Root Cause Analysis for Industrial Process Anomalies through the Integration of Knowledge Graph and Large Language Model

Qi Sun[1], Yahui Li[1], Chunjie Zhou[1], Yu-Chu Tian[2]

1. Huazhong University of Science and Technology, Wuhan 430074, P. R. China
E-mail: cjiezhou@hust.edu.cn

2. Queensland University of Technology, Brisbane 4001, Australia

Abstract: Root cause analysis for industrial process anomalies is critical for manufacturing activities. Industrial process alarms can provide crucial information to enable root cause analysis. However, the complex system structure causes a large number of alarms to emerge at the same time. To address this issue, we proposed an approach that utilizes knowledge graphs and large language models to provide comprehensible root cause analysis. Firstly, we extract knowledge such as historical anomalies from catalytic cracking operation manuals to construct an industrial process safety knowledge graph. Then, named entities in each alarm are extracted as keywords to retrieve factual knowledge from the knowledge graph. Finally, factual knowledge will be provided to the large language model as prior knowledge to infer the root cause of anomalies. Experimental results show that the proposed approach can accurately identify the root cause, thereby ensuring the safety of industrial processes.

Key Words: Root cause analysis, Knowledge graph, Large language model, Named entity recognition

1 Introduction

As a major backbone of a country, the safe and stable operation of the process industry is of paramount importance [1]. Its safety manufacturing is supported by the use of multiple sensors and safety management systems [2]. Due to the complexity of the crafting process and the composition of systems, a small deviation will cascade between several systems, resulting in a lot of alarms. Additionally, safety managers with varying quality levels sometimes cannot effectively and rapidly identify the root cause of anomalies. A large number of inner interrelated alarms makes safety managers tired of processing them, and the delayed processing of important alarms may even lead to irreparable losses. Therefore, a critical issue in present industrial safety production is how to comprehensively analyze anomalies by integrating a large amount of alarms, and ensure that safety managers can promptly and accurately identify the root cause of anomalies.

Root cause analysis research at present may be divided into three categories: traditional approaches utilizing expert systems and logic trees, deep learning approaches, and hybrid approaches [3]. Traditional approaches excel in providing interpretive explanations [4]. However, these approaches have a low degree of reasoning capacity and demand a high amount of domain knowledge. Since deep learning has become more popular recently, several researchers have tried to utilize it to build inference models for root cause analysis [5, 6]. Deep learning, a powerful representational tool, has the capacity to reason and abstractly represent knowledge. However, there are still several problems that need to be fixed with deep learning models, including their lack of interpretability and uncontrollable behavior. Some scholars have attempted to combine both approaches in order to better incorporate the benefits of each approach and increase their capacity to identify root causes. However, there is still a need for further text data mining for anomalies. Therefore, it is important to seek a root cause analysis approach that has minimal domain knowledge requirements, high interpretability, and a strong ability to reason in order to conduct root cause analysis more efficiently.

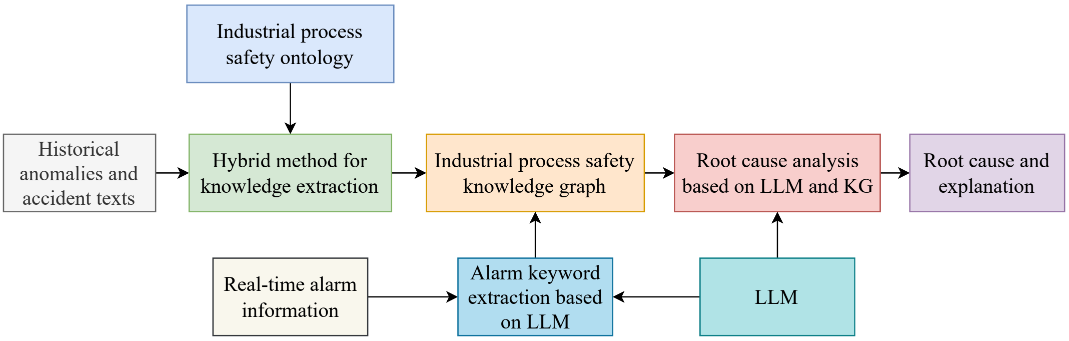
Historical anomalies involving industrial processes and other knowledge can provide data support for root cause analysis. At the same time, the growth of the knowledge graph and large language model presents novel approaches to using such knowledge efficiently [7, 8]. This paper proposes a root cause analysis approach that combines large language models with knowledge graphs, which can improve the efficiency of safety managers in processing alarms and provide timely, reasonable, and clear anomaly resolution solutions for safety managers. Firstly, we used historical anomalies and other items of the catalytic cracking process to build an industrial process safety knowledge graph. Secondly, we developed a named entity recognition module using a large language model. This module is used to recognize devices, states, physical quantities, and materials in the text. Finally, we utilize the named entity recognition module to search for information related to the current alarms in the industrial process safety knowledge graph, which will be used as the large language model's prior knowledge. The root cause of anomalies is discovered with the use of the text inference ability of large language models.

