

Article

Not peer-reviewed version

---

# Research on Construction and Application of Aircraft Fault Knowledge Graphs

---

[Xilang Tang](#) , Guo Chi , Lijie Cui , [Andrew W.H. Ip](#) , [Kai Leung Yung](#) , [Xiaoyue Xie](#) \*

Posted Date: 28 April 2023

doi: 10.20944/preprints202304.1140.v1

Keywords: Aircraft fault diagnosis; knowledge graph; deep learning; fault knowledge extraction; question-answering system



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Article

# Research on Construction and Application of Aircraft Fault Knowledge Graphs

Xilang Tang <sup>1</sup>, Guo Chi <sup>2</sup>, Lijie Cui <sup>1</sup>, Andrew W.H. Ip <sup>3</sup>, Kai Leung Yung <sup>4</sup> and Xiaoyue Xie <sup>1,\*</sup>

<sup>1</sup> Air Force Engineering University, Xi'an, China

<sup>2</sup> Engineering University of PAP, Xi'an, China

<sup>3</sup> University of Saskatchewan, Saskatoon, Canada

<sup>4</sup> The Hong Kong Polytechnic University, Hong Kong, China

\* Correspondence: xxyueamss@163.com

**Abstract:** When an aircraft malfunctions, quickly and accurately identifying the faulty unit is essential for ensuring normal operation. Unfortunately, maintenance engineers often struggle to acquire the necessary fault-related knowledge due to poor management and utilization of aircraft fault documents. To address this issue, we introduce knowledge graph technology into the field of aircraft fault diagnosis, exploring its construction and application for effective knowledge management. Our work starts by analyzing the critical knowledge elements required for aircraft fault diagnosis and designing a schema layer for fault knowledge graphs. We then we then combine deep learning and heuristic rules to extract fault knowledge from both structured and unstructured data, enabling the construction of aircraft fault knowledge graphs. Finally, we develop a fault question-answering system based on fault knowledge graphs that can accurately give solutions to questions posed by maintenance engineers. Our practice demonstrates that knowledge graphs provide an effective means of managing aircraft fault knowledge, assisting engineers in locating fault reasons accurately and quickly.

**Keywords:** Aircraft fault diagnosis; knowledge graph; deep learning; fault knowledge extraction; question-answering system

## 1. Introduction

Aircraft fault diagnosis involves identifying the cause of system function failures in aircraft under specific working conditions. Quickly and accurately locating the faulty unit is essential for repairing aircraft and ensuring their proper functioning. With the development of aircraft health management technology, some faults that have been considered in the design phase can be diagnosed by the aircraft itself [1]. However, a large number of faults still rely on the experience of engineers to be diagnosed when they cannot be predicted in the design phase. Given the complexity of aircraft, fault diagnosis is a typical knowledge-intensive activity[2,3]. Maintenance engineers need to spend a lot of time reading the related text materials and mastering knowledge of related to the aircraft's structure, function, signals, and possible fault causes, etc. With this knowledge, engineers conduct observations, measurements, and other test methods to gradually locate the fault unit. Unfortunately, this diagnostic process is often inefficient, slow, and can lead to prolonged aircraft downtime, resulting in severe economic losses.

To improve the efficiency of aircraft fault diagnosis, researchers have proposed developing interactive electronic manuals and other fault auxiliary diagnosis systems [4]. These advanced diagnostic tools aim to guide maintenance engineers in taking reasonable test steps to quickly and accurately locate the fault unit. The key point of developing such a system is to build a complete fault knowledge base and design an efficient knowledge reasoning method to automatically generate diagnosis strategies. While common knowledge-base models such as fault tree model [5,6], Petri net model [7,8], rule model [9,10], and causality graph [11] have been used, they suffer from poor generalization ability, lack flexibility, and limited dynamic updates, making it difficult to share

diagnostic experiences and knowledge among maintenance engineers in a timely manner. Therefore, like other industries, aircraft fault diagnosis requires efficient information integration, processing, and service-oriented information visualization [12,13], thus enabling machine learning algorithms to automatically mine and present necessary knowledge, avoiding significant manpower to create a knowledge base.

