

# A knowledge-graph enhanced large language model-based fault diagnostic reasoning and maintenance decision support pipeline towards industry 5.0

Yunfei Ma<sup>a</sup>, Shuai Zheng  <sup>a</sup>, Zheng Yang<sup>a</sup>, Hongcheng Pan<sup>a</sup> and Jun Hong<sup>b</sup>

<sup>a</sup>School of Software Engineering, Xi'an Jiaotong University, Xi'an, People's Republic of China; <sup>b</sup>School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, People's Republic of China

## ABSTRACT

Industry 5.0 highlights the human-machine collaboration and the sustainability of intelligent manufacturing. Under this background, fault diagnosis, as a key technical component, imposes new requirements for efficient human-machine interaction. The ease of use and outstanding natural language processing capabilities of Large Language Models are believed to enhance the efficiency of human-machine interaction in fault diagnosis. But, LLMs usually exhibit limitations in their ability to incorporate new knowledge, the generation of hallucinations, and the transparency, rendering them unusable in the field of fault diagnosis. In this paper, we propose a novel fault diagnostic pipeline enhanced by knowledge graph, termed the Fault Diagnostic Reasoning Knowledge Graph LLM (FDRKG-LLM). This pipeline employs LLMs for complex fault diagnose tasks and construct knowledge graph to enhance the precise reasoning performance of the LLM. The effectiveness of the FDRKG-LLM is evaluated by a self-constructed product fault diagnose database. Experimental results demonstrate that the FDRKG-LLM outperforms existing retrieval-augmented generation models in assisting the analysis of mechanical equipment faults and providing reliable guidance. Hopefully, this research will pave the way for the widespread application of LLM-based solutions in the Industry 5.0.

## ARTICLE HISTORY

Received 27 June 2024  
Accepted 11 February 2025

## KEY WORDS

Domain-specific large language models; fault diagnostic reasoning; knowledge graph; mechanical equipment; industry 5.0

## SUSTAINABLE DEVELOPMENT GOALS

SDG 9: industry; innovation; and infrastructure

## 1. Introduction

With the ongoing development of industrial technology, the global manufacturing industry is advancing from industry 4.0 to industry 5.0 (Leng et al. 2023). Industry 4.0 refers to intelligence networking of machines and process for the Industry based on CPS (Cyber Physical Systems) (Grosse et al. 2023; Xu et al. 2021). It is a technology that achieves intelligent control using embedded network systems. Through automation, the Industrial Internet of Things (IIoT), and the deployment of intelligent systems, Industry 4.0 has realised smart manufacturing and data-driven production process automation. Unlike Industry 4.0, which focuses on the operation of automation and intelligent systems, Industry 5.0 places greater emphasis on the collaboration between human operators and intelligent systems, aiming to enhance human decision-making capabilities, personalised experiences, and ease of operation (B. Wang et al. 2024). This means that intelligent systems should not only have adaptive abilities but also collaborate with humans through natural interactions, assisting operators in completing complex tasks more easily and effectively. In the context, the fault diagnosis of mechanical equipment also faces new challenges

and opportunities. In real industrial scenarios, traditional fault diagnosis relies on expert experience, requiring experts to master the equipment mechanism and the content of standard maintenance manuals and to quickly locate faults based on the fault information. However, this process requires diagnostic personnel to have substantial knowledge reserves. To address the above problems, researchers propose a series of fault diagnosis methods based on knowledge graphs (C. Chen et al. 2023; Sui et al. 2023), utilising the knowledge graph to organise fault diagnosis knowledge in the form of a graph structure. However, these methods need to invoke several NLP sub-models, like named entity recognition and intent identification (M. Chen et al. 2023; Dagdelen et al. 2024). Constructing supervised datasets and models is a time-consuming and labour-intensive process (Han et al. 2022; Yun et al. 2019).

To achieve more efficient fault diagnosis, researchers are turning their attention to Large Language Models (LLMs). LLMs have powerful natural language understanding and generation capabilities. And it is able to process data such as sensor data, fault logs, and technical documentation from equipment, and interact with

operators through natural language. This aligns perfectly with the human-centric philosophy of Industry 5.0. For example, through natural language dialogue, LLMs can assist operators in obtaining fault information and diagnostic suggestions in the simplest manner, without requiring a deep technical background (Fraile et al. 2023). LLMs can serve as an intermediary layer between users and the database. It can understand user needs and use their powerful code generation capabilities to retrieve information from the fault diagnosis database for user. However, despite the significant potential of LLMs in human-machine interaction and intelligent diagnostics, they also have some inherent limitations that restrict their application in mechanical equipment fault diagnosis. **Hallucination** (Ji et al. 2023). The content generated by LLMs may be inconsistent with the real world facts or with the user's input. LLMs is essentially a statistical model, which cannot fundamentally solve this problem by itself. This poses a serious obstacles to the application of LLMs in the vertical domains such as healthcare (Alberts et al. 2023), finance (Wu et al. 2023) and biology (Leippold 2023; Luu and Buehler 2024). **Outdated Information.** The efficacy of LLMs is inherently limited by their training corpora, resulting in LLMs that rapidly become outdated as new information emerges. LLMs need to be retrained each time they are updated. This process is not only resource intensive, but also incurs significant costs. **Transparency.** LLMs also have been criticised for its black-box nature and lack of interpretability (Zhou et al. 2024), which are infeasible as LLMs implicitly encodes the knowledge embedded in the data in a parameterised form during the training process of LLMs.

Therefore, relying solely on the current general LLMs is insufficient for industrial applications. It is necessary to infuse specific domain knowledge into these general LLMs. To address this, this paper introduces a novel fault diagnostic pipeline enhanced by knowledge graph, termed the Fault Diagnostic Reasoning Knowledge Graph LLM (FDRKG-LLM). It utilises a general LLM to perform tasks like named entity recognition and intent recognition, such that users do not need to build datasets or train models for these tasks. FDRKG-LLM addresses the limitations of LLM in term of hallucination, outdated information, and explainability by using a structured fault diagnosis knowledge graph. FDRKG-LLM uses LLM as a natural language interface between the user layer and the data layer, allowing users to interact with the fault diagnosis knowledge graph using natural language. The fault diagnosis knowledge graph provides knowledge related to mechanical fault diagnosis for the LLM, improving the reliability of LLMs in the fault diagnosis field. FDRKG-LLM is a human-centric, knowledge-based fault diagnosis system that can extend

the capability boundaries of diagnosticians, enhancing decision-making ability. It is user-friendly even for individuals without a computer science background. In summary, the main contributions of this paper are listed as follows.

- (1) FDRKG-LLM, an application-specific framework in the field of mechanical product fault diagnostic reasoning, is proposed. LLM and KG are combined to optimise the fault diagnostic reasoning task framework.
- (2) FDRKG-LLM improves the accuracy and reliability of the fault diagnosis results by introducing a retrieval evaluator and empowering the LLM to use the Web search as an external data source.
- (3) Extensive experiments are conducted using novel evaluation methods such as BertScore, m3eScore, GPT-4 ranking and so on. The results show that FDRKG-LLM demonstrates state-of-the-art performance in the application of fault diagnostic reasoning.

Overall, our approach provides the technical foundation for facilitating collaboration between humans and artificial intelligence. We hope this research paves the way for the widespread application of LLM-based solutions in Industry 5.0.

## 2. Related work

FDRKG-LLM is a method that combines LLM and KG to address fault diagnosis in the context of Industry 5.0. Therefore, this section summarises the current popular fault diagnosis methods, the application of LLMs in the context of Industry 5.0, and the approaches of synergizing KGs and LLMs.

### 2.1. Approach of fault diagnosis

Fault diagnosis is crucial for enhancing the reliability and efficiency of mechanical equipment. Recently, researchers have developed a variety of methods. Existing fault diagnosis methods are mainly divided into three categories: methods based on physical-mathematical model analysis, data-driven methods, and knowledge-driven methods. In 1971, Beard et al. proposed a fault diagnosis method based on analytical models, which requires precise mathematical models and diagnoses the tested information according to specific mathematical methods (Beard 1971). Li et al. addresses the state and fault estimation problems for a class of quantised Takagi-Sugeno(T-S) fuzzy systems with sensor and actuator faults. Finally,

simulation examples of rigid-body satellite attitude control T-S fuzzy system are presented to verify the effectiveness of the proposed approaches (A. Li et al. 2022). methods based on physical-mathematical model analysis is appropriate for simple in structure, small-scale mechanical devices such as chips, integrated circuits, and aerospace components. However, it often has limitations in real production environments due to difficulties in accurately modelling mechanical equipment. In 1994, Askin et al. introduced a data-driven fault diagnosis approach. This method uses comparative analysis of online data and historical offline data stored in the system to predict fault occurrence and equipment status (Askin, Dror, and Vakharia 1994). Data-driven fault diagnosis methods involve analyzing vast amounts of data generated by industrial systems and applying machine learning algorithms to monitor and predict the health of complex industrial equipment, thereby guiding maintenance personnel to make decisions. Such data includes temperature, vibration, pressure, and other data collected by sensors, as well as electric current, voltage, and other signals generated during equipment operation. Data-driven fault diagnosis methods are mainly used before the occurrence of fault. And they do not depend on accurate analytical models and domain knowledge. The method employs black-box models like neural networks that do not explain the fault mechanism. Wang et al. proposed a fault diagnosis method based on the AlexNet convolutional neural network, achieving accuracies of 99.43 and 99.67% on the coal washing machine dataset and maintenance fault dataset, respectively (Z. Wang et al. 2021). Wen et al. proposed a novel convolutional neural network based on LeNet-5 for fault diagnosis, achieving significant results on the motor bearing dataset, self-priming centrifugal pump dataset, and axial piston hydraulic pump dataset (L. Wen et al. 2018). The knowledge-based method is a typical interactive query approach, which generally depend on an extensive fault knowledge base. The knowledge base contains fault modes, potential causes, and corresponding solution strategies for different types of equipment. This method is applied after a fault occurs. It aims to match the acquired fault description data with the information in the knowledge base to complete fault identification and diagnosis. Xia et al. develop a knowledge graph for maintenance of oil drilling equipment based on a defined ontological concept model and accumulated maintenance data and employed an Attention-Based Compressed Relational Graph Convolutional Network to predict potential solutions and elucidate faults within maintenance tasks (Xia et al. 2023). Martinez et al. propose a knowledge graph-based fault repair method that reasons about the current state of machines or processes for fault classification,

successfully diagnosing specific faults in electrical transformers and offering corresponding solutions (Martinez-Gil et al. 2022). However, completing a fault diagnosis based on a large-scale knowledge base is a knowledge-intensive task. The method firstly use NER to identify candidate entities. Then, entity and relationship retrieval are performed. The methods require the training of specific models for extraction and linking. When the field of application is changed, or the database increases, the model has to be re-trained, which is time-consuming and inefficient.