The main contributions of this paper are as follows:

1) Construct an industrial process safety knowledge graph using industrial process operation manuals, which supports conducting root cause analysis by integrating all alarms.
2) Utilize large language models to extract named entities from industrial process alarms, and then use the obtained entities to retrieve related facts from the knowledge graph as prior knowledge for the large language model.
3) Combine large language models with knowledge graphs to the root cause analysis for industrial process anomalies, which possesses excellent interpretability and powerful reasoning capabilities.

The rest of the paper is organized as follows. Section 2 introduces the root cause analysis approach. Section

This work is supported by National Natural Science Foundation (NNSF) of China under Grant 62127808.



Industrial process safety ontology
Historical anomalies and accident texts
Hybrid method for knowledge extraction
Industrial process safety knowledge graph
Root cause analysis based on LLM and KG
Root cause and explanation
Real-time alarm information
Alarm keyword extraction based on LLM
LLM

Fig. 1: The overall structure of the proposed root cause analysis approach

3 presents the experimental results along with illustrative descriptions. Finally, the conclusion is expounded in Section 4.

2 Framework of root cause analysis

In this section, we present an approach to root cause analysis, which can be divided into three steps. To begin with, we extract historical anomalies and accidents from the catalytic cracking operation manuals to construct an industrial process safety knowledge graph. Due to the varying formats of the data to be extracted in the catalytic cracking operation manuals, different data extraction algorithms are applied. The PDF table reading tools are applied to extract historical anomalies and normal operation statuses of devices. Regular matching is applied to extract historical accidents. To achieve high-precision device extraction and ensure the quality of the industrial process safety knowledge graph, we don't employ the named entity recognition approach but instead adopt a special approach. An approach based on rules, part-of-speech tagging, and word frequency statistics is applied to extract devices. By integrating these extracted structured data with the industrial process safety ontology, an industrial process safety knowledge graph can be constructed. The industrial process safety knowledge graph can provide information related to anomalies and offer knowledge support for the root cause analysis of anomalies. Secondly, in order to retrieve relevant knowledge about alarms in the industrial process safety knowledge graph, we propose an approach that utilizes a large language model for named entity recognition. Named entity recognition is applied to extract keywords from alarms, and then relevant nodes in the industrial process safety knowledge graph are retrieved based on these keywords. The factual triplets formed by these nodes are provided as prior knowledge to the large language model. Finally, the large language model serves as an inference engine, leveraging its ability to learn prior knowledge and analyze the causal relationships between alarms. By examining these causal relationships, the model can deduce the underlying root cause of anomalies. Figure 1 shows the overall structure of the proposed root cause analysis approach.

2.1 Knowledge graph construction

The operation manual, as an essential text resource for industrial production safety managers, offers comprehensive information on special systems such as control, process, and historical anomalies. To describe historical anomalies and other system safety information in industrial processes in an organized manner, we apply several natural language processing algorithms to extract useful data from raw unstructured text. Combined with our designed ontology, these structured data are used to construct an industrial process safety knowledge graph. Considering that in industrial processes, devices often have multiple mentions, alias tables are introduced for entity linking. Figure 2 shows the overall process of constructing the industrial process safety knowledge graph. Next, we will provide a detailed introduction to the approaches of constructing the industrial process safety knowledge graph using the operation manual.