Knowledge graph technology aims to mine entity relationships from massive, fragmented information and present them as formalized semantic networks, providing better knowledge mining, representation, and management methods for artificial intelligence [14–16]. At present, knowledge graphs has been widely used in medical[17], education[18], finance[19], e-commerce[20], safety [21,22], and others, advancing AI from perceptual intelligence to cognitive intelligence [15,23]. With knowledge graph technology, fault knowledge can be extracted from vast, multi-source, heterogeneous, and unstructured fault data [24], such as aircraft maintenance and support textbooks, aircraft test and diagnosis records, aircraft quality control data, and fault analysis and research reports. By integrating the extracted data into a structured interconnected fault knowledge graph and applying reasoning technology, intelligent fault diagnosis strategies can be generated to guide maintenance engineers to locate the fault unit quickly and accurately. As a result, an increasing number of scholars have started exploring the application of knowledge graph technology in the field of fault diagnosis. For instance, Zhang utilized fault maintenance logs as the research object and proposed a method to construct a fault diagnosis event logic knowledge graph for transmission systems [25]. Meanwhile, Liu described the correlation between fault components of transformers for fault diagnosis by constructing knowledge graphs from maintenance records [26].

While previous studies have mainly focused on constructing fault knowledge graph models, exploration of their application remains relatively weak. Moreover, construction and application research on fault knowledge graph for aircraft is rarely little. The current study aims to explores both the construction and application of aircraft fault knowledge graphs, with three significant contributions:

(a) The model of aircraft fault knowledge is constructed by consulting experts in the field of fault diagnosis, comprehensively sorting out the elements of aircraft fault knowledge, and identifying the entity types, attributes, and relationship patterns involved in fault diagnosis based on the practical needs of smart fault diagnosis of aircraft.

(b) A fault knowledge extraction method based on deep learning assisted by heuristic rules is used comprehensively. Unstructured fault documents are mostly extracted by deep learning methods, while semi-structured data, tables, or normative documents with certain structural characteristics are extracted by heuristic rules.

(c) We built a question and answer system based on aircraft fault knowledge graphs, comprising question preprocessing, question analysis, graph retrieval, and answer generation modules.

## 2. Construction and application framework of aircraft fault knowledge graphs

The construction and application framework of fault knowledge graphs is shown in the Figure 1. It contains five layers, namely fault data layer, knowledge extraction layer, knowledge graph layer, application technology layer and human-computer interaction layer.

The fault data layer consists of a vast and varied amount of unstructured documentation related to faults accumulated over an aircraft's lifecycle, such as FMECA (failure mode, effect, and criticality analysis) documents, maintenance support textbooks, test diagnosis records, quality control data, support records, fault analysis, and research reports. These documents are valuable knowledge sources for aircraft fault diagnosis, but their multi-source heterogeneity poses challenges in mining relevant diagnosis information effectively. They originate from multiple sources, with no uniform format or standard, so that the traditional structured data analysis method cannot deal with such fault data effectively. Therefore, artificial intelligence methods must be developed to mine these documents and learn the hidden fault knowledge, building an intelligent fault diagnosis knowledge base.

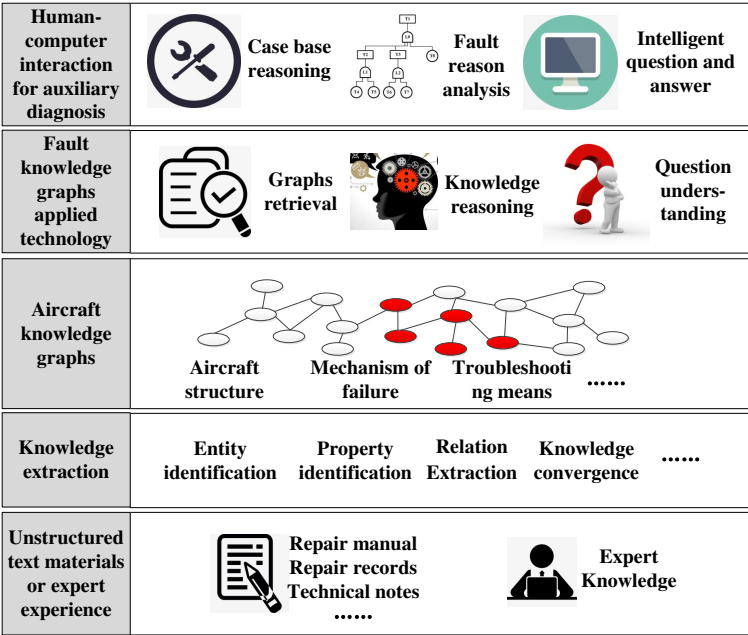


Figure 1. Construction and application framework of aircraft fault knowledge graphs

The knowledge extraction layer involves identifying the entities, attributes, and relationships related to faults from the collected documents to build structured and interconnected fault knowledge graphs. In the past, linguistic heuristic rules and statistics were used for fault knowledge extraction[27, 28], requiring sufficient prior knowledge for designing reasonable extraction rules or artificial features. At present, with the continuous improvement of deep learning technology, knowledge extraction based on this technology are increasingly being applied due to their advantages of automatic feature extraction from unmarked data, without relying on prior knowledge[27–31].