## 2.2. Large language model and industry 5.0

Recently, pre-trained language models based on Transformer have demonstrated superior performance in various NLP tasks. Studies show that the performance of the model increases as the parameter size grows. Large language models with parameters in the tens of billions or even hundreds of billions exhibit certain human-like characteristics, such as in-context learning and emergent abilities. LLMs have demonstrated tremendous application potential across various fields and expanded the boundaries of human capabilities, thereby enhancing human productivity. For example, a study by Noy and Zhang indicates that ChatGPT significantly enhances the productivity of workers in mid-level professional collaboration tasks (Noy and Zhang 2023). Another study by Brynjolfsson et al. show that the productivity of call centre operators using GAI (Generative artificial intelligence) increased by 14%, with less experienced employees seeing productivity gains of over 30% (Brynjolfsson, Li, and Raymond. 2023). LLMs can gain widespread popularity across various fields largely due to their applicability and interactivity. User can utilise LLMs to complete various tasks through simple conversational interactions, significantly lowering the technical barrier and allowing non-technical personnel to easily access and use these advanced AI technologies. This aligns with the human-centric characteristics of Industry 5.0. In the medical field, Dave et al. assessed the potential of LLMs in clinical text summarisation. The ensuing clinical reader suggests that LLM summaries tend to be more popular than medical expert summaries. LLM is expected to reduce documentation burden for clinical (Van Veen et al. 2024). Soroosh et al. explored the potential of ChatGPT Advanced Data Analysis (ADA) to close the knowledge gap between machine learning developers and clinical practitioners. the head-to-head comparison of the ChatGPT ADA-crafted ML models and their respective manually crafted counterparts revealed no significant differences in traditional performance metrics ( $p \geq 0.072$ ) (Tayebi Arasteh et al. 2024). In the agricultural

field, Zhao et al. integrated large language models, agricultural knowledge graphs, and graph neural network techniques to build a system for detecting *Elaeagnus angustifolia* diseases. This system provides robust technological tools for disease management and decision support in actual agricultural production (Zhao et al. 2024). To address the insufficient data analysis capabilities and limited autonomy of traditional agricultural robot systems, Zhang et al. proposed a novel large model generative intelligent agricultural robot system. The system can analyze vast amounts of agricultural data using LLMs to facilitate optimal decision-making and autonomous operational capabilities (Z. Zhang et al. 2023). In the field of smart manufacturing, Ilya et al. demonstrated that using refined GPT-3 Codex can generate effective simulations for queuing and inventory management systems, when provide with a verbal explanation. This work provides a technical foundation for the collaborative development of simulation models by humans and artificial intelligence (Jackson, Jesus Saenz, and Ivanov 2024). Li et al. introduced LLM4CAD, which utilises LLMs to generate CAD programmes that indirectly create 3D design objects by comprehending user needs. This method not only significantly lowers the technical barrier for CAD practitioners but also enhances their efficiency (for example, using LLMs can quickly process and generate a large number of repetitive 3D objects) (X. Li, Yuewan, and Zhenghui 2024).

Although LLMs have broad application potential across various fields, their success primarily relies on vast amounts of general data. This enables LLMs to excel in tasks such as code generation, text summarisation, and semantic understanding. First, LLMs are typically trained on publicly available datasets, which results in a lack of in-depth knowledge specific to certain industries. For instance, the specialised terminology, pathological knowledge, and clinical cases in the medical field are often absent from the training data of LLMs. Secondly, the training data for LLMs often lacks up-to-date information, which make them susceptible to falling behind in rapidly changing sectors like technology and finance. Therefore, to enhance the application effectiveness of LLMs in specific fields, injecting domain knowledge into LLMs becomes crucial.

### **2.3. The approaches of synergizing KGs and LLMs**

Knowledge graph (Xiao et al. 2023) represents a novel approach to database construction, distinguishing itself from structured databases such as Database and Data Table. Knowledge graph consists of a number of triples to store the factual knowledge. It facilitates explicit knowledge representation and provides interpretable reasoning

pathways (Y. Zhang et al. 2023). Knowledge graph has the following advantage: **Operability**. Knowledge graph supports the straightforward addition, deletion, and updating of knowledge. Operations on Knowledge graphs can be executed with precision down to individual pieces of knowledge, including specific entities and relations (Q. Zhang et al. 2023). **Real-time updating** (Guo, Cao, and Yi 2022). Knowledge graphs can be updated in real-time, ensuring the timeliness of the contained information even with the introduction of a single piece of data. **Traceability** (B. Chen and Bertozzi 2023). Each piece of knowledge utilised in answering queries within Knowledge graph is accompanied by its source, such as literature or hyperlinks, allowing users to assess the validity of the responses.

As shown in Table 1, due to the generality, generativity and semantic comprehension of LLMs, as well as the Operability, Real-time updating and traceability of knowledge graphs (Gilbert, Nikolas Kather, and Hogan 2024; Pan et al. 2024), an intriguing feature of LLMs and KGs is that they are complementary in many of their strengths and weakness. The complementarity opens the possibility of combining the approaches, to create a ‘dream team’ approach to finish fault diagnostic task (He et al. 2024).

**Hallucination:** The tendency of a system to generate false or unfounded information as if it were true. **Staleness:** The issue of a system providing information that is outdated and not reflective of the current context or data. **Non-Transparency:** The system’s inability to provide clear explanations or sources for the answers it gives, making it difficult to trace the origin of its conclusions. **Bias:** The inclination within a system’s output to reflect prejudiced or skewed perspectives, often manifesting in discriminatory or harmful content. **Usability:** The ease with which humans can interact with a system, including factors like user-friendliness and accessibility. **Evaluability:** The extent to which there are reliable and accepted methods available to assess the quality of a system’s outputs. **General-sense:** The breadth of scenarios or applications where a system can be effectively utilised. **Domain Capabilities:** The depth of a system’s expertise and processing abilities within a specific area or field. **Semantic Understanding:** The capability of a system to interpret and process the underlying meanings and complexities of natural language. **Suitability for fault diagnostic tasks:** The effectiveness of a system in aiding technicians with accurate diagnostics and decision-making for repairs and fault diagnosis.

Many researchers have explored the integration of LLMs with KGs. Think on Graph (Sun et al. 2023) treats the LLM as an agent, thinking step by step along the path of KGs. The use of beam search algorithm in the LLM

**Table 1.** Comparative analysis of using LLM alone, KG alone and in combination.

Property	LLM alone	Advantage (+)		Advantage (+) Disadvantage (-) Neutral (=)	Combination of LLM and KGs
		Disadvantage (-)	Neutral (=)		
Hallucination	High	—	None	+	Complementarity
Staleness	High	—	Neutral	+	None
Transparency	Low	—	High	+	Complementarity
Bias	High	—	Neutral	=	None
Usability	High	+	Low	—	Complementarity
Evaluability	Low	—	High	+	Complementarity
General-sense	Moderate	+	limited	—	Complementarity
Domain	Low	—	High	+	Complementarity
Capabilities					
Semantic-understanding	Limited to moderate	+	Limited	—	Complementarity
Suitability for fault diagnostic tasks	Only for low-risk tasks	—	Only tasks not requiring contextual reasoning	—	Complementarity potentially for moderate risk tasks needing contextual reasoning

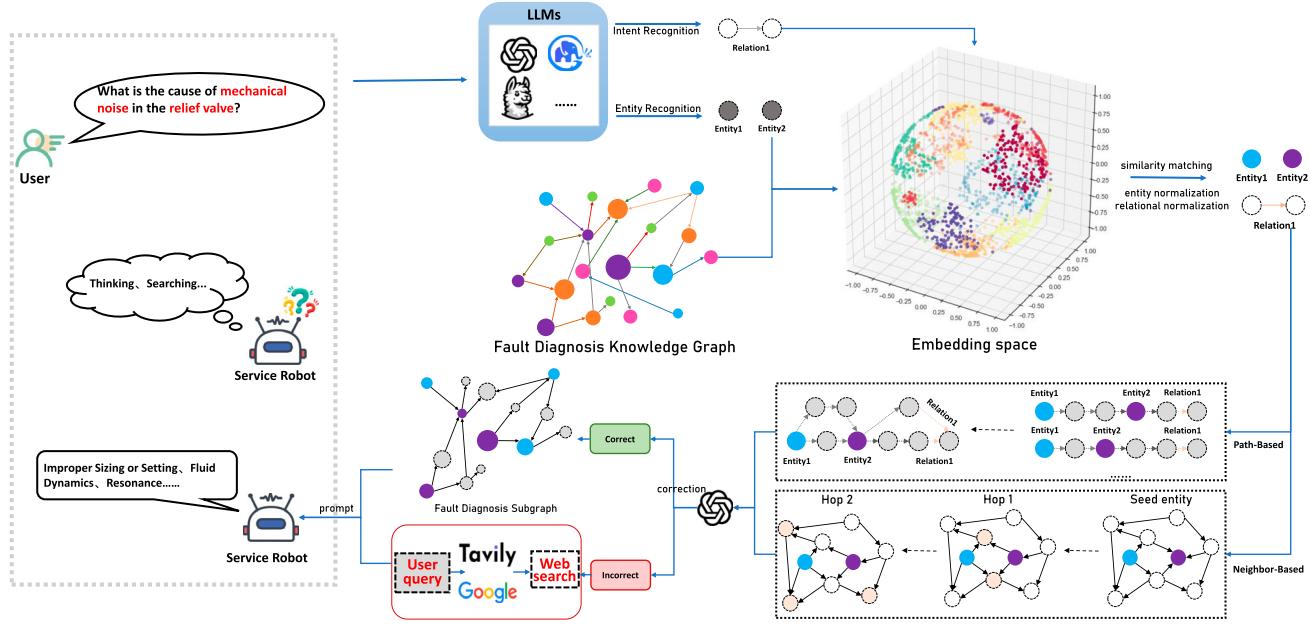
reasoning process allows LLM to dynamically explore multiple inference paths in the KGs and make appropriate decisions until a corresponding answer to the problem is found. KAPING (Baek, Fikri Aji, and Saf-fari 2023) firstly retrieves the relevant facts to the input question from the knowledge graph based on semantic similarities. Then it prepends the retrieved facts to the input question in the form of the prompt, which is then forwarded to LLMs to generate the answer. Wen et al. build a prompting pipeline that endows LLMs with the capability of comprehending KG inputs and inferring with a combined implicit knowledge and the retrieved external knowledge. Through prompt, LLMs construct MindMap and generate answers through external knowledge and knowledge subgraph, which improves the interpretability of the LLMs (Y. Wen, Wang, and Sun 2023). Wang et al. propose knowledge-tuning, which leverages structured medical knowledge bases for the LLMs to grasp domain knowledge efficiently and facilitate reliable response generation (H. Wang et al. 2023). Cheng et al. utilise a LLM to guide knowledge graph embedding by treating the triples in KG as text sequences, training the descriptions of relations and entities, and then using the description sequences as inputs for fine-tuning and judging the legitimacy of the triples. This method makes full use of the textual information to obtain a better performance of the knowledge graph representation (Cheng et al. 2023). Nurendra Choudhary et al. propose a novel method – Language-guided Abstract Reasoning over Knowledge graphs. It formulates complex knowledge graph reasoning as a combination of contextual knowledge graph search and logical query reasoning to take advantage of graph extraction algorithms and big data (Choudhary and Reddy 2023).

The above approach of using KG+LLM is mainly categorised into three kinds:(1) Transform the knowledge in the knowledge graph into text input to LLM. This

is a common method to construct a domain-specific LLM, but the method ignores the graph structure of the knowledge graph, and also suffers from problems such as incorrect retrieval and lengthy documents. (2) Constructing question-answer pairs using information from the knowledge graph to fine-tune LLM (Lester, Al-Rfou, and Constant 2021; X. L. Li and Liang 2021; Zaken, Ravfogel, and Goldberg 2021). Although this method can save computability to a certain extent and help construct knowledge graphs, the flexibility of this method is still low, because even if the large model is fine-tuned, the computability required is still large (Huang et al. 2023; Liu et al. 2022). (3) Multiple invocations of LLM make it dynamically explore pathways on the knowledge graph. This method can maximise the reasoning ability of the LLM, but it relies heavily on the construction of a high-quality, simple structure of the knowledge graph. The complex structure of the knowledge graph will make LLM cannot find the answer to the question, and the multiple invocations of LLM will make the burden on the server exponentially heavier.

