitate user interaction with the KG-RAG framework, we introduce a simple FMEA chatbot interface. Section 4.1 covers our implementation details. Sections 4.2 and 4.3 detail the methodology and results of the user experience design study. Section 4.4 assesses the chatbot’s numerical data retrieval effectiveness by measuring context precision and recall. Finally, Section 4.5 discusses the results.

4.1. Implementation details

This Section details the specifics of our implementation and the FMEA data. For our results, we chose an actual FMEA dataset from a production of high-voltage systems (HVS) from prismatic battery cells. Example data from the FMEA is presented in Table 1. With up to 35% of the total production costs of an electric vehicle, HVS makes up the vehicle’s most expensive component. Therefore, there is a great interest in the need for well-defined quality measurements [51].

To construct the multi-label FMEA-KG with literals, we employed Neo4j [32], which provides us with its graph query language, Cypher. The FMEA-KG comprises 3,107 nodes encompassing process steps, failure modes, failure effects, failure causes, and failure measures. A total of 4,576 relations connect the nodes. An overview of the number

of relationships *min*, *max*, and *average* for each label in the FMEA-KG can be found in Table 2. The average number of relationships is between 1.39 (*FailureMeasure*) and 20.30 (*ProcessStep*). For example, there is a *FailureMeasure* named “Wait for work instruction”, which is related to 17 different *FailureCauses*.

The FMEA-KG contains 3,046 unique paths that can be traced from the root node *FailureMode*. That means that for each *FailureMode*, a path can be represented as a set of triples such that no two sets are identical. In other words, for each unique path P , there exists a set $S = \{(n_1, r_1, n_2), (n_2, r_2, n_3), \dots, (n_k, r_k, n_{k+1})\}$ where no other set S' in the graph is identical to S . A sample path is presented in Table 3.

To compute the vector embedding of this path, we iterate over the connected nodes to retrieve their respective properties and concatenate them into a single string. For example: “Process step: Cell stacking, Failure mode: Incorrect cell placement, Failure effect: Misalignment of the cooling system, S: 7 (...)” The vector embedding is then computed using the text-embedding-ada-002 model [52].

Another important aspect is the instruction design for running the LLM’s inference jobs. The instructions ensure that the LLM generates precise and relevant responses, enhancing the overall performance of context retrieval. The LLM of our KG-RAG framework is OpenAI’s GPT-4 model with version 1106-Preview [35]. We implement the KG-RAG framework’s business logic as the backend service with a REST-API and employ a chat interface for the study (cf. Fig. 1).¹

4.2. Study methodology

Although FMEA is a complex expert topic, information retrieval is relevant to all shop floor technical professionals involved in risk management. In our study, we engaged 10 participants, including FMEA experts and non-experts, to evaluate the effectiveness of our

¹ For those interested in reproducing the results with an example FMEA or exploring the framework further, the code for the RAG framework’s backend service is available on GitHub at <https://github.com/lukasbahr/kg-rag-fmea>.

Table 3

Example path P_1 from the FMEA-KG illustrates a sequence of nodes and relationships that trace a failure mode, its effects, causes, and mitigation measures.

$$P_1 = \{$$

(Cold Solder Joint, occursAtProcessStep Soldering Process),
 (Cold Solder Joint, resultsInFailureEffect, Intermittent Connection),
 (Cold Solder Joint, isDueToFailureCause, Inadequate Heating),
 (Inadequate Heating, isImprovedByFailureMeasure, Increase Heating Temperature)

$$\}$$

KG-RAG framework. The participants were assigned to extract information from the FMEA chatbot (cf. Section 4.1) and the original FMEA-Excel spreadsheet as a baseline. Assessing the performance of an information retrieval framework for FMEA presents unique challenges, as (i) there is no established baseline due to different FMEA approaches, (ii) the validity of the evaluation process in the study depends on the subject, and (iii) limited availability of domain experts. Nonetheless, in the context of FMEA, Excel as a comparative baseline is grounded in its widely used tool to work with FMEA in the manufacturing sector, particularly for small and medium-sized companies.