The knowledge graph layer refers to the structured "large-scale" fault knowledge graphs extracted from massive documents. It is called "large-scale" because the aircraft system is very complex, with hundreds of thousands of components. When the failures, signal parameters and composition relationship of each component are described, the number of nodes and edges in the graph will be in the millions. Therefore, in order to facilitate the query and update of the large-scale graph, it is necessary to use database management system (DBMS) such as Graph DBMS, RDF Stores, etc[32,33].

The application technology layer matches similar fault cases and infers possible fault causes using the fault knowledge graph, generating diagnosis strategies or answering maintenance engineers' questions through semantic retrieval, knowledge reasoning, problem understanding, and other technologies.

The human-computer interaction layer refers to the visual interface directly facing the maintenance engineer, including the visual display of fault knowledge graph, the visual interface of fault reasoning analysis, and the interactive interface of intelligent question and answer.

3. Construction of aircraft fault knowledge graphs schema

In general, a domain knowledge graph is constructed in a top-down manner. In this construction process, it is the first step to design the graph schema according to the professional application requirements. That is, we must first define the types of entities, attributes, relationships that may exist in the graph. With schema, instances of entities, attributes and relationships can then be extracted automatically from unstructured documents to build the domain knowledge graph[34,35]. The construction of a fault knowledge graph falls under this method, so it is crucial to create a schema for fault diagnosis. Typically, the schema layer of a knowledge graph is formed using ontology modeling. Ontology is a method used to clearly define knowledge units within domain knowledge

[36]. For instance, "component", "isPartOf", and "subsystem" represent an ontology-based schema, while "receiver", "isPartOf", and "radar" are examples of this schema, i.e., a knowledge graph. Recently, researchers have focused on ontology-based fault knowledge modeling to achieve fault knowledge extraction and sharing[37–40]. As a result, ontology is well-suited for describing a fault knowledge model, and knowledge graphs provide an ideal platform for acquiring and expressing fault knowledge as instances of an ontology.

### 3.1. Construction principle

The schema layer, or ontology layer, plays a crucial role in determining the form of the fault knowledge graph and its practical application. Therefore, we adhered to the following principles when designing the schema for the aircraft fault knowledge graph:

- **Oriented towards fault diagnosis requirements:** The schema layer has significant impact on the shape and quality of the fault knowledge graph as well as its diagnostic efficiency. Hence, during schema construction, it is essential to fully understand the knowledge elements required for fault diagnosis and seek the opinions of field experts and frontline aircraft maintenance personnel. Relying on expert assessment, we aimed to create a fault-diagnosis-oriented schema that would be recognized by industry professionals.
- **Continuous refinement:** Despite being top-down, the schema needs iterative refinement based on feedback from the application and technical challenges. For instance, at the outset, we defined entity types such as subsystems, components, parts, and units of varying levels. However, while extracting knowledge, it was impossible to determine a specific component's level from context alone. Therefore, we optimized the design of schema layer and uniformly defined these entity types as "unit."
- **Simple and effective:** As the schema layer complexity increases, more entities and relationships need extraction from text, leading to extensive labor costs for corpus labeling. Therefore, the schema should be simplified as much as possible under the condition of meeting the diagnostic business needs, and avoid unacceptable manual labeling costs.
- **Consider both structured and unstructured documents:** Aircraft fault data comprises both structured and unstructured data formats. Structured data typically includes Excel and standardized XML formats, rich in information that can be directly converted into triples. Ideally, the schema should encompass all structured data information possible. Unstructured data mainly comprises Word and PDF documents, requiring deep learning and other techniques to extract knowledge. To simplify entity classification, the schema assigns structured data information as properties to reduce the number of entity types required.