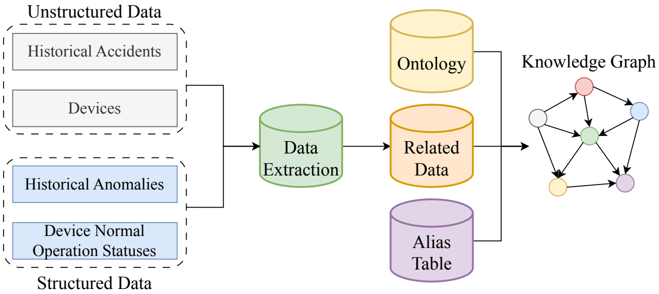


Fig. 2: The construction process of the industrial process safety knowledge graph

2.1.1 Ontology design

An ontology is a structured and unambiguous representation of concepts in a particular field of interest. It includes a set of classes or concepts and their properties, which describe the features and attributes of each concept. The relations between classes and properties are also represented, along with any constraints or restrictions that apply to them [9].

We created the industrial process safety ontology in order to conduct the root cause analysis based on the historical anomalies and historical accidents in the operation manual. The ontology we created contains six classes: anomaly, cause, phenomenon, solution, accident, and device. In the design of industrial process safety ontology, we not only simply record the anomalies and their corresponding causes, but also include other additional information. This information can provide additional knowledge for root cause analysis, stimulating the reasoning ability of large language models, thereby enabling more accurate root cause analysis.

In addition, we expect to provide corresponding solutions and phenomenons while offering the root cause, aiming to assist safety managers in promptly addressing anomalies. The industrial process safety ontology revolves around anomalies as its core, with all other types of entities linked to anomalies. By locating a specific anomaly, one can retrieve all the relevant information associated with that anomaly. Figure 3 shows the overall structure of the industrial process safety ontology, which includes a total of six classes and five relations.

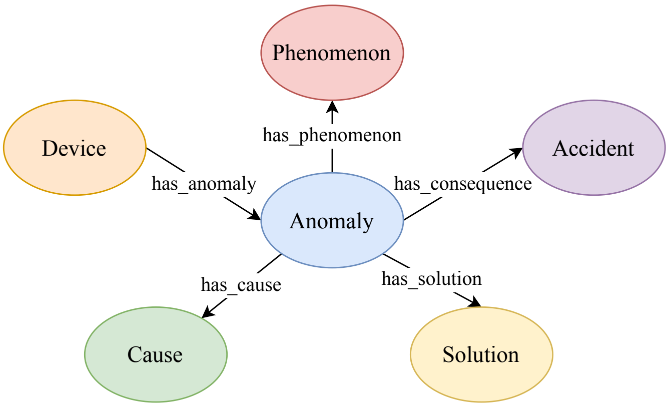


Fig. 3: The industrial process safety ontology

2.1.2 Data extraction

The extraction of structured information from unstructured data, including entities, relations, objects, and events, is the core goal of the data extraction process [10]. To construct the industrial process safety knowledge graph, we aim to extract four types of information: the name of a device, the normal operation status of a device, historical anomaly, and historical accident. The data in the industrial processes typically contains a high number of professional abbreviations and industry information confidentially, making it difficult to ensure the performance of large language models for data extraction. Because the quality of the fundamental data is critical to the performance of root cause analysis tasks, this paper employs different natural language processing algorithms paired with human verification to extract the data.

The PDF document of the catalytic operation manual used as the data source is written in Chinese. In this document, historical accidents are recorded in a specific format of text. In contrast, historical anomalies are recorded in tabular form, making them easier to extract. The normal operating statuses of devices are also recorded in tabular form. However, the names of devices are scattered throughout the text and are not presented in a structured format. Therefore, it is necessary to employ different data extraction algorithms for different types of data since they are supplied in various forms.