Throughout the study, each participant was asked to complete three distinct tasks, each designed to test the retrieval of different types of FMEA information. The tasks were designed to simulate scenarios professionals might encounter when dealing with FMEA on the shop floor. For example, “In step X process, there are faults due to Y. What prevention measure is provided to eliminate the fault?” For each task, the performance of the RAG framework and Excel was evaluated using five metrics: the correctness of the information retrieved, the usability of the interface, the relevance of the results to the task at hand, the completeness of the information provided, and the time taken to retrieve the necessary data. Employing these metrics allowed us, in addition to a qualitative assessment, to quantify and understand the advantages and potential drawbacks of the KG-RAG framework.

We give the subjects the following definitions for correctness, usability, relevance, completeness, and retrieval time.

Correctness: Evaluates the degree of precision of the information retrieved by the subject and whether it corresponds to the task requirement. For example, are the results based on information from the FMEA database or hallucinated by the model.

Usability: Measures the ease of use with which participants interact with the system interface. The interface is user-friendly, intuitive, and easy to navigate.

Relevance: Evaluates how closely the retrieved information aligns with the context of the question. Is there any unnecessary information for the task revealed?

Completeness: Indicates if the retrieved information provides a full and thorough answer to the task. For example, the result should match the inquiry for a task that asks for three failure causes.

Retrieval time: Measure the time and thus the efficiency of the retrieval process, particularly how long it takes a subject to obtain the information.

After each task, we asked the participants to assign a score (1-5) for each metric, reflecting their notion of how well they achieved the task with the FMEA chatbot and the Excel spreadsheet. We also asked them to share their thoughts and comments during and after each task and stopped the time.

4.3. Results of the study

The results of the user experience design study have been positive throughout, and the FMEA chatbot was across the board higher evaluated, indicating an improvement over Excel to access FMEA information. The correctness of the information retrieved from the FMEA chatbot is evaluated at 7.53% higher than Excel. Completeness and relevance achieve 13.47% and 18.23% higher results. Usability increased by 124.29%. Although the measured time the participants took for

each task is, on average, 00:32 min (24.72%) shorter with the FMEA chatbot, their subjective perception of time differs with an evaluation of the retrieval time of 74.28% higher compared to Excel. The highest standard deviation for the chatbot appears for completeness (0.92) and corresponds to the thoughts and comments we received from the participants. Table 4 summarizes the results.

FMEA experts highlighted that the given context added to the answer helped to build trust in the chatbot’s answer. The contexts provided sensible explanations, reduced the need for follow-up questions, and thus increased the relevance of the inquiry result. “Context is great: gives a sensible explanation with more information. Saves to re-ask.” Other experts said that working with FMEA in Excel spreadsheets is not a pleasant task, for example, when seeking multiple explanations for a failure cause or gathering a general overview of the FMEA. One participant expressed: “The chatbot approach makes the FMEA much more interesting for acquiring knowledge and for simply using. So far, FMEA with Excel has not been used pleasantly, and you always need the help of the FMEA owners.”

All participants advertised the ease of use of the FMEA chatbot. The ten participants were already familiar with chat-based LLMs, which helped them adapt to the setting. In particular, the ability to correctly interpret and respond to inquiries was highlighted. One user noted the proficiency of knowledge retrieval even though the input had vague language and, as a non-FMEA expert, did not use the precise terminology typically necessary for search queries in FMEA tools. “There was a spelling mistake in the query, which was relatively imprecise - I did not use the precise term I was searching for but still got the correct answer.” Another participant stated: “It is just cool that you do not have to be precise when describing and still get the right answer.”

Although all participants had a positive impression of the proposed approach, some experts stated that a more thorough evaluation of the validity of the data is necessary before deploying it on the shop floor. Another remark from FMEA experts was that it would be beneficial to integrate the chatbot into an existing FMEA tool.