### 3. Problem definition

The goal of this paper is to retrieve the corresponding fault diagnostic result from a stored fault case database using specific methods, given a newly submitted fault query described in natural language by the user. The specific method is achieved by combining external fault diagnosis knowledge base with LLM. Specifically, this paper injects knowledge from the external fault diagnosis knowledge graph into general LLM to improve its performance. Assume that  $q$  stands for the user question,  $a$  stands for the label corresponding to question  $q$ ,  $\emptyset_{LLM}$  stands for the adopted large language model, KG stands for the external knowledge base,  $A$  stands for the fault diagnosis result of LLM for a given user problem  $q$  and



**Figure 1.** Knowledge graph-enhanced fault diagnostic reasoning pipeline – FDRKG-LLM.

external knowledge base  $KG$ . The aim of our method is to minimise  $\|A - a\|$ , where  $A = \emptyset LLM(q, KG)$ .

## 4. Method

The overview of the proposed method in this paper is shown in Figure 1, which is mainly divided into four stages. The four parts are information extraction based on In-Context Learning, acquisition of fault diagnostic reasoning subgraph, correction of fault diagnostic reasoning subgraph and generating natural language answer by LLM based on the fault diagnostic subgraph.

### 4.1. Information extraction based on in-context learning

For information extraction in fault diagnostic reasoning task, the task is simplified into three phases. In the first phase, named entity recognition, our goals are to identify possible entities from the question. To enhance the entity recognition accuracy of LLMs, we employ In-Context Learning approach within the prompt construction. This involves samples  $\{Q_k, E_k\}$ , where each sample consists of user query  $Q_k$  and a corresponding set of entities  $E_k$  identified within that query. The samples, node type of knowledge graph and the user query are then input to LLM to ask the LLM to find the relevant entities in the user query. To facilitate access to the answer, the LLM is asked to output the entity  $E_q$  and the entity type  $E_t$  in form of a list (i.e.  $[[E_{q1}, E_{t1}], \dots, [E_{qn}, E_{tn}]]$ ). If the query does not contain an entity or the entity is not recognised by LLM, LLM will output none in response. The second

phase, intent recognition, is similar to the first stage of the methodology and focuses on identifying the types of entities in the knowledge graph that are most likely to find the answer to the given query. Utilising a method akin to the first stage, the LLM outputs the identified entity types as a list (i.e.  $[R_{r1}, \dots, R_{rk}]$ ). If the answer to the query is not recognised in the node type, LLM will respond by outputting none. In the third phase, Entity Normalisation and Relation Normalisation, the entities identified by LLM are aligned with entities in the knowledge graph. Specifically, we employ the m3e embedding model to encode  $E_q$  and  $E_{kg}$  within the knowledge graph into dense vector embeddings. Based on the entity type  $E_t$ , cosine similarity is used to align  $E_q$  with the  $E_{kg}$  in the knowledge graph as shown in equation (1). This process facilitates the linking of entities mentioned in the query to their corresponding entities within the knowledge graph, enhancing the integration of query information with structured knowledge.

$$\begin{aligned} \cos(E_q, E_t) &= \frac{E_q \cdot E_t}{\|E_q\| \times \|E_t\|} \\ &= \frac{\sum_{i=1}^n (E_q^i \times E_t^i)}{\sqrt{\sum_{i=1}^n (E_q^i)^2} \times \sqrt{\sum_{i=1}^n (E_t^i)^2}} \end{aligned} \quad (1)$$

In the case of complex queries, the simultaneous use of a mixture of vectors and keywords to retrieve nodes in the knowledge graph can greatly improve the retrieval accuracy.

## 4.2. Acquisition of fault diagnostic reasoning subgraph

Distinct from conventional database systems, KGs excel in structuring concepts, entities and their relational frameworks, thereby facilitating efficient retrieval capabilities. KGs support complex queries through flexible retrieval methods, including path-based search and neighbour-based search. This advanced approach enables the aggregation of dispersed, granular information, offering a more comprehensive understanding of data. In Section 4.1, define the standard entity as  $E_{qkg}$ , the entity type as  $E_{tkg}$ .  $E_{qkg} = \{[E_{qkg}^1, E_{qkg}^2, E_{qkg}^3, \dots], [E_{tkg}^1, E_{tkg}^2, E_{tkg}^3, \dots]\}$ . And define the intent recognition result of the question as  $R_r$ .  $R_r = [R_r^1, R_r^2, \dots]$ . Additionally, define the fault diagnostic knowledge graph by  $\omega = \{ \langle E, R \rangle \mid E \in E_{kg}, R \in R_{kg} \}$ . where  $E_{kg}$  and  $R_{kg}$  represent the set of entities and the set of relations in the knowledge graph, respectively.

### 4.2.1. Path-based fault diagnostic reasoning subgraph acquisition

The section connects entities in  $E_{qkg}$  by tracing their intermediary pathway within  $\omega$ . Further, a pseudo-code for the part is given as shown in Algorithm ??.

- (a) Choose one node in  $E_{qkg}^1$  as the start node  $v_1$ . Place the remaining nodes in a candidate node set  $E_{cand}$ . Explore at most  $k$  hops from  $v_1$  to find the next node  $v_2$ , where  $v_2 \in E_{cand}$ . If  $v_2$  is successfully reached within  $k$  hops, update the start node as  $v_2$  and remove  $v_2$  from  $E_{cand}$ . If  $v_2$  cannot be found within  $k$  hops, connect the segments of paths obtained so far and store them in  $G_{path}^n$ . Then, choose another node  $v'_1$  from  $E_{cand}$  as the new start node, and remove both  $v'_1$  and  $v_2$  from  $E_{cand}$ .
- (b) Check if  $E_{cand}$  is empty. If it is not empty, iterate (a) to find the next segment of the path. If it is empty, connect all segments to build a set of subgraph and put them into  $G_{path}^n$ . Then, the node type in  $R_r$  is used as the end entity. Finally, the entity and the corresponding entity type in  $E_{qkg}$  are used as the start node and the intermediary node to obtain  $G_{path}$ .

### 4.2.2. Neighbour-based fault diagnostic reasoning subgraph acquisition

The section aims to incorporate more query-relevant information into the fault diagnostic subgraph. (1) We find the neighbours  $E_{n1}^*$  of entity  $E_{n1}$  in  $E_{qkg}$  through 1-hop, where may be more than one adjacent node, and add the triples  $E_{n1}, R, E_{n1}^*$  to  $G_{neigh}$ . (2) We will use the m3e

---

### Algorithm 1: Path-based fault diagnostic reasoning subgraph acquisition

---

```

Input:  $E_{qkg} = \{E_{qkg}^1, E_{qkg}^2, \dots\}, R_r, k, \omega$ 
Output:  $G_{path} = [G_{path}^1, G_{path}^2, \dots, G_{path}^n]$ 
1 Select a random node  $v_1$  in  $E_{qkg}$  and set it as the starting node;
2 Set  $E_{cand} = E_{qkg} - \{v_1\}$   $G_{path}^n = [v_1]$ ;
3 while  $E_{cand} \neq \emptyset$  do
4   Explore  $k$  hops in the knowledge graph  $\omega$  with  $v_1$  as the starting node and
5   find any node  $v_2$  in  $E_{cand}$ ;
6   if  $v_2$  is found: then
7      $G_{path}^n.append(v_2);$ 
8     Remove  $v_2$  from  $E_{cand}$ ;
9     Update  $v_1=v_2$  ( $v_2$  becomes the new start node);
10  else
11     $G_{path}^n.append(R_r);$ 
12    Update  $G_{path}^n$  (Retrieve the entity nodes based on  $R_r$  and the connected entities, and store them in  $G_{path}^n$ );
13     $G_{path}.append(G_{path}^n);$ 

```

---

embedding model to check if  $E_{n1}^*$  is semantically related to the query. If so, we further expand the adjacent nodes of  $E_{n1}^*$  and add triples to  $G_{neigh}$ . Finally obtain the set of  $G_{neigh}$ ,  $G_{neigh} = \{G_{neigh}^1, G_{neigh}^2, \dots, G_{neigh}^{n2}\}$ .

### 4.2.3. Fuzzy-based fault diagnostic reasoning subgraph acquisition

In some cases, the entities in the user query are so obscure that only the corresponding entity type  $E_{tkg}$  can be detected. The standard entity cannot be extracted or only part of the entities  $E_{qkg}$  can be extracted. If the extracted information is not enough to construct the subgraph, then we get the entity types of the entities in the query in fault diagnostic knowledge graph schema. Replace the entities in the original question with the entity types, and then according to the relationships of relational normalisation, complete the edges of the query subgraph with the help of fault diagnostic KG schema to complete the construction of the query subgraph. However, the subgraph obtained by this method is very redundant, so we finally use cosine similarity to find the top-k paths that are most similar to the question as the output  $G_{fuzzy}$ ,  $G_{fuzzy} = \{G_{fuzzy}^1, G_{fuzzy}^2, \dots, G_{fuzzy}^{n3}\}$ .

### 4.3. Correction of the fault diagnostic reasoning subgraph

LLM is a type of generative model. The content generated by LLM has a strong dependence on the knowledge of fault diagnostic subgraph. If the fault diagnostic subgraph contains a large amount of irrelevant information, it will lead to difficulties in obtaining accurate knowledge for LLM. The irrelevant information may mislead the LLM and cause LLM to generate hallucinations. To minimise triple paths in the Fault Diagnostic subgraph that are not related to the user query, an approach is adopted where the query is systematically linked with each triple pathway across  $G_{path}$ ,  $G_{neigh}$ ,  $G_{fuzzy}$ . Subsequently, the LLM acts as a retrieval evaluator to evaluate the relevance of triple pathways to the user query. The correlation is divided into two confidence levels in correct and incorrect. If correct, it means that the retrieved triple path is relevant to the user's question and the triple path is retained. If it is incorrect, this indicates that the retrieved triple paths are all considered irrelevant, which does not help the generation then remove them. If none of the retrieved triples are relevant to the question, it is extremely important to seek complementary external knowledge. Large-scale web searches are integrated as a strategic extension of RAG using Tavily API. This corrective action helps to overcome the awkward challenge of not being able to refer to reliable knowledge.

### 4.4. Generating natural language answer by LLM based on the fault diagnostic subgraph

In order to get the accurate answer, we connect system message,  $G_{path}$ ,  $G_{neigh}$ ,  $G_{fuzzy}$  and question to form the LLM prompt and ask LLMs to generate the final answer. Usually  $G_{path}$  contains the most relevant path to the problem.  $G_{neigh}$  and  $G_{fuzzy}$  contain descriptions and additional information about the entities related to the problem. The combination of these triple pathways makes the prompt contain more comprehensive information. LLM uses a rich general corpus and knowledge in the pre-training phase. The relationship between knowledge and the reasoning process is not decoupled during the pre-training process of LLM. We are not sure that LLM fully understands the logical relationships behind the knowledge. However the reasoning paths extracted from the fault diagnostic knowledge graph contain both knowledge and reasoning processes, and thus can trigger and guide the LLM to generate the final answer. And the constraints on the path of reasoning through the graph can effectively mitigate the hallucination that exists in LLMs.