For historical accidents, which are texts with specific formats, we use regular matching to extract the content and parse it to obtain structured historical accident data. For data stored in tabular format, we utilize a PDF table reading tool to extract the content of the tables. However, the names of devices require a different extraction algorithm since it is not presented in formatted text. We find that most names of devices end with a few fixed words, such as device and tower. Therefore, we rely on this characteristic and adopt a special extraction algorithm. We use a combination of part-of-speech tagging and word frequency statistics to extract the names of devices. First, use a part-of-speech tagging tool to segment the original text and tag part-of-speech. We assume that the last word of a device must be one of the few words we stipulate. For the part-of-speech tagging results, we extend up to two words based on a noun word to generate candidate names of devices. Then, we use the approach of word frequency statistics to eliminate the words whose word frequency in the candidate set is less than the selected threshold. Finally, we included a manual verification procedure to increase the quality of the industrial process safety knowledge graph development, assuring the correctness and reliability of root cause analysis.

2.1.3 Entity linking

The process of entity linking (EL) involves identifying and linking entity mentions to corresponding entries in a provided database or dictionary of entities [11]. There are often many different references to the same device involved in an industrial process. The ”first regenerator” involved in the catalytic cracking process is also sometimes referred to as the ”primary regenerator”. The purpose of entity linking is to connect these two distinct references to the same entity.

Entity linking techniques can be classified into four categories: standard entity linking, cross-domain entity linking, linking to any database, and zero-shot entity linking. Given the characteristic of our data and the uniqueness of domain knowledge, standard entity linking approaches are most appropriate. The alias table is a prominent approach in standard entity linking, offering intuitive advantages. Therefore, we choose to build an alias table using word frequency statistics and text similarity to achieve entity linking.

First, we train a Word2Vec model using the original text as the corpus to obtain the vector representation of words. The Word2Vec model comprises two types: the Continuous Bag of Words (CBOW) model and the Skip-gram model [12, 13]. We choose to use the Skip-gram model to obtain word vector representations as it often exhibits superior performance. Next, we use a candidate set generation strategy similar to the data extraction part, but no longer restricted to nouns as the base, and no longer restricted to the last word. This is because the result of part-of-speech tagging on the last word of a device is not sure to be a noun, and the mentions of some devices may not end with the few words we limited. Similarly, we use the method of word frequency statistics to filter the candidate set. Finally, the Word2Vec model is used to obtain a vector representation of each mention. A vector representation of a mention is the average of the vector representations of the words it contains. We compute the cosine similarity of the vector representations between the mentions in the candidate set and the device data. Entity and mention pairs with a similarity greater than a selected threshold can be considered distinct mentions of the same entity. We then manually check and correct the alias table

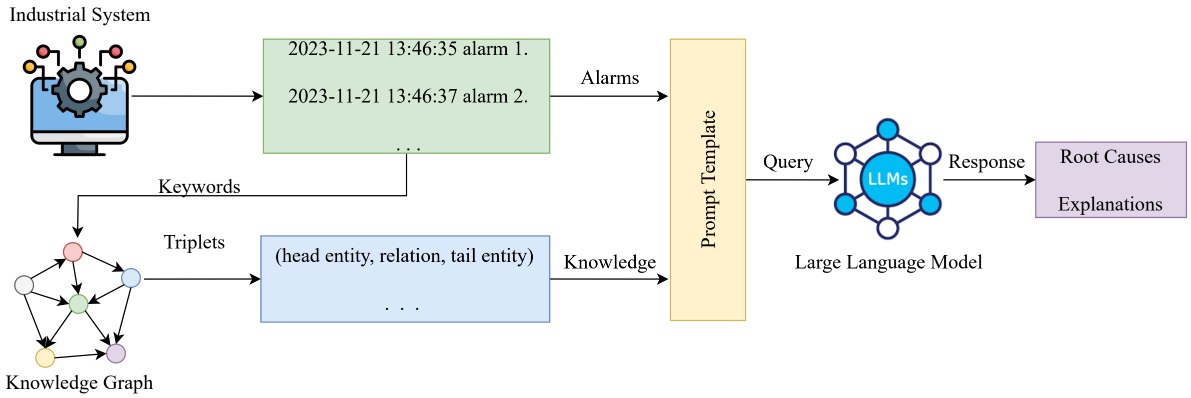


Fig. 4: The process of large language model inference

built using this approach. Furthermore, some mentions of the devices contain English codenames, which cannot be handled using this approach. So we manually add some of these mentions to the alias table.