4.4. Evaluation of the context retrieval

The handling of FMEAs requires numerical data extraction to improve risk mitigation strategies. For example, experts may need to determine the primary cause of failure within a particular process step, imposing questions such as “What failure mode has the highest RPN for the process step X?” In addition, experts also employ basic data analysis, such as identifying the process step with the mean RPN. An FMEA Excel spreadsheet would entail a series of computational actions. The merits of our proposed KG-RAG framework allow the retrieval of numerical data and basic statistical analysis of FMEA data.

To validate the effectiveness of our method, we evaluate the KG-RAG framework on a validation dataset comprising entries with questions and verified ground truth information about our FMEA data. These questions were a mix of open-ended questions (e.g., “How can we mitigate clocking in the cell stacking process?”), yes/no questions (e.g., “Did the machine stop working due to overheating?”), and ranking questions (e.g., “Rank the failure modes of the welding process step of the cell contact system by the risk priority number.”). We involved FMEA experts to ensure that the validation dataset is unbiased and that the ground truth data are accurately defined. As a benchmark, we compare our KG-RAG framework against a baseline KG-RAG implementation that uses vector search only and a baseline RAG implementation with a random chunking strategy that operates on tabular structured data [11]. The evaluation of the context retrieval is twofold, with the intent to verify both context recall and precision on the validation dataset. Context recall (CR) measures the extent to which the retrieved context aligns with the ground truth data and is defined by

$$CR = \frac{\#ground\ truth\ attributable\ statements}{\#sentences\ in\ ground\ truth}, \quad (2)$$

for which # is the number of total appearance [11]. Context precision (CP) is evaluated based on the ranking of the information within the

Table 4

We asked $n = 10$ participants to rate (1-5) each of the three tasks according to the depicted metrics for the FMEA chatbot and Excel. The results show the accumulated mean value with the standard deviation.

	Correctness	Usability	Relevance	Complete	Retrieval Time	Time Taken [min]
Excel	4.38 ± 0.80	2.10 ± 0.70	3.95 ± 1.02	3.86 ± 1.10	2.76 ± 0.77	01:51 ± 01:26
KG-RAG	4.71 ± 0.71	4.71 ± 0.46	4.67 ± 0.91	4.38 ± 0.92	4.81 ± 0.40	01:19 ± 01:00

Table 5

Results on context recall (CR) and context precision (CP) for a baseline RAG implementation and our proposed KG-RAG framework with and without query search.

	CR	CP
Baseline RAG (vector search)	0.17	0.29
KG-RAG (vector search)	0.22	0.44
KG-RAG	0.46	0.82

context and its relevance. It is calculated by

$$CP = \frac{1}{\# \text{relevant information}} \sum_{m=1}^n \left(\frac{\# \text{relevant information till } m}{m} \times r_m \right), \quad (3)$$

where n denotes the total number of information in the context and r_m is the binary value whether the information to the m th piece of information is true ($r_m = 1$) or false ($r_m = 0$) [53].

Table 5 presents the results for CR and CP comparing the KG-RAG framework with and without query search and the baseline RAG implementation. The results show that with our approach, CR and CP (0.46, 0.82) are higher than without query search (0.22, 0.44) and the baseline RAG implementation (0.17, 0.29). An examination of the retrieved context for the baseline RAG implementation and the KG-RAG framework with and without query search is shown in Table 6, for an example inquiry, "How many failure effects with a S value of over 5 exist?" The context size is set to $k = 3$. The contexts show that the method can correctly reason its answer from the incomplete context with vector search only. However, it could not correctly retrieve the number of failure effects from the FMEA data. The contexts of the baseline RAG are arbitrarily constructed, making it difficult for humans to trace the reasoning behind the answers.

4.5. Discussion of the results

In developing a KG-enhanced RAG framework specifically tailored for FMEA, the results in Section 4.3 show significant advances in factual knowledge recall. The KG-RAG framework's improved ability to reason and comprehend vague queries enhances FMEA's accessibility and user-friendliness, benefiting experts and non-experts in extracting insights. In addition, reduced information retrieval time helps practitioners in the workplace by enabling agile decision-making and problem-solving.