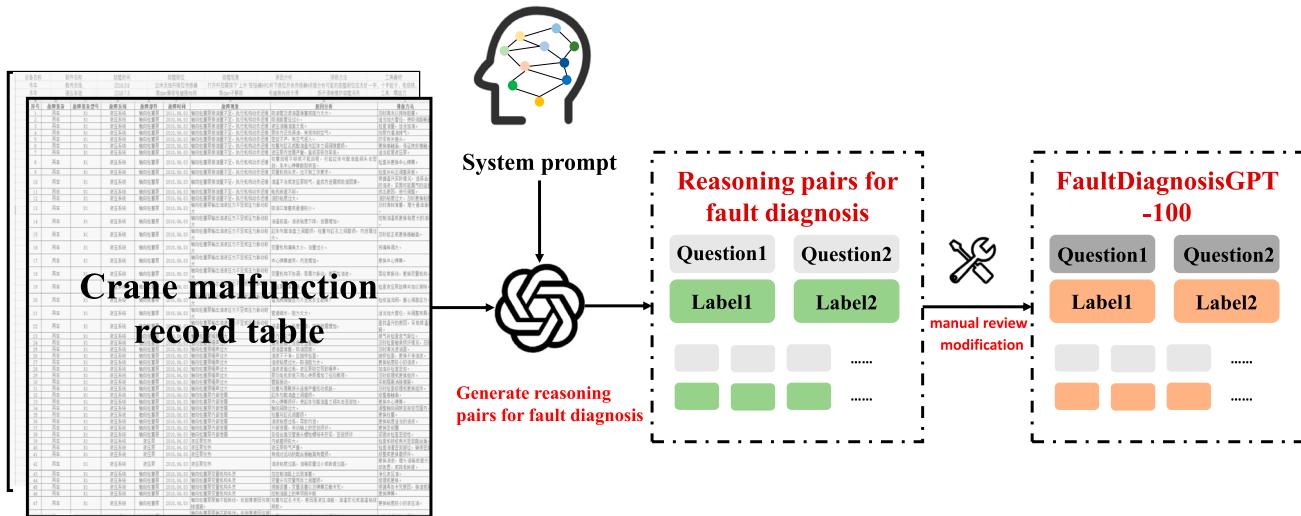
## 5. Experiments

In the section, we establish some baselines to evaluate our methods for a suite of reasoning tasks, as shown in Figure 3. To ensure that the fault diagnostic reasoning pairs can effectively accommodate the diverse questions that users might ask during their interactions with the reasoning system, we use real-world crane fault record table to generate a fault diagnosis dataset with GPT-4, as shown in the Figure 2. Firstly, we obtain a crane fault diagnosis record from a certain company, containing 255 records across three major systems: mechanical, hydraulic, and electronic. Then, we input the fault diagnosis record into GPT-4 and generated reasoning pairs via prompts. Next, to further structure the dataset, we manually review and edit the reasoning pairs, including steps such as data cleaning and deduplication. Consequently, we develop the specialised fault diagnosis dataset FaultDiagnosisGPT-100. All datasets are composed in Chinese.

As shown in Figure 3, we pick the vanilla GPT-3.5 and GPT-4 as the baselines that infer with only implicit knowledge. We then add three retrieval-augmented baselines: BM25 retriever, Embedding retriever, and Table retriever. We use GPT-4 as the backbone for all retrieval-based methods. To evaluate the performance of all method quantitatively, our employs some evaluative methods: m3eScore, BertScore (Devlin et al. 2018; T. Zhang et al. 2019), GPT-4 Rating (Y. Wen, Wang, and Sun 2023) and so on.

### 5.1. Collection of data

We utilise malfunction record tables for cranes. It originally formatted as Excel spread-sheets. These tables meticulously detail ten types of information: equipment, model number, system, the component involved, time of the malfunction, fault phenomenon, analysis, troubleshooting method, the unit and the location. Structured tabular data, distinct from text, represents a form of data not encountered by LLM during pretraining phases. Consequently, this might impede the LLMs' ability to fully comprehend such data, so we employ OpenAI's 'text-davinci-002' model to linearise and expand the tabular data using the column names of the tables as keywords, followed by manual review to obtain the textual records of fault diagnostic. Furthermore, to enhance the generalisability of our methods, the column names from the fault diagnosis tables are used as entity type to construct knowledge graph, as shown in Figure 4. The fault diagnosis knowledge graph currently contains 1341 entities and 1328 relationships.



**Figure 2.** Using LLM to assist in building datasets.

### 5.2. Evaluation metrics

We adapt generic metrics for evaluation, including BLEU, METEOR, BERTScore, M3eScore, and GPT-4 Ranking. The calculation of each metric is presented.

- **BLEU.** It evaluates performance by measuring the similarity between the generated text and a set of reference texts. The similarity is achieved by comparing the overlap of N-grams (i.e. consecutive sequences of n words) between the generated text and the reference texts. Commonly, BLEU considers n-grams of lengths 1 to 4, which means that it checks for matches of single words(unigrams), two consecutive words(bigrams), three consecutive words(trigrams), and four consecutive words(four-grams) corresponding to BLEU-1, BLEU-2, BLEU-3, BLEU-4. BLEU compares the N-gram in the LLM output text with the N-grams in the reference text to calculate the BLEU-N score. The calculation method is shown below:

$$\text{BLEU} - N = \frac{\sum_{N\text{-gram} \in \text{Gen}} \min(\text{count}_{\text{Gen}}(N\text{-gram}), \text{count}_{\text{Ref}}(N\text{-gram}))}{\sum_{N\text{-gram} \in \text{Gen}} \text{count}_{\text{Gen}}(N\text{-gram})}$$

*Gen* refers to the text generation output by the LLM; *Ref* refers to the reference text;

$\text{count}_{\text{Gen}}(N\text{-gram})$  represents the number of times an N-gram appears in the LLM-generated text;  $\text{count}_{\text{Ref}}(N\text{-gram})$  represents the number of times the N-gram appears in the reference text.

- **METEOR.** It is based on heuristic string matching and the reconciled mean of one-dimensional precision and recall. It computes exact match precision and exact match recall, while allowing fallback from exact unary matching to matching stems, synonyms and

paraphrases. The calculation method is shown below:

$$P = \frac{m}{W_{\text{Gen}}}$$

$$R = \frac{m}{W_{\text{Ref}}}$$

$$F_{\text{means}} = \frac{PR}{\alpha P + (1 - \alpha)R}$$

$$\text{METEOR} = (1 - pen) \times F_{\text{means}}$$

$W_{\text{Gen}}$  represents the number of words in the LLM output text;

$W_{\text{Ref}}$  represents the number of words in the reference text;

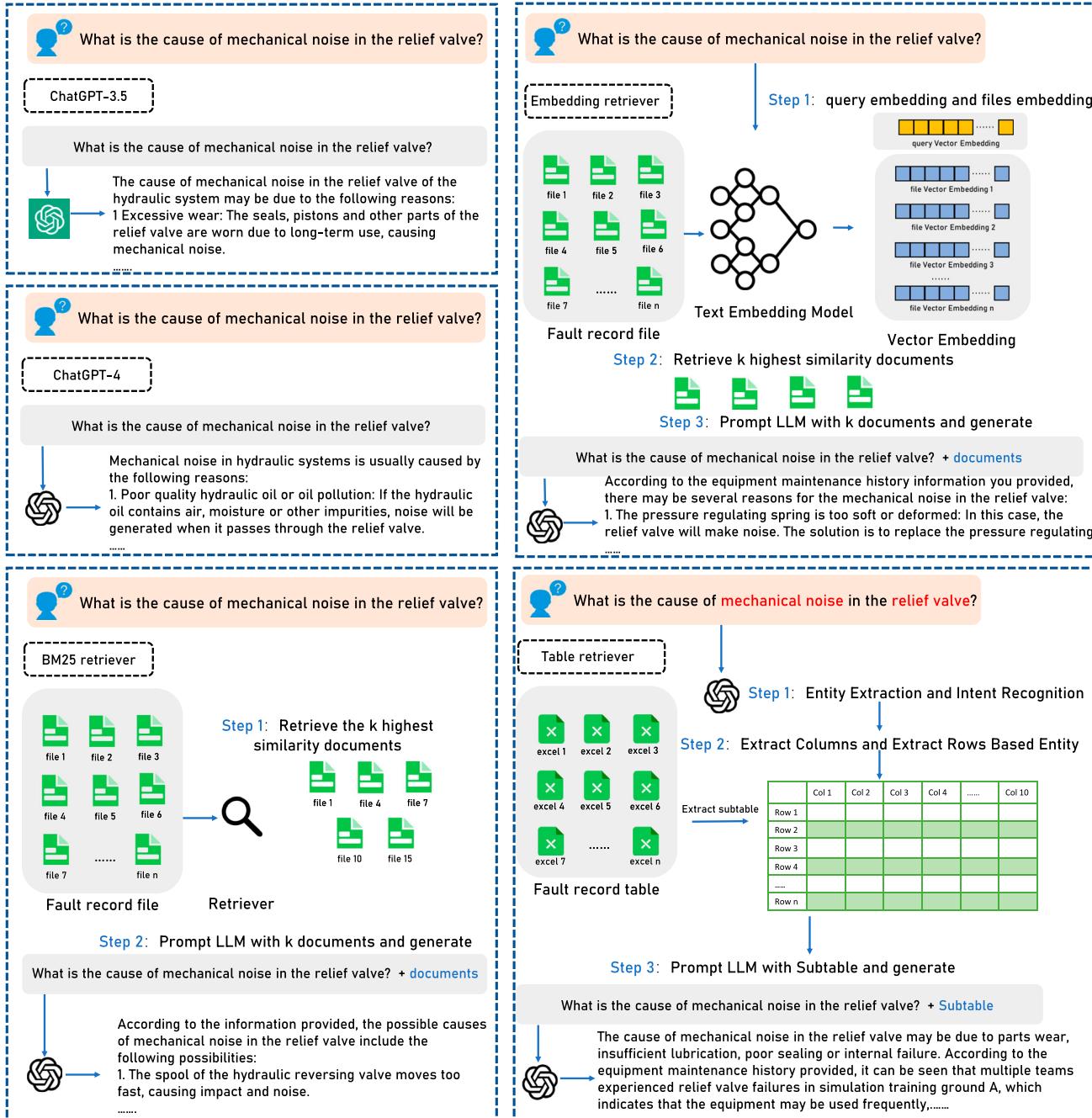
$m$  is the number of words common to the LLM output text and reference text;

$\alpha$  is the tunable hyperparameter;

*pen* is the penalty factor.

- **BERTScore.** It computes a similarity score for each token in the generated text with each token in the reference sentence. Unlike BLEU and METEOR, BERTScore computes token similarity using contextual embeddings.

Suppose that the text output from LLM is  $Y = [y_1, y_2, \dots, y_n]$ . The reference text is  $X = [x_1, x_2, \dots, x_n]$ . Respectively,  $[y_1, y_2, \dots, y_n]$  and  $[x_1, x_2, \dots, x_n]$  represent the token in  $Y$  and  $X$ . Given a tokenised reference sentence  $X = [x_1, x_2, \dots, x_n]$ , the embedding model generates a sequence of vectors  $\vec{X} = [\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n]$ . Similarly, the tokenised candidate  $Y = [y_1, y_2, \dots, y_n]$  is mapped to  $\vec{Y} = [\vec{y}_1, \vec{y}_2, \dots, \vec{y}_n]$ . The vector representation allows for a soft measure of similarity. The complete score matches each vector in  $\vec{X}$  to a vector in  $\vec{Y}$  to compute Recall. And the complete score matches each vector in  $\vec{Y}$  to a vector in



**Figure 3.** Traditional baseline methods: LLM only(ChatGPT-3.5 and ChatGPT-4), BM25 retriever + LLM, Embedding retriever + LLM, Table retriever + LLM.