2.2 Prior knowledge retrieval

In order to retrieve information about alarms from the industrial process safety knowledge graph as prior knowledge for the large language model, we identify named entities in the alarms. Recently, large language models have demonstrated powerful capabilities in various traditional natural language processing tasks. We are experimentally applying them to the task of named entity recognition in the industrial domain.

Using a large language model for named entity recognition involves three steps: constructing prompts, parsing the results, and validating the results. Firstly, it is necessary to construct appropriate prompts. Through the prompts, the large language model needs to be informed of the entity type to be extracted and the text to be extracted. Instruct the large language model to search for specified types of entities from the given text to be extracted, and return the results in a specific format. Furthermore, large language models may not always return results in the specified format, thus requiring the need for result parsing. By configuring the result parsing algorithm, it is possible to tolerate slight deviations from the specified format in the returned results. Finally, the obtained results need to be verified. To verify whether the extracted entities exist in the text to be extracted, if they do not exist, discard them.

Based on these extracted entities, retrieve the nodes in the industrial process safety knowledge graph whose node names contain these entities, and return factual triplets where they are used as the head entity. These factual triplets contain all the information about alarms in the industrial process safety knowledge graph, which can serve as prior knowledge for large language models to provide domain expertise. The large language model only needs to reason based on this content to deduce the root cause.

2.3 Large language model inference

With the help of the large language models' reasoning ability, we expect to identify all knowledge related to alarms in the industrial process safety knowledge graph, which can be provided as prior knowledge for the large language model to analyze the root causes of anomalies. Figure 4 displays the process of constructing prompts when using a large language model for root cause reasoning.

Firstly, we retrieve the relevant information about the current alarms from the industrial process safety knowledge graph using the method described in Section 2.2, which serves as prior knowledge for the large language model. These prior knowledge can provide factual descriptions, and then leveraging the reasoning ability of the large language model to analyze the root causes. We guide the large language model to infer the root cause of anomalies by using prompts. Provide the alarm and alarm-related fact triplets to the large language model to guide it in inferring the root cause based on this prior knowledge. Then, have it return the inference results in the specified format and parse the results to obtain the ultimate root causes. However, providing only the final analysis result lacks interpretability, so we have the large language model explain the result based on prior knowledge. By reading the explanation, one can determine whether the analysis result is correct and also gain an understanding of the reasoning chain of large language models. Finally, based on the interpretation of the result, we can decide whether to regenerate the analysis result until we obtain a reasonable outcome. Based on this approach, it integrates the powerful reasoning ability of large language models and demonstrates the excellent interpretability of knowledge graphs.

3 Experiments

3.1 Data extraction

Due to the variety of industrial text formats, we use different data extraction algorithms to extract four types of information: name of a device, normal operation status of a device, historical anomaly, and historical accident. For devices, we limit the last word to one of apparatus, tower, valve, furnace, pipe, machine, tank, pump, system, or device. We utilize word frequency statistics to remove from the candidate set any data with a word frequency of less than 10, leaving 232 device names. Finally, we manually go through the remaining device names and keep 176 of them. We utilize a PDF reading tool called pdfplumber to extract the normal operation status of the device, resulting in a total of 68 items of device status information. We extract historical

Table 1: An example of historical anomalies

Name	Content
Anomaly	The pressure of the lubricating oil main pipe of the main fan unit is low. Low pressure alarm is triggered.
Phenomenon	The bearing temperature is rising. The temperature of the oil being returned has increased. The lubricating oil pipeline is leaking.
Cause	The pressure regulating valve has malfunctioned. The lubricating oil pipe is blocked. Find and fix the leaks to eliminate them, and shut down when the volume is large.
Solution	Please contact for repair and manually adjust the oil pressure. Shut down if necessary to investigate the cause.

anomalies using pdfplumber, managing to gather 130 items in total. For historical accidents, we employ regular text matching techniques, resulting in the extraction of 45 items. Table 1 shows a typical example of historical anomalies.