The results in Section 4.4 demonstrate that integrating a graph query search with a vector search enables basic analytical capabilities on datasets with mixed data types, such as FMEA, improving CR and CP. We further demonstrate that for datasets with a predefined structure, such as in expert domain data like FMEA, a well-defined chunking strategy can significantly enhance the results over random chunking strategies. This and the context visualization within the chat interface foster confidence in the KG-RAG framework's capabilities. One of the main challenges we faced was dealing with abbreviations and specific terminology used in our FMEA. The FMEA contains many abbreviations and specific terminology unknown to the LLM, as some are undisclosed. To address this issue, we replaced less-known abbreviations (e.g., CSS) with their full forms (e.g., cell contacting system).

Although the study presents generally positive outcomes, the very nature of the domain fitting of our approach brings threats to validity. The challenge lies in obtaining statistical significance across metrics such as correctness, usability, relevance, completeness, and retrieval

time, given the size of our user experience design study. In addition, conducting an automated evaluation proves challenging in the absence of the necessary expert knowledge, which complicates efforts to fully assess the performance of the KG-RAG framework.

On the technical level, further important aspects need to be explored. In particular, we did not test with alternative LLMs for text generation and vector embedding, as access to other online language models is restricted by data protection regulations. There are additional chunking strategies worth exploring. While we employ DFS to capture the underlying information of the FMEA-KG, other potentially more efficient path algorithms, such as breadth-first search or iterative deepening, are worth evaluating. Furthermore, other graph algorithms like community detection and centrality measures could provide deeper insights for summarization and analysis.

Discussion on handling epistemic uncertainty

Uncertainty handling is a critical aspect of the KG-RAG framework, particularly in dealing with ambiguity and incomplete data. The framework employs a dual-layered validation process: graph query results are prioritized for their symbolic accuracy. At the same time, vector search outputs are supplemented with cosine distance metrics to signal semantic reliability. This layered approach ensures that uncertainty is clearly communicated to the user, reducing the risk of erroneous inferences.

The KG-RAG framework integrates human oversight as a key component to further mitigate risks. Users can independently validate AI outputs by presenting retrieved context and distance metrics. This mechanism fosters transparency and supports informed decision-making, particularly in complex, domain-specific scenarios like FMEAs in manufacturing. The high usability and correctness score observed during the user experience design study underscores the effectiveness of these measures. Users expressed trust in the system due to its contextual clarity and the interpretability of uncertainty indicators.

Broader implications of AI in risk management

Integrating AI in risk management within industrial settings carries significant broader implications. As defined in ISO/IEC 31010 [5], particular predictive quality has become prevalent in the manufacturing industry. Predictive quality aims to proactively assess risks and identify failures and defects before they occur, defining a new era of AI-enhanced risk management [54]. This, combined with recent developments in data-driven statistical modeling or integration of virtual reality and augmented reality technologies, provides practitioners greater insights into the production process, improved predictive capabilities, and proactive risk assessment [55,56].

Using chatbots for risk management in manufacturing carries important implications that must be addressed. Before implementing chatbots, it is essential to examine ethical considerations, data privacy, and the importance of critical oversight. First and foremost, user safety must be ensured. For FMEAs that guide failure modes with potentially dangerous remedial measures, verifying information with domain experts is crucial to avoid any risk or harm to users. Second, user privacy must be protected, as interactions with chatbots may involve sensitive information. Lastly, every decision made by the chatbot requires trained users, particularly on the shop floor, and continuous human oversight to ensure safe and accurate operation.

5. Conclusion

In this paper, we propose a KG-RAG framework that synergies KG with LLM to enhance the QA capabilities within the FMEA domain. For that, we formalize a set-theoretic standardization of the FMEA,

Table 6

Example results of the answers and retrieved contexts for the question, “How many failure effects with an S value of over 5 exist?” and ground truth, “There are 14 failures that have a severity (S) score greater than 5.” The context size is set to $k = 3$. The question is formulated in particular to test the retrieval of numerical information.