### 3.2. The schema of fault knowledge graph

Based on the above principles, several iterations of optimization were performed to define the schema presented in Figure 2, and the definitions of entity types and their relationships are shown in Tables 1 and 2. The schema comprises "unit", "signal", "fault", "fault case", and "test method" entity types. Notably, the schema does not distinguish fault phenomena, modes, and causes as separate entities but rather defines them collectively as "fault." This decision is based on the fact that these three types of entities are expressed similarly in unstructured text; for example, the fault phenomenon "right engine speed is too high" may correspond to the fault cause "right engine FRV cannot be opened". Machine learning algorithms cannot reliably differentiate between these entities based on contextual features alone. Further, distinguishing between these entities adds little value to fault diagnosis since establishing a "cause" relationship between faults A and B suffices to infer that A caused B. Moreover, faults can form propagation chains where fault A triggers fault B, and fault B leads to fault C. In such cases, it is necessary to identify the component in the propagation chain where the fault occurred. Defining "fault cause" as a distinct entity would complicate the description of fault propagation chains. In summary, "fault" and its "cause" relationship are defined in the schema. Figure 3 shows an example

of local knowledge graph under the defined schema, and the elements in the graph shall be extracted from the fault documents with some methods.

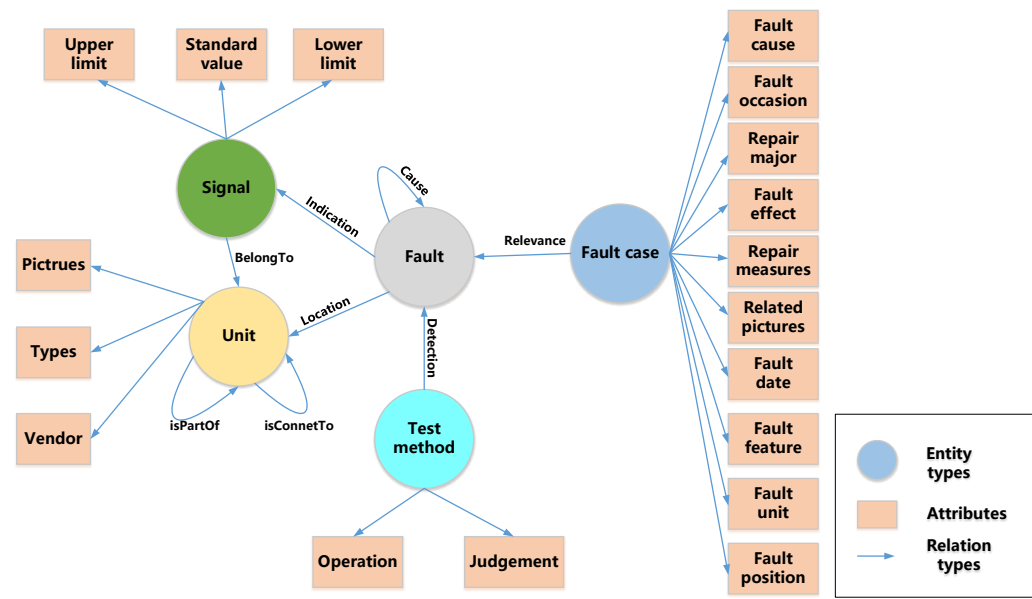


Figure 2. The schema of fault knowledge graph

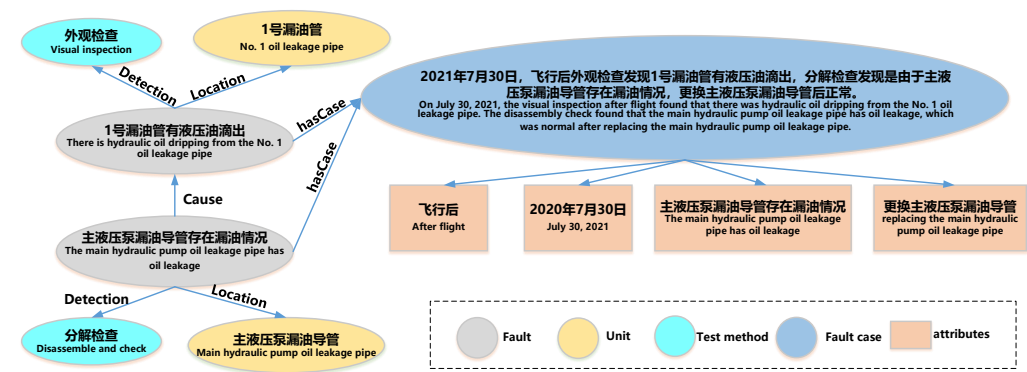


Figure 3. An example of local knowledge graph under defined schema

Table 1. Definition of entity type

Entity type	Connotation
Unit	An entity element at a specific level of the system (such as subsystem, component, part, etc.). A unit can be composed of several subunits, or it can be a subunit of a larger unit. A unit's attributes include types, vender, picture.
Signal	The parameters that can be measured. Many faults may be characterized as abnormal signals. It includes attributes such as standard value, upper limit and lower limit.
Fault	Description of all abnormal events, such as abnormal function, parameter deviation, mechanical damage, etc., which includes fault phenomenon, fault mode and fault cause.
Fault case	Specific actual cases of fault. A fault may occur multiple times at different times and places, and each occurrence is a fault case. It includes fault time, fault location, fault effect and other attributes.
Test method	The method to detect a fault, which is the key knowledge to guide the maintenance engineer to quickly locate the fault unit. It contains two attributes: operation and judgment.