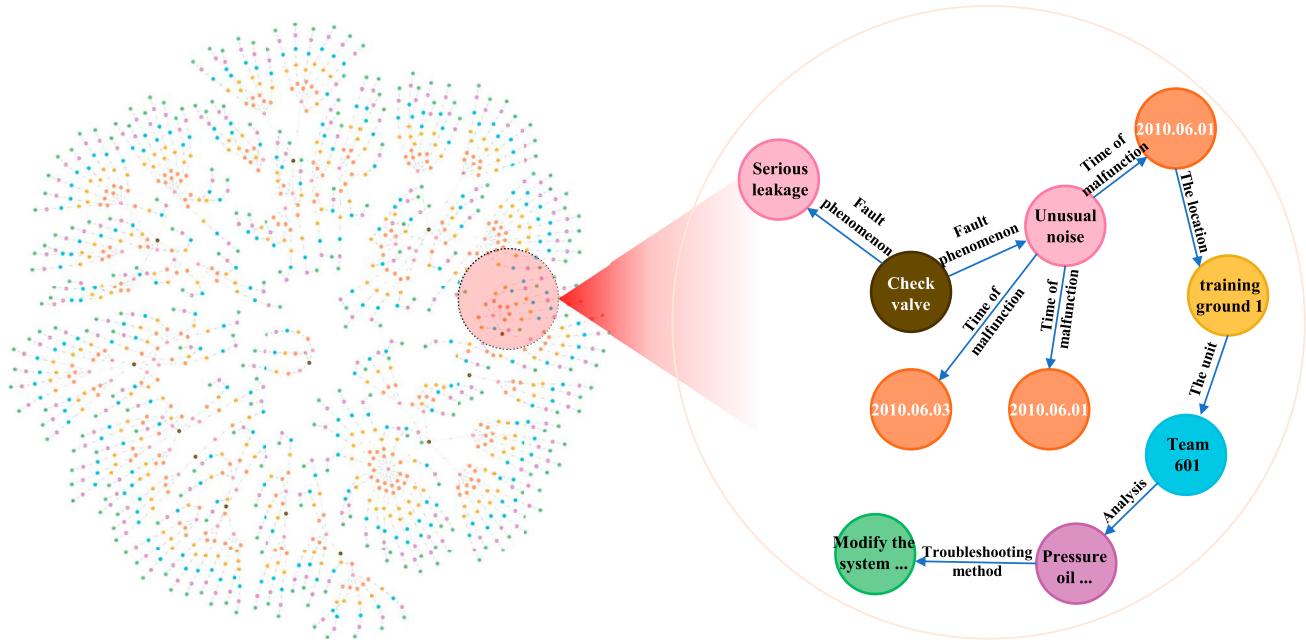
$\vec{X}$  to compute Precision. Use greedy matching to maximise the matching similarity score, where each vector is matched to the most similar vector in the other sentence. Combine Precision and Recall to compute an F1 measure.

$$R_{BERT} = \frac{1}{\|\vec{X}\|} \sum_{\vec{x}_i \in \vec{X}} \max_{\vec{y}_j \in \vec{Y}} \vec{x}_i^T \vec{y}_j$$

$$P_{BERT} = \frac{1}{\|\vec{Y}\|} \sum_{\vec{y}_j \in \vec{Y}} \max_{\vec{x}_i \in \vec{X}} \vec{x}_i^T \vec{y}_j$$

$$F1_{BERT} = 2 \frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}}$$

- **M3eScore.** It is similar to BERTScore, with the difference that the m3e model is used, because the m3e



**Figure 4.** Crane fault diagnostic knowledge graph.

mode is more suitable for Chinese corpus.

$$M3eScore = \frac{X_{Gen}^T \cdot X_{Ref}}{\|X_{Gen}\| \times \|X_{Ref}\|}$$

$X_{Gen}$  is the vector of LLM output text encoded by m3e model;

$X_{Ref}$  is the vector of reference text encoded by m3e model;

- **GPT-4 Ranking.** It used the GPT-4 model to rank the quality of answers generated by each searcher using real answers as a reference. And this method can simulate human judgement to the greatest extent possible.

### 5.3. Fault diagnostic reasoning test

We use FaultDiagnosisGPT-100 to test how LLMs deal with reasoning in fault diagnosis domain, and conduct tests using NER and IR results from GPT-4. The calculation of each metric is presented. We pick the vanilla GPT-3.5 and GPT-4 as the baselines that infer with only implicit knowledge. And we add three retrieval-augmented baselines: BM25 retriever (keyword-based retrieval) (Robertson, Zaragoza, and Taylor 2004), Text-Embedding retriever (semantic-based retrieval), Table retriever (table-understanding retrieval). We use GPT-4 as the backbone for all retrieval-based methods. When using ChatGPT, ChatGPT usually returns the table in Markdown format, which may be the reason why ChatGPT is heavily pre-trained with code from Github used for training, which is also the reason why Markdown format is used as the prompt input LLM in this section.

Table 2 show the performance of FDRKG-LLM and various baseline methods on FaultDiagnosisGPT-100. As can be seen from the Figure 5, the vanilla baseline GPT-3.5 and GPT-4 make the worst answers. This suggests that GPT-3.5 and GPT-4 do not introduce knowledge related to fault diagnosis in the c phase. Using only the implicit knowledge of the LLM may lead to seemingly reasonable but incorrect answers, further illustrating the importance of introducing an external knowledge base to mitigate the hallucination. The introduction of an external knowledge base has improved BERTScore, suggesting that LLM has successfully assimilated knowledge from the context to produce increasingly nuanced and semantically aligned output.

Adopting BM25, a retrieval method based on dictionary and keyword matching, has the worst effect compared to other retrieval methods although the indexes such as BETRScore are improved. It shows that it is difficult to retrieve the information related to the question using BM25, and even retrieve the information that is not related to the question. The accuracy of FDRKG-LLM is improved compared with Embedding Retriever and Table Retriever methods. The performance of FDRKG-LLM is better than Embedding Retriever performance, which indicates that structured ternary paths as part of the prompt can better stimulate the comprehension and reasoning ability of LLM than using chunks of documents.

On the one hand, although BM25 Retriever, Embedding Retriever and Table Retriever all introduce external knowledge bases via a passive retrieval mechanism,

<p><b>In simulated training ground A, what failure has occurred in the axial piston motor of the crane's hydraulic system?</b></p>	
<b>Question</b>	(a)
<b>Reference</b>	(b)
<b>Output</b>	Low output speed, low output torque, internal leakage, abnormal noise.
<p><b>GPT-3.5</b></p> <p>I am sorry, but I am unable to provide real-time information or fault reports for a specific simulator training field A. Axial piston motors in hydraulic systems can fail for a variety of reasons, such as hydraulic fluid contamination, damaged seals, leaks, pump failure, etc. A detailed inspection and diagnosis is recommended to determine the exact cause of the malfunction.</p> <p>Generally speaking, hydraulic system failure may manifest itself as symptoms such as abnormal crane movement, decreased speed, unusual noise, and abnormal temperature rise. The steps to solve the problem include checking the quality of the hydraulic oil, checking the working condition of each hydraulic component, and checking whether there is any leakage in the hydraulic pipeline. It is best to have experienced maintenance personnel or professional engineering.</p> <p><b>GPT-4</b></p> <p>I'm sorry, but I can't provide this information because I'm an AI and I don't have direct access to real-time events or details of specific failures of a particular device. What I can provide is information about some common failures that may occur with axial piston motors in crane hydraulic systems, such as leaks, pressure drops, inefficient operation, or the motor not working at all. Specific faults may require a detailed inspection by a service professional to determine.</p> <p><b>FDRKG-LLM without Corrective</b></p> <p>Based on historical records, failures of axial piston motors in Simulated Training Ground A include:</p> <ul style="list-style-type: none"> <li>1. Low output speed and low output torque</li> <li>-Trying to improve the oil supply according to the cause -Tightening the connecting bolts of each bonding surface and checking the sealing -Checking its damaged parts and repairing or replacing them</li> <li>2. Internal leakage</li> <li>-Repair and grind the oil distribution plate and cylinder end face.</li> <li>-Replace the spring - Replace the plunger pump - Replace the seal - Tighten the bolt.</li> <li>3. Abnormal sound</li> <li>-Reinstall or replace -Check sealing and take sealing measures -Replace clean fluid -Correct concentricity -Replace fluid with less viscosity -Repair or replace motor -Add isolation cover to isolate from external vibration sources</li> </ul> <p>Based on these histories, the main failures that have occurred with axial piston motors involve degradation of output performance, internal leakage, and abnormal noises. Troubleshooting methods involve ..... the addition of isolation shields.</p>	
<p><b>Embedding Retriever:</b> I am sorry, but I am unable to provide real-time information or fault reports for a specific simulator training field A. Axial piston motors in hydraulic systems can fail for a variety of reasons, such as hydraulic fluid contamination, damaged seals, leaks, pump failure, etc. A detailed inspection and diagnosis is recommended to determine the exact cause of the malfunction.</p> <p>Generally speaking, hydraulic system failure may manifest itself as symptoms such as abnormal crane movement, decreased speed, unusual noise, and abnormal temperature rise. The steps to solve the problem include checking the quality of the hydraulic oil, checking the working condition of each hydraulic component, and checking whether there is any leakage in the hydraulic pipeline. It is best to have experienced maintenance personnel or professional engineering.</p>	
<p><b>Table Retriever:</b> I'm sorry that I can't answer your question based on the information provided. If you can provide me with more information about the maintenance history of your equipment, I can help you analyze and answer the questions..</p>	
<p><b>BM25 Retriever:</b> I am sorry, but I am unable to provide real-time information or fault reports for a specific simulator training field A. Axial piston motors in hydraulic systems can fail for a variety of reasons, such as hydraulic fluid contamination, damaged seals, leaks, pump failure, etc. A detailed inspection and diagnosis is recommended to determine the exact cause of the malfunction.</p> <p>Generally speaking, hydraulic system failure may manifest itself as symptoms such as abnormal crane movement, decreased speed, unusual noise, and abnormal temperature rise. The steps to solve the problem include checking the quality of the hydraulic oil, checking the working condition of each hydraulic component, and checking whether there is any leakage in the hydraulic pipeline. It is best to have experienced maintenance personnel or professional engineering.</p>	

**Figure 5.** Example of the FDRKG-LLM instance.**Table 2.** Performance of all methods.

	BERTScore				m3eScore		GPT-4 Ranking ↓	
	Precision ↑	Recall ↑	F1Score ↑	BLEU-1 ↑	BLEU-2 ↑	Accuracy ↑		
FDRKG-LLM	<b>0.851</b>	<b>0.689</b>	<b>0.759</b>	<b>0.208</b>	<b>0.144</b>	<b>0.818</b>	<b>0.432</b>	<b>1.575</b>
ChatGPT3.5	0.727	0.611	0.662	0.045	0.005	0.792	0.094	5.425
ChatGPT4	0.722	0.601	0.655	0.037	0.004	0.790	0.082	4.575
BM25 retriever	0.740	0.620	0.672	0.071	0.014	0.791	0.128	4.0125
Embedding retriever	0.815	0.649	0.722	0.125	0.070	0.807	0.305	3.0375
Table retriever	0.833	0.661	0.735	0.147	0.097	0.817	0.348	2.375

which often leads to results burdened with excessive redundant information. FDRKG-LLM can rely on their graph structuring and path retrieval to obtain information with complete Semantic, thus reducing the problem of lengthy retrieval results. On the other hand, both table and knowledge graph are highly structured data, but LLM is trained on one-dimensional text (e.g. books and web-pages) while table is two-dimensional. natural language is one-directional that is meant to be read left-to-right, in a sequential manner. Distinct from pure text, tables deliver rich information through the interaction between rows and columns in the tabular structure, which enhances the data capacity but also increases the difficulty for language models to understand them. It may be another reason for the low accuracy of Table-Retriever.

#### 5.4. Ablation test

To further validate the effectiveness of correction of the fault diagnostic reasoning subgraph, experiments are conducted on the FaultDiagnosisGPT-100 dataset by removing and adding the corrective operation, and the results of the experiments are shown in the Table 3.