3.2 Entity linking

After segmenting the original text, we utilize 10,724 sentences for training the Word2Vec model. In these sentences, we employ regular matching to eliminate non-Chinese words. A 50-dimensional vector representation is used for each word, with a window size of seven and a minimum word frequency of seven. The Word2Vec model is then trained for ten epochs using this configuration.

We establish a similarity threshold at 0.8. If the similarity between the vector representations of two strings surpasses this threshold, we can potentially view them as distinct mentions that refer to the same entity. For instance, if the similarity between the ”first regenerator” and ”primary regenerator” surpasses a specified threshold, we consider them as the same entity. We first sort the string pairs with similarities above the threshold, ordering them from the highest level of similarity to the lowest. Afterward, we conduct a manual review of the sorted results. If two strings do not refer to the same entity, we remove them from the list. We employ a PDF retrieval tool to look up the device name, swiftly obtain its potential code names, and incorporate them into the alias table. Ultimately, we maintain 100 aliases for 176 devices. Table 2 shows a representative part of the alias table.

Table 2: An example of alias table

Name	Alias
CRC cooler steam drum	CRC steam drum, D-116
Waste heat boiler	CO waste heat boiler
Sewage pretreatment station	Unit 406

3.3 Named entity recognition

In order to comprehensively gather all the information related to current alarms in the industrial process safety knowledge graph, we utilize a large language model to recognize named entities within the alarms. In industrial process alarms, there are typically four types of entities, which include devices, materials, physical quantities, and states. Identifying these four types of entities aids us in retrieving prior knowledge, allowing the large language model to infer the root causes. We evaluated the performance of ChatGPT and GPT4 on a set of 680 manually annotated samples and assessed them using precision, recall, and F1 scores as evaluation metrics. Assuming the number of correctly identified named entities is denoted as c, the number of predicted named entities is denoted as p, and the actual number of named entities is denoted as g, the calculation of these three evaluation metrics is as follows:

$$Precision = \frac{c}{p} \tag{1}$$

$$Recall = \frac{c}{g} \tag{2}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

The precision reflects the proportion of true named entities among the predicted named entities, and a higher precision indicates a higher accuracy of the model in identifying the named entities. The recall reflects the proportion of correctly predicted named entities out of all the true named entities, and a higher value indicates that the model is able to identify as many true named entities as possible. On the other hand, F1 is a metric that takes into account both precision and recall, reflecting the overall performance of the model.

Table 3: The performances of large language model on named entity recognition

Model	Precision(%)	Recall(%)	F1(%)
ChatGPT	53.5	40.4	46.0
GPT4	**80.2**	**72.6**	**76.2**

Table 3 presents the performance of two large language models on the named entity recognition task. It can be observed that ChatGPT performs poorly in named entity recognition on this dataset, achieving only a 46.0% F1 score. In comparison, GPT4 achieved an F1 score of 76.2%, which is 30.2% higher than ChatGPT, and attained a precision of 80.2%. This indicates that GPT-4 possesses a more powerful reasoning ability compared to ChatGPT, thus achieving superior performance.

3.4 Root cause analysis

We have constructed a dataset consisting of 436 samples to assess the proposed root cause analysis approach. Separate tests were conducted to evaluate the performance of ChatGPT and GPT4, in conjunction with the industrial process safety knowledge graph, in the root cause analysis task. The evaluation metrics included precision, recall, and F1 score. We apply the proposed root cause analysis approach to the constructed dataset and compare the inference results with the ground truth to obtain the three evaluation metrics. These three evaluation metrics are calculated similarly to the method described in Section 3.3. With the same equations, c represents the number of correctly identified root causes, p represents the number of predicted root causes, and g represents the number of true root causes.