**Table 2.** Definition of relation type

Relation type	Connotation	Head entity	Tail entity
Cause	Causality relation between fault	fault	fault
Detection	It indicates that a fault can be detected by a test method	fault	test method
isPartOf	A unit is part of another unit	unit	unit
isConnectTo	There is a physical contact or signal input/output relationship between a unit and another unit	unit	unit
BelongTo	A signal belongs to a unit	signal	unit
Location	A fault happens at a unit	fault	unit
Indication	a fault can be indicated by a signal	fault	signal
Relevance	a fault case has relevance to a fault	fault case	fault

#### 4. Fault knowledge extraction

Fault knowledge extraction involves identifying the entity and relationship types defined by the schema presented above from both structured and unstructured fault documents. This paper utilizes a comprehensive fault extraction method, mainly based on deep learning supplemented by heuristic rules. For unstructured fault documents, the deep learning method is primarily used for knowledge extraction, while heuristic rule methods are applied to semi-structured data, tables, or standard documents with specific structural characteristics. The following section focuses on introducing the deep learning method for extracting information from unstructured documents.

##### 4.1. Entity recognition

As previously discussed, deep learning has become a vital approach for knowledge extraction due to its ability to automatically discover features for classification or detection tasks. Among various deep learning methods, pre-trained language model embeddings such as BERT[41] are emerging as a new paradigm for entity recognition[42]. Common basic models for entity recognition based on BERT include BERT+Softmax, BERT+CRF, BERT+Span, BERT+MRC, and other similar methods[42–44]. Rather than treating entity recognition solely as a sequence labeling problem, BERT+MRC formulates it as a machine reading comprehension (MRC) task[43]. This method can introduce label information by transforming the span extraction method into a question-and-answer format. However, Pan found that BERT-MRC does not fully utilize the knowledge information of labels and proposed an enhanced language representation with label knowledge for span extraction, referred to as LEAR (Label-knowledge Enhanced Representation)[44].

The fundamental idea behind span extraction is to determine the start and end positions of entities in unstructured text. However, previous models did not consider the prompt for the end position given the starting position. Consequently, we further propose to integrate the identified starting position information into the language representation based on LEAR, which simplified as SP-LEAR (Start Position and Label-knowledge Enhanced Representation). The framework of SP-LEAR is shown in Figure 4.

The two key steps of SP-LEAR are as follows:

**(1) Fusion of type annotations into sentences.** Assume that  $n$  is the length of sentence, and  $m$  is the length of label annotation. After process of BERT network model,  $h_i \in \mathcal{R}^d$  is the embedding of the  $i$ th token of the sentence for any  $1 \leq i \leq n$ , and  $h_j^c$  is the embedding of the  $j$ th token of the annotation of type  $c$  for any  $1 \leq j \leq m$ . To fuse the information of label annotations into the sentences, the attention mechanism is used. The attention score is obtained by,

$$a(h_i, h_j^c) = \frac{\exp(\mathbf{W}^q h_i) \cdot (\mathbf{W}^k h_j^c)}{\sum_j \exp(\mathbf{W}^q h_i) \cdot (\mathbf{W}^k h_j^c)} \quad (1)$$

where,  $\mathbf{W}^q, \mathbf{W}^k \in \mathcal{R}^{d \times d}$  are learnable parameters of the fully connected layers. Then the fusion features are obtained by,

$$O_i^c = \tanh(\mathbf{W}^v(\mathbf{W}^q h_i + a(h_i, h_j^c) \mathbf{W}^k h_j^c) + \mathbf{b}) \quad (2)$$

where  $\tanh(\cdot)$  is the hyperbolic tangent function, and  $\mathbf{W}^v \in \mathcal{R}^{d \times d}$  and  $\mathbf{b} \in \mathcal{R}^d$  are learnable parameters of the fully connected layers.