	Answer	Contexts
Baseline RAG (vector search)	There are a total of 3 failure effects with an S value of over 5.	["3D Scanner 1 in deployment 3D System Generates pseudoIO related to defect images ...", "Contact system CSK Lid mounting Issue Low voltage battery production ...", "Contact system CSK Lid mounting Insulation issue 12 Particles ..."]
KG-RAG (vector search)	There is one failure with a S-value of over 5.	["Process step: Form cell stacks, Failure effect: Scrap cells/module, S: 6 (...)", "Process step: Form cell stacks, Failure effect: Process failure/increase throughput times, S: 3 (...)", "Process step: Form cell stacks, Failure effect: no particular effects, S:1 (...)"]
KG-RAG	There are 14 failures with a S score of over 5.	["NumberFailureEffectsWithSOver5: 14"]

allowing any FMEA to be transposed into KG. Building on this standardization, we develop a schema to model the FMEA data as a multi-label KG with literals. To compute the vector embeddings of the FMEA, we suggest an algorithm to fully capture the information from the KG. We then offer a solution for a novel KG-RAG framework that can retrieve FMEA information and allow for analytical inquiries.

The results of the user experience design study suggest a promising direction for the retrieval and interpretation of the semantic content of the FMEA data, enabling a more nuanced and analytical approach to risk assessment. Additionally, the evaluation of the context retrieval demonstrates that retrieval of numerical data is improved when augmented with the graph query interface of the KG. The findings indicate that a general integration of LLMs into the shop floor could advance quality strategies.

However, in the future, many questions remain open. For one, the work's focus is on the retrieval of information, thereby leaving the open investigation to other LLMs for their effectiveness in generating and computing vector embeddings. A suite of graph algorithms could further refine chunking strategies for KG-built RAG applications. Moreover, the framework can be extended to support guided user input, which could significantly improve the data quality of the FMEA database. Other thinkable investigations are around expanding the knowledge base to encompass other quality management data, such as PDCA or manufacturing data, which could allow for a broader application of a KG-RAG framework across different quality assurance contexts.

Funding

This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy, the BayVFP's “Digitalization” funding line, the KIProQua project and the BMW Group.

CRedit authorship contribution statement

Lukas Bahr: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Christoph Wehner:** Writing – original draft, Validation, Methodology, Conceptualization. **Judith Wewerka:** Writing – review & editing, Validation, Investigation. **José Bittencourt:** Validation, Funding acquisition, Data curation. **Ute Schmid:** Funding acquisition, Conceptualization. **Rüdiger Daub:** Writing – review & editing, Supervision, Project administration, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Lukas Bahr reports financial support, administrative support, article publishing charges, and equipment, drugs, or supplies were provided by BMW Group. Christoph Wehner reports financial support was provided by Bavarian Ministry of Economic Affairs, Regional Development and Energy. José Bittencourt and Judith Wewerka are employed at BMW Group. If there are other authors, they declare that they have no known

competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the engineers and co-workers on the shop floor who participated in the evaluation of the KG-RAG framework. The authors further gratefully acknowledge the financial support from the Bavarian Ministry of Economic Affairs and the BMW Group.

Data availability

The authors do not have permission to share data.