By adding corrective operations to each retriever, indicators such as BERTScore and BLEU-1 have been improved, indicating that adding corrective operations can improve the ability to automatically correct and efficiently utilise retrieved documents. Adding a network retrieval module can make LLM no longer limited to its own parameter knowledge, and further utilise network resources and optimise knowledge, which can also

**Table 3.** Results of ablation experiments.

	BERTScore				m3eScore		
	Precision ↑	Recall ↑	F1Score ↑	BLEU-1 ↑	BLEU-2 ↑	Accuracy ↑	Meteor ↑
FDRKG-LLM	<b>0.851</b>	<b>0.689</b>	<b>0.759</b>	<b>0.208</b>	<b>0.144</b>	<b>0.823</b>	<b>0.432</b>
FDRKG-LLM without Corrective	0.849	0.684	0.757	0.204	0.140	0.818	0.429
BM25 Retriever	0.740	0.620	0.672	0.071	0.014	0.791	0.128
BM25 Retriever with Corrective	0.743	0.625	0.678	0.077	0.015	0.796	0.138
Embedding Retriever	0.815	0.649	0.722	0.125	0.070	0.807	0.305
Embedding Retriever with Corrective	0.823	0.654	0.727	0.135	0.072	0.817	0.314
Table Retriever	0.833	0.661	0.735	0.147	0.097	0.817	0.348
Table Retriever with Corrective	0.834	0.664	0.737	0.150	0.097	0.819	0.362

**Table 4.** Intent Recognition results for different LLMs.

	Zero-shot %	One-shot %	Two-shot %	Three-shot %
ChatGPT-3.5	86.5	73.0	31.0	62.0
ChatGPT-4	98.5	100	99.5	100
Yi-34B	87.5	95.0	93.5	98.0
ERNIE	86.0	88.0	84.5	95.5
ChatGLM3	54.0	65.0	67.5	82.0

improve the effectiveness and robustness of LLM output to a certain extent.

## 6. Discussions

We provide an in-depth analysis of the impact of LLM on the FDRKG-LLM framework. NER(Named Entity Recognition) and IR(Intent Recognition) are the most fundamental and critical tasks within the realm of natural language processing (NLP). Constructing supervised datasets is a time-consuming and labour-intensive process (Wei et al. 2023; Xiao et al. 2024). LLM can transform all NLP tasks into text generation through prompt-learning. This approach eliminates the need for parameter adjustments. We evaluate the knowledge extraction capability of ChatGPT3.5, ChatGPT4 (Achiam et al. 2023), ChatGLM3 (Zeng et al. 2022), ERINE, and Yi-34B. Since LLM utilises its text generation capability for information extraction. It is traditionally based on BIO and BIEO to accomplish the named entity recognition task differently. The evaluation of the NER and IR tasks in this paper adopts a scoring system, with all correct scores being 1 point, partially correct scores being 0.5 points, and incorrect scores being 0 points. The results are shown in Tables 4 and 5. A zero-shot means that the LLM is prompted directly, a one-shot means that a case for the task is provided in the prompt and so on.

In the task of entity recognition and intent recognition, GPT-4 exhibits superior performance. In intent recognition, it achieves near-perfect accuracy without any samples. GPT-4 achieves an accuracy of 100% by one-shot learning. In the tasks of NER, GPT-4 demonstrates the highest capacity for entity extraction, with its

**Table 5.** Named entity Recognition results for different LLMs.

	Zero-shot %	One-shot %	Two-shot %	Three-shot %
ChatGPT-3.5	63.5	79.5	84.0	86.5
ChatGPT-4	86.5	82.5	89.5	91.5
Yi-34B	57.5	76.5	85.0	85.5
ERNIE	74.5	90.0	86.5	87.5
ChatGLM3	76.5	80.0	86.5	88.5

precision increasing as more samples are introduced. This increment in accuracy suggests that ChatGPT-4's strengths in both named entity recognition and intent recognition derive from its extensive knowledge base and superior context learning abilities.

ChatGLM3 exhibits continuous improvements in precision for intent recognition and entity extraction tasks as more samples are provided. For instance, in the intent recognition task, ChatGLM3's final accuracy reached 82%, marking a 28% increase from zero-shot learning. ERNIE achieves a peak accuracy of 95.5%, a 9% improvement from zero-shot learning, while Yi-34B reaches an accuracy of 98%, an increase of 10.5% from zero-shot learning. These results underscore the strong generalisation capabilities of LLMs, demonstrating their ability to enhance intent recognition based on user context rather than solely relying on the memorisation of training corpora. The peak accuracy of ChatGPT-3.5 from zero-learning reaches 86.5%, indicating that its context learning ability appears slightly inferior compared to other models. This pattern suggests that irrespective of the homogeneity of cases (memories) types, an excessive accumulation of cases (memories) may ultimately impede the LLM's learning efficacy and output quality. Determining the optimal capacity of cases for specific domain emerges as a pertinent inquiry for future research.

Additionally using the results of information extraction from each LLM, the results of completing the fault diagnostic reasoning are shown in the Table 6. As expected, due to the fact that the information extraction ability of ChatGPT4 is the best, FDRKG-ChatGPT4 is

**Table 6.** The impact of different LLM on the FDRKG-LLM framework.

	BERTScore			m3eScore			Meteor ↑
	Precision ↑	Recall ↑	F1Score ↑	BLEU-1 ↑	BLEU-2 ↑	Accuracy ↑	
FDRKG-ChatGPT3.5	0.831	0.690	0.746	0.188	0.129	0.803	0.401
FDRKG-ChatGPT4	<b>0.851</b>	<b>0.689</b>	<b>0.759</b>	0.208	<b>0.144</b>	<b>0.823</b>	<b>0.432</b>
FDRKG-ERNIE4	0.849	0.686	0.757	<b>0.211</b>	<b>0.144</b>	<b>0.823</b>	0.431
FDRKG-ChatGLM3	0.808	0.612	0.695	0.079	0.052	0.811	0.257
FDRKG-Yi-34B	0.812	0.616	0.699	0.072	0.048	0.793	0.251

able to find the most relevant entities to the user's query with relative accuracy, which motivates the LLMs to make use of the retrieved subgraphs of knowledge and improves the accuracy of the fault diagnostic reasoning. Other models are limited by their information extraction capability. In addition, we are surprised that although the information extraction ability of ERNIE4 is slightly worse than ChatGPT4, the results of FDRKG-ERINE4 and FDRKG-ChatGPT4 fault diagnostic reasoning are similar in all the metrics. It indicates that with the help of the special graph structure of the knowledge graph, the corresponding entities are retrieved through the associations between the entities. It illustrates the robustness of the FDRKG-LLM framework and stability.

## 7. Conclusion and future work

Our paper presents a human-centred industrial application referred to as Fault Diagnostic Reasoning Knowledge Graph LLM (FDRKG-LLM). Traditional handbook, expert system, and knowledge graph-based fault diagnose approaches utilise the exact fault description as inquiry to match the previously constructed database. In the era of Industry 4.0, traditional fault diagnosis systems often require a high degree of technical sensitivity and a deep familiarity with domain-specific structured query languages for data access and manipulation. FDRKG-LLM integrates fault diagnosis knowledge graphs with general LLMs. FDRKG-LLM employs LLM as a natural language interface between the user layer and the data layer, allowing users to express their queries in natural language, thereby enabling more natural and intuitive database interactions.

Fault diagnosis KG is used to provide knowledge support in the field of mechanical equipment for generic LLM, to make up for the lack of specialised domain knowledge in the corpus of LLM. Combining the advantages of KG and LLM, FDRKG-LLM solves the problem of semantic ambiguity and contextual understanding of KG in the face of natural language, and enables LLM to reason based on its own implicit knowledge and the retrieved knowledge subgraph, which improves the interpretability of LLM. In addition, for the cases of retrieval errors and not retrieving relevant triples, a web

search tool is introduced to use its retrieved results as a new external knowledge source, which empowers the LLM to actively conduct web searches, and improves the ability to efficiently utilise the retrieved documents and the robustness of the generation of the LLM. Through our proposed FaultDiagnosisGPT-100, we demonstrate that our approach, FDRKG-LLM, achieves remarkable gains over vanilla LLMs and retrieval-augmented generation methods. We hope that our method will pave the way for the widespread application of LLM-based solutions in the industrial sector.

Although this method has shown good responsiveness on the self-built FaultDiagnosisGPT-100 dataset, it still has certain limitations. The performance level of LLM is highly related to detailed fault diagnostic knowledge. In the future, FDRKG-LLM will integrate increasing amounts of data and knowledge to enhance the accuracy and efficiency of fault diagnosis systems. In this way, the amount of knowledge will be significantly increased and therefore the application performance will be enhanced. On the other hand, due to GPU resource limitations, the LLM used in this paper is a proprietary model. Although the performance of proprietary models is stronger compared to open-source models, proprietary models pose risks related to data security and privacy breaches. To address the problem, we will shift towards open-source large language models and expand the inclusion of high-quality fine-grain data. We will fine-tune open-source models to enhance the reliability and security of LLMs in more industrial scenarios.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Notes on contributors



**Yunfei Ma** is currently pursuing the Ph.D. degree in Computer Science and Technology at Xi'an Jiaotong University. His research interests include LLM and knowledge graph collaborative reasoning, and intelligent manufacturing.



**Shuai Zheng** received the BS, MS, and Ph.D degrees from Xi'an Jiaotong University in 2008, 2011 and 2017, respectively. He is now a professor in the School of Software Engineering at Xi'an Jiaotong University. His research interests include intelligent manufacturing, LLM-based production research, and knowledge-graph-based engineering.



**Zheng Yang** is currently pursuing a Master's degree in Software Engineering at Xi'an Jiaotong University. His research interests include intelligent process planning and large language models.



**Hongcheng Pan** is currently pursuing the Master of Engineering degree in Software Engineering at Xi'an Jiaotong University. His research interests include Optimization algorithm for workshop scheduling capacity.



**Jun Hong** is currently a Professor in Xi'an Jiaotong University, Xi'an, China. He received the Ph.D degree in mechanical engineering from Xi'an Jiaotong University in 2001. His research interests include intelligent manufacturing and digital manufacturing.