Assuming there are two alarms, namely ”The riser reaction temperature drops rapidly” and ”The amount of settler storage has dropped significantly”. Firstly, extract the named entities from these two alarms, including the riser, reaction temperature, settler, and storage. Then, we retrieve relevant factual triples from the industrial process safety knowledge graph based on these named entities. Take these factual triples about the two alarms as prior knowledge for the large language model to infer the root cause. In conclusion, it can be inferred that the root cause of these two anomalies is the ”Regeneration slide valve partially closed or fully closed”.

Table 4: The performances of large language model on root cause analysis

Model	Precision	Recall	F1
ChatGPT	41.1	69.0	51.5
GPT4	**79.2**	**95.0**	**86.4**

Table 4 shows the performance of ChatGPT and GPT4 on the root cause analysis task. The performance of ChatGPT is poor, with only 51.5% F1 score, especially with a precision of 41.1%. In comparison, GPT4 performs much better, with an F1 score of 86.4%, a 34.9% improvement over ChatGPT. In addition, GPT4 achieved a precision of 79.2%, a 38.1% improvement over ChatGPT, which is nearly twice as much. Most impressively, GPT4 achieved a recall of 95%. These results demonstrate the powerful reasoning ability of GPT4, which has a significant increase in reasoning ability compared to ChatGPT. Therefore, large language models have the potential to be applied to the root cause analysis of anomalies in industrial processes and have shown promising performance. Combining the good explainability of knowledge graphs, the root cause analysis approach we propose can provide a clear chain of reasoning and accurate results.

4 Conclusion

In order to address the issue of generating a large number of alarms in the event of anomalies occurring in the industrial processes of the oil sector, we propose a root cause analysis approach that combines the knowledge graph with a large language model. This approach combines the powerful knowledge representation and interpretability of knowledge graphs with the strong reasoning ability of large language models. Integrating the advantages of both can provide significant assistance in analyzing the root causes of industrial process anomalies.

The proposed approach of root cause analysis based on a large language model and knowledge graph can effectively extract large amounts of information and decision-making experience from historical anomalies. Due to the excellent interpretability of knowledge graphs, the results of root cause analysis can be easily understood by safety managers, enabling timely measures to be taken to ensure industrial process safety. This approach holds promise for enhancing the prevention of unknown accidents in the future, thereby improving the safety of industrial processes.

References

[1] John Lee, Ian Cameron, and Maureen Hassall. Improving process safety: What roles for digitalization and industry 4.0? *Process safety and environmental protection*, 132:325–339, 2019.

[2] Amin Asadzadeh, Mehrdad Arashpour, Heng Li, Tuan Ngo, Alireza Bab-Hadiashar, and Ali Rashidi. Sensor-based safety management. *Automation in Construction*, 113:103128, 2020.

[3] Eduardo e Oliveira, Vera L Miguéis, and José L Borges. Automatic root cause analysis in manufacturing: an overview & conceptualization. *Journal of Intelligent Manufacturing*, 34(5):2061–2078, 2023.

[4] Kerelous Waghen and Mohamed-Salah Ouali. Interpretable logic tree analysis: A data-driven fault tree methodology for causality analysis. *Expert Systems with Applications*, 136:376–391, 2019.

[5] Qiuping Ma, Hongyan Li, and Anders Thorstenson. A big data-driven root cause analysis system: Application of machine learning in quality problem solving. *Computers & Industrial Engineering*, 160:107580, 2021.

[6] Ricardo Manuel Arias Velásquez and Jennifer Vanessa Mejía Lara. Root cause analysis improved with machine learning for failure analysis in power transformers. *Engineering Failure Analysis*, 115:104684, 2020.

[7] Chang Liu and Shiwu Yang. Using text mining to establish knowledge graph from accident/incident reports in risk assessment. *Expert Systems with Applications*, 207:117991, 2022.

[8] Jonathan H Choi, Kristin E Hickman, Amy Monahan, and Daniel Schwarcz. Chatgpt goes to law school. *Available at SSRN*, 2023.

[9] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001.

[10] Kiran Adnan and Rehan Akbar. An analytical study of information extraction from unstructured and multidimensional big data. *Journal of Big Data*, 6(1):1–38, 2019.

[11] Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. Zero-shot entity linking by reading entity descriptions. *arXiv preprint arXiv:1906.07348*, 2019.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.