References

- [1] M. Colledani, T. Tolio, A. Yemane, Production quality improvement during manufacturing systems ramp-up, *CIRP J. Manuf. Sci. Technol.* 23 (2018) 197–206, <http://dx.doi.org/10.1016/j.cirpj.2018.07.001>.
- [2] L. Possner, L. Bahr, L. Roehl, C. Wehner, S. Groeger, Anwendung von causal-discovery-algorithmen zur root-cause-analyse in der Fahrzeugmontage, 2024, <http://dx.doi.org/10.48550/arXiv.2407.16388>.
- [3] C. Wehner, M. Kertel, J. Wewerka, Interactive and intelligent root cause analysis in manufacturing with causal Bayesian networks and knowledge graphs, in: 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), IEEE, Florence, Italy, 2023, pp. 1–7, <http://dx.doi.org/10.1109/VTC2023-Spring57618.2023.10199563>.
- [4] J. Eirich, D. Jäckle, M. Sedlmair, C. Wehner, U. Schmid, J. Bernard, T. Schreck, ManuKnowVis: How to support different user groups in contextualizing and leveraging knowledge repositories, *IEEE Trans. Vis. Comput. Graphics* 29 (8) (2023) 3441–3457, <http://dx.doi.org/10.1109/TVCG.2023.3279857>.
- [5] ISO/IEC, Risk management — risk assessment techniques, *Tech. Rep., International Organization for Standardization*, Geneva, Switzerland, 2019.
- [6] IEC, Iec 60812:2018 - failure modes and effects analysis (fmea and fmeca), *Tech. Rep., International Electrotechnical Commission (IEC)*, Geneva, Switzerland, 2018.
- [7] N.W. Ozarin, Bridging software and hardware FMEA in complex systems, in: 2013 Proceedings Annual Reliability and Maintainability Symposium (RAMS), IEEE, Orlando, FL, 2013, pp. 1–6, <http://dx.doi.org/10.1109/RAMS.2013.6517739>.
- [8] C.S. Carlson, *Effective FMEAs: Achieving safe, reliable, and economical products and processes using failure mode and effects analysis*, first ed., Wiley, New Jersey, 2012, <http://dx.doi.org/10.1002/9781118312575>.
- [9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., New York, NY, USA, 2020, pp. 1877–1901.
- [10] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and knowledge graphs: A roadmap, *IEEE Trans. Knowl. Data Eng.* (2024) 1–20, <http://dx.doi.org/10.1109/TKDE.2024.3352100>.
- [11] K. Shuster, S. Poff, M. Chen, D. Kiela, J. Weston, Retrieval augmentation reduces hallucination in conversation, in: Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 3784–3803, <http://dx.doi.org/10.18653/v1/2021.findings-emnlp.320>.