## Data availability statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## ORCID

Shuai Zheng  <http://orcid.org/0000-0003-3801-9359>

## References

- Achiam, Josh, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, et al. 2023. "Gpt-4 Technical Report." Preprint, arXiv:2303.08774.
- Alberts, Ian L., Lorenzo Mercolli, Thomas Pyka, George Prenosil, Kuangyu Shi, Axel Rominger, and Ali Afshar-Oromieh. 2023. "Large Language Models (LLM) and Chat-GPT: What Will the Impact on Nuclear Medicine Be?" *European Journal of Nuclear Medicine and Molecular Imaging* 50 (6): 1549–1552. <https://doi.org/10.1007/s00259-023-06172-w>.
- Askin, Ronald G., Moshe Dror, and Asso J. Vakharia. 1994. "Printed Circuit Board Family Grouping and Component Allocation for a Multimachine, Open-Shop Assembly Cell." *Naval Research Logistics (NRL)* 41 (5): 587–608. [https://doi.org/10.1002/\(ISSN\)1520-6750](https://doi.org/10.1002/(ISSN)1520-6750).
- Baek, Jinheon, Alham Fikri Aji, and Amir Saffari. 2023. "Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering." Preprint, arXiv:2306.04136.
- Beard, Richard Vernon. 1971. "Failure Accommodation in Linear Systems Through Self-Reorganization." PhD diss., Massachusetts Institute of Technology.
- Brynjolfsson, Erik, Danielle Li, and Lindsey R. Raymond. 2023. *Generative AI at Work*. Technical Report. National Bureau of Economic Research.
- Chen, Bohan, and Andrea L. Bertozzi. 2023. "AutoKG: Efficient Automated Knowledge Graph Generation for Language Models." In *2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy*, 3117–3126. IEEE. <https://doi.org/10.1109/BigData59044.2023.10386454>.
- Chen, Chong, Tao Wang, Yu Zheng, Ying Liu, Haojia Xie, Jianfeng Deng, and Lianglun Cheng. 2023. "Reinforcement Learning-Based Distant Supervision Relation Extraction for Fault Diagnosis Knowledge Graph Construction under Industry 4.0." *Advanced Engineering Informatics* 55:101900. <https://www.sciencedirect.com/science/article/pii/S1474034623000289>.
- Chen, Mingyang, Wen Zhang, Yuxia Geng, Zehong Xu, Jeff Z. Pan, and Huajun Chen. 2023. "Generalizing to Unseen Elements: A Survey on Knowledge Extrapolation for Knowledge Graphs." Preprint, arXiv:2302.01859.
- Cheng, Siyuan, Ningyu Zhang, Bozhong Tian, Xi Chen, Qingbing Liu, and Huajun Chen. 2023. "Editing Language Model-Based Knowledge Graph Embeddings." Preprint, arXiv:2301.10405.
- Choudhary, Nurendra, and Chandan K. Reddy. 2023. "Complex Logical Reasoning over Knowledge Graphs using Large Language Models." Preprint, arXiv:2305.01157.
- Dagdelen, John, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S. Rosen, Gerbrand Ceder, Kristin A. Persson, and Anubhav Jain. 2024. "Structured Information Extraction from Scientific Text with Large Language Models." *Nature Communications* 15 (1): 1418. <https://doi.org/10.1038/s41467-024-45563-x>.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. "Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding." Preprint, arXiv:1810.04805.
- Fraile, Francisco, Foivos Psaromatis, Faustino Alarcón, and Jordi Joan. 2023. "A Methodological Framework for Designing Personalised Training Programs to Support Personnel Upskilling in Industry 5.0." *Computers* 12 (11): 224. <https://doi.org/10.3390/computers12110224>.
- Gilbert, Stephen, Jakob Nikolas Kather, and Aidan Hogan. 2024. "Augmented Non-Hallucinating Large Language Models as Medical Information Curators." *NPJ Digital Medicine* 7 (1): 100. <https://doi.org/10.1038/s41746-024-01081-0>.
- Grosse, Eric H., Fabio Sgarbossa, Cecilia Berlin, and W. Patrick Neumann. 2023. "Human-Centric Production and Logistics System Design and Management: Transitioning from Industry 4.0 to Industry 5.0." *International Journal of Production Research* 61 (22): 7749–7759. <https://doi.org/10.1080/00207543.2023.2246783>.
- Guo, Quan, Shuai Cao, and Zhang Yi. 2022. "A Medical Question Answering System Using Large Language Models and Knowledge Graphs." *International Journal of Intelligent Systems* 37 (11): 8548–8564. <https://doi.org/10.1002/int.v37.11>.

- Han, Huihui, Jian Wang, Xiaowen Wang, and Sen Chen. 2022. "Construction and Evolution of Fault Diagnosis Knowledge Graph in Industrial Process." *IEEE Transactions on Instrumentation and Measurement* 71:1–12. <https://doi.org/10.1109/TIM.2022.3200429>.
- He, Zhe, Balu Bhasuran, Qiao Jin, Shubo Tian, Karim Hanna, Cindy Shavor, Lisbeth Garcia Arguello, Patrick Murray, and Zhiyong Lu. 2024. "Quality of Answers of Generative Large Language Models Versus Peer Users for Interpreting Laboratory Test Results for Lay Patients: Evaluation Study." *Journal of Medical Internet Research* 26:e56655. <https://doi.org/10.2196/56655>.
- Huang, Heyan, Xiao Liu, Ge Shi, and Qian Liu. 2023. "Event Extraction With Dynamic Prefix Tuning and Relevance Retrieval." *IEEE Transactions on Knowledge and Data Engineering* 35 (10): 9946–9958. <https://doi.org/10.1109/TKDE.2023.3266495>.
- Jackson, Ilya, Maria Jesus Saenz, and Dmitry Ivanov. 2024. "From Natural Language to Simulations: Applying AI to Automate Simulation Modelling of Logistics Systems." *International Journal of Production Research* 62 (4): 1434–1457. <https://doi.org/10.1080/00207543.2023.2276811>.
- Ji, Ziwei, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. "Survey of Hallucination in Natural Language Generation." *ACM Computing Surveys* 55 (12): 1–38. <https://doi.org/10.1145/3571730>.
- Leippold, Markus. 2023. "Thus Spoke GPT-3: Interviewing a Large-Language Model on Climate Finance." *Finance Research Letters* 53:103617. <https://doi.org/10.1016/j.frl.2022.103617>.
- Leng, Jiewu, Yuanwei Zhong, Zisheng Lin, Kailin Xu, Dimitris Mourtzis, Xueliang Zhou, Pai Zheng, Qiang Liu, J. Leon Zhao, and Weiming Shen. 2023. "Towards Resilience in Industry 5.0: A Decentralized Autonomous Manufacturing Paradigm." *Journal of Manufacturing Systems* 71:95–114. <https://doi.org/10.1016/j.jmsy.2023.08.023>.
- Lester, Brian, Rami Al-Rfou, and Noah Constant. 2021. "The Power of Scale for Parameter-Efficient Prompt Tuning." Preprint, arXiv:2104.08691.
- Li, Ang, Guangren Duan, Ming Liu, and Jingbo Fu. 2022. "Fault-Tolerant Quantized Sliding Mode Observers Design for a Class of Takagi-Sugeno Fuzzy System With Unmeasurable Premise Variable." *IEEE Transactions on Fuzzy Systems* 30 (7): 2312–2324. <https://doi.org/10.1109/TFUZZ.2021.3079956>.
- Li, Xiang Lisa, and Percy Liang. 2021. "Prefix-Tuning: Optimizing Continuous Prompts for Generation." Preprint, arXiv:2101.00190.
- Li, Xingang, Sun Yuewan, and Sha Zhenghui. 2024. "LLM4CAD: Multi-Modal Large Language Models for 3D Computer-Aided Design Generation." In *Proceedings of the ASME 2024 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 6: 36th International Conference on Design Theory and Methodology (DTM), Washington, DC, USA, August 25–28, 2024, V006T06A015*. ASME. <https://doi.org/10.1115/DETC2024-143740>.
- Liu, Haokun, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A. Raffel. 2022. "Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning." In *Advances in Neural Information Processing Systems*, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, 1950–1965. Vol. 35. New Orleans, LA: Curran Associates. [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/0cde695b83bd186c1fd456302888454c-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/0cde695b83bd186c1fd456302888454c-Paper-Conference.pdf).
- Luu, Rachel K., and Markus J. Buehler. 2024. "BioinspiredLLM: Conversational Large Language Model for the Mechanics of Biological and Bio-Inspired Materials." *Advanced Science* 11 (10): 2306724.
- Martinez-Gil, Jorge, Georg Buchgeher, David Gabauer, Bernhard Freudenthaler, Dominik Filipiak, and Anna Fensel. 2022. "Root Cause Analysis in the Industrial Domain Using Knowledge Graphs: A Case Study on Power Transformers." *Procedia Computer Science* 200:944–953. <https://doi.org/10.1016/j.procs.2022.01.292>.
- Noy, Shakked, and Whitney Zhang. 2023. "Experimental Evidence on the Productivity Effects of Generative Artificial Intelligence." *Science (NY)* 381 (6654): 187–192. <https://doi.org/10.1126/science.adh2586>.
- Pan, Shirui, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. "Unifying Large Language Models and Knowledge Graphs: A Roadmap." *IEEE Transactions on Knowledge and Data Engineering* 36 (7): 3580–3599. <https://doi.org/10.1109/TKDE.2024.3352100>.
- Robertson, Stephen, Hugo Zaragoza, and Michael Taylor. 2004. "Simple BM25 Extension to Multiple Weighted Fields." In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, 42–49. New York, NY: Association for Computing Machinery. <https://doi.org/10.1145/1031171.1031181>.
- Sui, Yicheng, Yuzhe Zhang, Jianjun Sun, Ting Xu, Shenglin Zhang, Zhengdan Li, Yongqian Sun. 2023. "LogKG: Log Failure Diagnosis through Knowledge Graph." *IEEE Transactions on Services Computing* 16 (5): 3493–3507. <https://doi.org/10.1109/TSC.2023.3293890>.
- Sun, Jiashuo, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. 2023. "Think-on-Graph: Deep and Responsible Reasoning of Large Language Model with Knowledge Graph." Preprint, arXiv:2307.07697.
- Tayebi Arasteh, Soroosh, Tianyu Han, Mahshad Lotfinia, Christiane Kuhl, Jakob Nikolas Kather, Daniel Truhn, and Sven Nebelung. 2024. "Large Language Models Streamline Automated Machine Learning for Clinical Studies." *Nature Communications* 15 (1): 1603. <https://doi.org/10.1038/s41467-024-45879-8>.
- Van Veen, Dave, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek. 2024. "Adapted Large Language Models Can Outperform Medical Experts in Clinical Text Summarization." *Nature Medicine* 30 (4): 1134–1142. <https://doi.org/10.1038/s41591-024-02855-5>.
- Wang, Zhijian, Wenlei Zhao, Wenhua Du, Naipeng Li, and Junyuan Wang. 2021. "Data-Driven Fault Diagnosis Method Based on the Conversion of Erosion Operation Signals into Images and Convolutional Neural Network." *Process Safety and Environmental Protection* 149:591–601. <https://www.sciencedirect.com/science/article/pii/S0957582021001300>.
- Wang, Haochun, Sendong Zhao, Zewen Qiang, Zijian Li, Nuwa Xi, Yanrui Du, MuZhen Cai 2023. "Knowledge-Tuning Large Language Models with Structured Medical Knowledge Bases