- [12] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G.D. Melo, C. Gutierrez, S. Kirrane, J.E.L. Gayo, R. Navigli, S. Neumaier, A.-C.N. Ngomo, A. Polleres, S.M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, *ACM Comput. Surv.* 54 (4) (2021) 71:1–71:37, <http://dx.doi.org/10.1145/3447772>.
- [13] S. Schramm, C. Wehner, U. Schmid, Comprehensible artificial intelligence on knowledge graphs: A survey, *J. Web Semant.* 79 (2023) 100806, <http://dx.doi.org/10.1016/j.websem.2023.100806>.
- [14] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, J. Taylor, Industry-scale knowledge graphs: lessons and challenges, *Commun. ACM* 62 (8) (2019) 36–43, <http://dx.doi.org/10.1145/3331166>.
- [15] V.V. Nigam, S. Paul, A.P. Agrawal, R. Bansal, A review paper on the application of knowledge graph on various service providing platforms, in: 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2020, pp. 716–720, <http://dx.doi.org/10.1109/Confluence47617.2020.9058298>.
- [16] O. Serrat, *The five whys technique*, in: Knowledge Solutions, Springer Singapore, Singapore, 2017, pp. 307–310.
- [17] E. Ruijters, M. Stoelinga, Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools, *Comput. Sci. Rev.* 15–16 (2015) 29–62, <http://dx.doi.org/10.1016/j.cosrev.2015.03.001>.
- [18] IEC, IEC 60812:2006 - Analysis Techniques for System Reliability - Procedure for Failure Mode and Effects Analysis (FMEA), Tech. Rep., International Electrotechnical Commission (IEC), Geneva, Switzerland, 2006.
- [19] R. Schmitt, T. Pfeifer, *Qualitätsmanagement: Strategien, Methoden, Techniken, 5., überarbeitete Auflage*, Hanser, München, 2015.
- [20] D.H. Stamatis, *Failure mode and effect analysis: FMEA from theory to execution*, ASQ Quality Press, Milwaukee, USA, 2003.
- [21] E. Henshall, I.F. Campean, B. Rutter, A systems approach to the development and use of FMEA in complex automotive applications, *SAE Int. J. Mater. Manuf.* 7 (2) (2014) 280–290, <http://dx.doi.org/10.4271/2014-01-0740>.
- [22] J.E. Hunt, D.R. Pugh, C.J. Price, Failure mode effects analysis: A practical application of functional modeling, *Appl. Artif. Intell.* 9 (1) (1995) 33–44, <http://dx.doi.org/10.1080/08839519508945466>.
- [23] B. Almannai, R. Greenough, J. Kay, A decision support tool based on QFD and FMEA for the selection of manufacturing automation technologies, *Robot. Comput. Integr. Manuf.* 24 (4) (2008) 501–507, <http://dx.doi.org/10.1016/j.rcim.2007.07.002>.
- [24] M. Yazdi, Enhancing system safety and reliability through integrated FMEA and game theory: A multi-factor approach, *Safety* 10 (1) (2024) <http://dx.doi.org/10.3390/safety10010004>.
- [25] H. Li, R. Moradi, M. Yazdi, E. Zarei, Nash equilibrium-based FMEA for risk prioritization in hydrogen refueling station design, in: M. Yazdi (Ed.), *Progressive Decision-Making Tools and Applications in Project and Operation Management: Approaches, Case Studies, Multi-Criteria Decision-Making, Multi-Objective Decision-Making, Decision under Uncertainty*, Springer Nature Switzerland, Cham, 2024, pp. 39–52.
- [26] S. Mittal, A. Joshi, T. Finin, Thinking, fast and slow: Combining vector spaces and knowledge graphs, 2017, <http://dx.doi.org/10.48550/ARXIV.1708.03310>.
- [27] E.W. Schneider, *Course modularization applied: The interface system and its implications for sequence control and data analysis*, ERIC Clearinghouse, Alexandria, VA., 1973.
- [28] G.A. Miller, WordNet: A lexical database for English, *Commun. ACM* 38 (11) (1995) 39–41, <http://dx.doi.org/10.1145/219717.219748>.
- [29] S. Auer, J. Lehmann, R. Cyganiak, Z. Ives, DBpedia: A nucleus for a web of open data, in: *The Semantic Web*, vol. 4825, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 722–735.
- [30] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, ACM, Vancouver Canada, 2008, pp. 1247–1250, <http://dx.doi.org/10.1145/1376616.1376746>.
- [31] A. Singhal, Introducing the knowledge graph: things, not strings, 2012, URL <https://blog.google/products/search/introducing-knowledge-graph-things-not/>. visited on 01/01/2024.
- [32] Neo4j, Neo4j graph database & analytics – The leader in graph databases, Graph Database & Anal. URL <https://neo4j.com/>. visited on 11/04/2024.
- [33] T. Lin, Y. Wang, X. Liu, X. Qiu, A survey of transformers, *AI Open* 3 (2022) 111–132, <http://dx.doi.org/10.1016/j.aiopen.2022.10.001>.
- [34] M. Holland, K. Chaudhari, Large language model based agent for process planning of fiber composite structures, *Manuf. Lett.* 40 (2024) <http://dx.doi.org/10.1016/j.mfglet.2024.03.010>.
- [35] OpenAI, GPT-4, Technical Report, 2023, <http://dx.doi.org/10.48550/ARXIV.2303.08774>.
- [36] W.X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, J.-R. Wen, A survey of large language models, 2023, <http://dx.doi.org/10.48550/ARXIV.2303.18223>, ArXiv:2303.18223.
- [37] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y.J. Bang, A. Madotto, P. Fung, Survey of hallucination in natural language generation, *ACM Comput. Surv.* 55 (12) (2023) 1–38, <http://dx.doi.org/10.1145/3571730>.
- [38] V. Rawte, S. Chakraborty, A. Pathak, A. Sarkar, S.T.I. Tonmoy, A. Chadha, A. Sheth, A. Das, The troubling emergence of hallucination in large language models - An extensive definition, quantification, and prescriptive remediations, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 2541–2573, <http://dx.doi.org/10.18653/v1/2023.emnlp-main.155>.
- [39] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kavas, P. Sen, A survey of the state of explainable AI for natural language processing, in: K.F. Wong, K. Knight, H. Wu (Eds.), Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, Suzhou, China, 2020, pp. 447–459.
- [40] C. Wehner, F. Powlesland, B. Altakroui, U. Schmid, Explainable online lane change predictions on a digital twin with a layer normalized LSTM and layer-wise relevance propagation, in: *Advances and Trends in Artificial Intelligence. Theory and Practices in Artificial Intelligence*, vol. 13343, Springer International Publishing, Cham, 2022, pp. 621–632.
- [41] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., New York, NY, USA, 2020, pp. 9459–9474, <https://dl.acm.org/doi/abs/10.5555/3495724.3496517>.
- [42] H. Li, Y. Su, D. Cai, Y. Wang, L. Liu, A survey on retrieval-augmented text generation, 2022, <http://dx.doi.org/10.48550/arXiv.2202.01110>.
- [43] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey, Preprint (2023) <http://dx.doi.org/10.48550/ARXIV.2312.10997>.
- [44] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, <http://dx.doi.org/10.48550/ARXIV.1301.3781>, Preprint.
- [45] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186, <http://dx.doi.org/10.18653/v1/N19-1423>.
- [46] A. Neelakantan, T. Xu, R. Puri, A. Radford, J.M. Han, J. Tworek, Q. Yuan, N. Tezak, J.W. Kim, C. Hallacy, J. Heidecke, P. Shyam, B. Power, T.E. Nekoul, G. Sastry, G. Krueger, D. Schnurr, F.P. Such, K. Hsu, M. Thompson, T. Khan, T. Sherbakov, J. Jang, P. Welinder, L. Weng, Text and code embeddings by contrastive pre-training, 2022, <http://dx.doi.org/10.48550/ARXIV.2201.10005>, Preprint.
- [47] Automotive Quality and Process Improvement Committee, Potential failure mode and effects analysis (FMEA) including design FMEA, supplemental FMEA-MSR, and process FMEA, 2021, http://dx.doi.org/10.4271/J1739_202101.
- [48] X. Chen, S. Jia, Y. Xiang, A review: Knowledge reasoning over knowledge graph, *Expert Syst. Appl.* 141 (2020) 112948, <http://dx.doi.org/10.1016/j.eswa.2019.112948>.
- [49] J. Camacho-Collados, M.T. Pilehvar, From word to sense embeddings: A survey on vector representations of meaning, *J. Artificial Intelligence Res.* 63 (2018) 743–788, <http://dx.doi.org/10.1613/jair.1.11259>.
- [50] E. Hüllermeier, W. Waegeman, Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods, *Mach. Learn.* 110 (3) (2021) 457–506, <http://dx.doi.org/10.1007/s10994-021-05946-3>.
- [51] D. Küpper, The future of battery production for electric vehicles, 2018, URL <https://www.bcg.com/publications/2018/future-battery-production-electric-vehicles>. visited on 11/01/2023.
- [52] R. Greene, T. Sanders, L. Weng, A. Neelakantan, New and improved embedding model, 2022, URL <https://openai.com/index/new-and-improved-embedding-model/>. visited on 03/10/2024.
- [53] A. Salemi, H. Zamani, Evaluating retrieval quality in retrieval-augmented generation, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 2395–2400, <http://dx.doi.org/10.1145/3626772.3657957>.
- [54] M. Yazdi, E. Zarei, S. Adumene, A. Beheshti, Navigating the power of artificial intelligence in risk management: A comparative analysis, *Safety* 10 (2) (2024) <http://dx.doi.org/10.3390/safety10020042>.
- [55] C. Shang, F. You, Data analytics and machine learning for smart process manufacturing: Recent advances and perspectives in the big data era, *Engineering* 5 (6) (2019) 1010–1016, <http://dx.doi.org/10.1016/j.eng.2019.01.019>.
- [56] M. Yazdi, Augmented reality (AR) and virtual reality (VR) in maintenance training, in: *Advances in Computational Mathematics for Industrial System Reliability and Maintainability*, Springer Nature Switzerland, Cham, 2024, pp. 169–183.