- for Reliable Response Generation in Chinese.” Preprint, arXiv:2309.04175.
- Wang, Baicun, Huiying Zhou, Xingyu Li, Geng Yang, Pai Zheng, Ci Song, Yixiu Yuan, Thorsten Wuest, Huayong Yang, and Lihui Wang. 2024. “Human Digital Twin in the Context of Industry 5.0.” *Robotics and Computer-Integrated Manufacturing* 85:102626. <https://www.sciencedirect.com/science/article/pii/S0736584523001011>.
- Wei, Xiang, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, et al. 2023. “Zero-Shot information extraction via chatting with chatgpt.” Preprint, arXiv:2302.10205.
- Wen, Long, Xinyu Li, Liang Gao, and Yuyan Zhang. 2018. “A New Convolutional Neural Network-Based Data-Driven Fault Diagnosis Method.” *IEEE Transactions on Industrial Electronics* 65 (7): 5990–5998. <https://doi.org/10.1109/TIE.2017.2774777>.
- Wen, Yilin, Zifeng Wang, and Jimeng Sun. 2023. “Mindmap: Knowledge Graph Prompting Sparks Graph of Thoughts in Large Language Models.” Preprint, arXiv:2308.09729.
- Wu, Shijie, Ozan Irsoy, Steven Lu, Vadim Dabrowski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. “Bloomberggpt: A Large Language Model for Finance.” Preprint, arXiv:2303.17564.
- Xia, Liqiao, Yongshi Liang, Jiewu Leng, and Pai Zheng. 2023. “Maintenance Planning Recommendation of Complex Industrial Equipment Based on Knowledge Graph and Graph Neural Network.” *Reliability Engineering & System Safety* 232:109068. <https://doi.org/10.1016/j.ress.2022.109068>.
- Xiao, Youzi, Shuai Zheng, Jiewu Leng, Ruibo Gao, Zihao Fu, and Jun Hong. 2024. “An Assembly Process Planning Pipeline for Industrial Electronic Equipment Based on Knowledge Graph with Bidirectional Extracted Knowledge from Historical Process Documents.” *Journal of Intelligent Manufacturing* 1–21. <https://doi.org/10.1007/s10845-024-02423-1>.
- Xiao, Youzi, Shuai Zheng, Jiancheng Shi, Xiaodong Du, and Jun Hong. 2023. “Knowledge Graph-Based Manufacturing Process Planning: A State-of-the-Art Review.” *Journal of Manufacturing Systems* 70:417–435. <https://doi.org/10.1016/j.jmsy.2023.08.006>.
- Xu, Xun, Yuqian Lu, Birgit Vogel-Heuser, and Lihui Wang. 2021. “Industry 4.0 and Industry 5.0-Inception, Conception and Perception.” *Journal of Manufacturing Systems* 61:530–535. <https://www.sciencedirect.com/science/article/pii/S0278612521002119>.
- Yun, Feng, Zhai Feng, Li Baofeng, and Cao Yongfeng. 2019. “Research on Intelligent Fault Diagnosis of Power Acquisition Based on Knowledge Graph.” In *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), Xiamen, China*, 1737–1740. IEEE. <https://doi.org/10.1109/EITCE47263.2019.9095018>.
- Zaken, Elad Ben, Shauli Ravfogel, and Yoav Goldberg. 2021. “Bitfit: Simple Parameter-Efficient Fine-Tuning for Transformer-Based Masked Language-Models.” Preprint, arXiv:2106.10199.
- Zeng, Aohan, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, et al. 2022. “Glm-130b: An Open Bilingual Pre-Trained Model.” Preprint, arXiv:2210.02414.
- Zhang, Yichi, Zhuo Chen, Wen Zhang, and Huajun Chen. 2023. “Making Large Language Models Perform Better in Knowledge Graph Completion.” Preprint, arXiv:2310.06671.
- Zhang, Qinggang, Junnan Dong, Hao Chen, Xiao Huang, Daochen Zha, and Zailiang Yu. 2023. “Knowgpt: Black-Box Knowledge Injection for Large Language Models.” Preprint, arXiv:2312.06185.
- Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. “Bertscore: Evaluating Text Generation with Bert.” Preprint, arXiv:1904.09675.
- Zhang, Zhijun, An Pan, Xingru Li, and Yamei Luo. 2023. “Large-Model and Generative-Intelligence Agricultural Robot Systems.” In *2023 International Annual Conference on Complex Systems and Intelligent Science (CSIS-IAC), Shenzhen, China*, 752–759. IEEE. <https://doi.org/10.1109/CSIS-IAC60628.2023.10363912>.
- Zhao, Xinyan, Baiyan Chen, Mengxue Ji, Xinyue Wang, Yuhua Yan, Jinning Zhang, Shiyingjie Liu, Muyang Ye, and Chunli Lv. 2024. “Implementation of Large Language Models and Agricultural Knowledge Graphs for Efficient Plant Disease Detection.” *Agriculture* 14 (8): 1359. <https://doi.org/10.3390/agriculture14081359>.
- Zhou, Bin, Xinyu Li, Tianyuan Liu, Kaizhou Xu, Wei Liu, and Jinsong Bao. 2024. “CausalKGPT: Industrial Structure Causal Knowledge-Enhanced Large Language Model for Cause Analysis of Quality Problems in Aerospace Product Manufacturing.” *Advanced Engineering Informatics* 59:102333. <https://doi.org/10.1016/j.aei.2023.102333>.

## Appendix 1. An example of query steps of the proposed FDRKG-LLM

When the user asks the question like:

What could be the cause of the heat in the hydraulic pump of a crane’s hydraulic system?

Step 1: As shown in Figure A1, The model generates a named entity recognition result for the user’s question under the following prompt input.

The result of named entity recognition is: [‘hydraulic system’, ‘system’], [‘crane’, ‘the component involved’], [‘hydraulic pump’, ‘the component involved’], [‘the heat’, ‘fault phenomenon’].

Step 2: As shown in Figure A2, the model generates a Intent recognition result for the user’s question under the following prompt input.

The result of Intent recognition is: [analysis] based on the named entity recognition and intent recognition results, the model searches the knowledge graph by performing Acquisition of fault diagnostic reasoning subgraph. As showed in Figures A3 and A4, the result of this procedure is an inverse ordered similarity list composed of graph nodes and their connections.

Step 4: given the most similar graph node connections, which is also referred as a fault diagnostic reasoning subgraph, the correction procedure is performed. As shown in Figure A5, the prompt template for correction of the fault diagnostic reasoning.

Then, the pruned fault diagnosis inference subgraph is obtained, as in Figure A6.

Step 5: the final process is to output the subgraph in the way of natural language. As shown in Figure A7, generating natural language answer by LLM based on the fault diagnostic subgraph.

Finally, the output result is shown in Figure A8.

When the user asks the question like:

What could be the cause of the heat in the hydraulic pump of a crane's hydraulic system?

Step 1: As shown in Fig.A1, The model generates a named entity recognition result for the user's question under the following prompt input.

```
def LLM_NER (question) :
    template = """"
        You are a named entity recognition model, now I will give you a user's question. Please
        recognize the entity in each sentence, according to my requirement.

        First, there are only ten types of entities: equipment, model number, system, the component
        involved, time of the malfunction, fault phenomenon, analysis, troubleshooting method, the unit
        and the location.

        Second, Please present this in the form of a list, where the first element of the list is the entity
        name and the second element is the entity type. The output format is shaped as: ["Entity name 1",
        "Entity type 1"], ["Entity name 2", "Entity type 2"], .... If the sentence does not contain the
        specified entity type, you can output: [None].

        Lastly, Please do not output any extra words other than this list.

        This sentence is: "{question}"\n\n
        """
    prompt=template.format(question=question)
    response=openai.ChatCompletion.create(
        model = "gpt-4",
        messages = [
            {"role": "system", "content": "You are NER model."}
            {"role": "user", "content": prompt}
        ]
    )
    response_NER = response.choices[0].message.content
    return response_NER
```

**Figure A1.** The prompt template for named entity recognition missions.



```

def LLM_RE (question) :
    template = """"
        You are a Intent Recognition model, I'm going to give you a question, and you need to reason
        from this user's question which node type the answer to this question is in.

        First, there are only ten types of entities: equipment, model number, system, the component
        involved, time of the malfunction, fault phenomenon, analysis, troubleshooting method, the unit
        and the location.

        Second, Please present this in the form of a list. The output format is shaped as: [“Entity type
        1”], [“Entity type 2”], .... Lastly, Please do not output any extra words other than this list.

        This sentence is: “{question}”\n\n
    """
    prompt=template.format(question=question)
    response=openai.ChatCompletion.create(
        model = “gpt-4”,
        messages = [
            {“role”: “system”, “content”: “You are RE model.”}
            {“role”: “user”, “content”: prompt}
        ]
    )
    response_RE = response.choices[0].message.content
    return response_RE

```

**Figure A2.** The prompt template for intent recognition missions.

```

Gpath=[
'Crane->system->hydraulic system->the component involved->hydraulic pump->fault phenomenon->the heat->time of the
malfunction->2010.06.03->the location->Simulation Training Ground A->the unit->Team 396->analysis->Higher internal wear and
tear',
'Crane->system->hydraulic system->the component involved->hydraulic pump->fault phenomenon->the heat->time of the
malfunction->2010.06.03->the location->Simulation Training Ground A->the unit->Team 397->analysis->Severe suction of the
hydraulic pump',
'Crane->system->hydraulic system->the component involved->hydraulic pump->fault phenomenon->the heat->time of the
malfunction->2010.06.03->the location->Simulation Training Ground A->the unit->Team 398->analysis->Wear on mating contact
surfaces with relative motion',
'Crane->system->hydraulic system->the component involved->hydraulic pump->fault phenomenon->the heat->time of the
malfunction->2010.06.03->the location->Simulation Training Ground A->the unit->Team 399->analysis->Excessive fluid viscosity,
low tank capacity or high speeds'
]

```

**Figure A3.** Path-based fault diagnostic reasoning subgraph acquisition.

```

G_neigh=[  

'Crane->system->mechanical system',  

'Crane->system->electrical system',  

'Crane->system->hydraulic system->the component involved->Axial Piston Pumps',  

'Crane->system->hydraulic system->the component involved->Axial Piston Motors',  

'Crane->system->hydraulic system->the component involved->Directional valve',  

'Crane->system->hydraulic system->the component involved->Throttle valve',  

'Crane->system->hydraulic system->the component involved->One-way cylinder',  

'Crane->system->hydraulic system->the component involved->Steering cylinder',  

'Crane->system->hydraulic system->the component involved->Sealing ring',  

'Crane->system->hydraulic system->the component involved->Accumulator',  

'Crane->system->hydraulic system->the component involved->Cooler',  

'Crane->system->hydraulic system->the component involved->relief valve',  

'Crane->system->hydraulic system->the component involved->oil filter',  

'Crane->system->hydraulic system->the component involved->Fully hydraulic steering',  

'Crane->system->hydraulic system->the component involved->Hydraulic pump->fault phenomenon->the heat->time of the  

malfunction->2010.06.03'  

]

```

**Figure A4.** Neighbour-based fault diagnostic reasoning subgraph acquisition.

```

def LLM_Correction (question, subgraph_path) :  

    template = """""  

        Suppose you are a rater responsible for evaluating the relevance of retrieved documents to a  

user's problem.  

        The following information is retrieved from the Fault Diagnostic Reasoning Knowledge Graph.  

{instruction}  

        Here are the questions asked by users.{question}  

        Rate a document as relevant if it contains information that would be helpful in answering the  

question.  

"""""  

    prompt=template.format(question=question, instruction=subgraph_path)  

    response=openai.ChatCompletion.create(  

        model = "gpt-4",  

        messages =[  

            {"role": "system", "content": "You are NLP model."},  

            {"role": "user", "content": prompt}  

        ])  

    response_Correction = response.choices[0].message.content  

    return response_Correction

```

**Figure A5.** the prompt template for Correction of the fault diagnostic reasoning subgraph.



```

 $G_{sub} = [$ 
    'Crane->system->hydraulic system->the component involved->hydraulic pump->fault phenomenon->the heat->time of the
    malfunction->2010.06.03->the location->Simulation Training Ground A->the unit-> Team 396->analysis->Higher internal wear
    and tear',
    'Crane->system->hydraulic system->the component involved->hydraulic pump->fault phenomenon->the heat->time of the
    malfunction->2010.06.03->the location->Simulation Training Ground A->the unit-> Team 397->analysis->Severe suction of the
    hydraulic pump',
    'Crane->system->hydraulic system->the component involved->hydraulic pump->fault phenomenon->the heat->time of the
    malfunction->2010.06.03->the location->Simulation Training Ground A->the unit-> Team 398->analysis->Wear on mating
    contact surfaces with relative motion',
    'Crane->system->hydraulic system->the component involved->hydraulic pump->fault phenomenon->the heat->time of the
    malfunction->2010.06.03->the location->Simulation Training Ground A->the unit->Team 399->analysis->Excessive fluid
    viscosity, low tank capacity or high speeds'
 $]$ 

```

**Figure A6.** Fault diagnostic reasoning subgraph after pruning.

```

def Finally_Fault_Diagnostic_Reasoning (question, subgraph) :
    template = """
        Suppose you are a good engineer and you are responsible for maintaining equipment.
        The following information is retrieved from the Fault Diagnostic Reasoning Knowledge Graph.
        They follow the format entity->relationship->entity
        {instruction}
        Here are the questions asked by users. {question}
        Please answer the question based on the above path information of the knowledge graph and
        the your own information.""""
    prompt=template.format(question=question, instruction=subgraph)
    response=openai.ChatCompletion.create(
        model = "gpt-4",
        messages =
            [
                {"role": "system", "content": "You are engineer."},
                {"role": "user", "content": prompt}
            ]
    )
    Finally_Fault_Diagnostic_Reasoning = response.choices[0].message.content
    return Finally_Fault_Diagnostic_Reasoning

```

**Figure A7.** Generating natural language answer by LLM based on the fault diagnostic subgraph.

**FDRKG-LLM:**

The following malfunctions have occurred with the axial piston motor in Simulated Training Ground A:

1. abnormal sounds
2. internal leakage
3. low output speed and low output torque

**Figure A8.** The fault diagnosis reasoning results.

Copyright of International Journal of Production Research is the property of Taylor & Francis Ltd and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.