**(2) Fusion of start position into features.** Using the fused features as input, the start position classifier can output the probability of the  $i$ th token as the starting position of a span of the category  $c$ , represented as  $S_i^c$ . The new feature  $U_i^c$  are obtained by a concatenation of the aforementioned fusion features and start position probability as follows,

$$U_i^c = \tanh(\mathbf{M} \cdot \text{concat}(O_i^c, S_i^c)) \quad (3)$$

where  $\mathbf{M} \in \mathcal{R}^{d \times d}$  is learnable parameters of the fully connected layers. Finally, the new features are as input of end position classifier, and the probability of the  $i$ th token as the ending position of a span of the category  $c$  is obtained. The loss function of the model is defined to be  $\mathcal{L} = \mathcal{L}_s + \mathcal{L}_e$ , where  $\mathcal{L}_s, \mathcal{L}_e$  are respectively start loss function and end loss function and the two values are calculated by cross entropy. The nearest matching principle [45] which matches a start position of category  $c$  with the nearest next end position of  $c$  is used to decode the span.

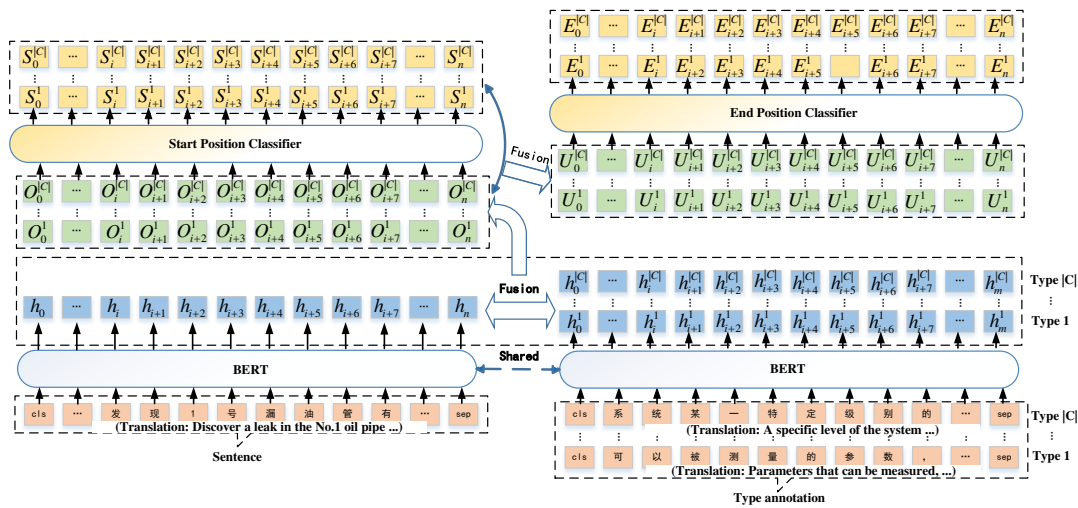


Figure 4. The framework of SP-LEAR

We labeled 4305 entities from 1000 paragraphs, with 700 paragraphs allocated for the training set and 300 paragraphs for the testing set. Our experiments involved different models using this dataset, and a comparison of the results is shown in Table 3. The best performing model results are highlighted in bold. It is apparent that the direct classification of BERT embeddings using softmax performs poorly since it does not consider the dependencies between characters. The "BERT+CRF" entity recognition model, which uses a CRF decoder to determine each token's "BIO" label, outperforms softmax but falls short of "BERT+span". LEAR is an improvement over "BERT+span" as it includes label information when generating features, resulting in higher recall. Compared to LEAR, our method incorporates the starting position into features when classifying the ending position. From the results, we observe that although SP-LEAR has lower recall than LEAR, accuracy improves significantly, with a higher F1 value.

**Table 3.** Results comparison of different model.

Model	P	R	F1
BERT+Softmax	0.283	0.430	0.341
BERT+CRF	0.757	0.797	0.778
BERT+SPAN	0.796	0.799	0.797
LEAR	0.797	<b>0.833</b>	0.814
SP-LEAR	<b>0.813</b>	0.827	<b>0.820</b>

Table 4 displays the performance of SP-LEAR for different entity recognition. Based on these results, we can draw an important conclusion that the "Unit" and "Signal" extraction performance is much better compared to "Fault" and "Test method". This is primarily due to "Unit" and "Signal" being nouns with shorter words and clear boundaries while "Fault" and "Test method" are short sentences containing predicates with relatively vague boundaries. Consequently, further research work is necessary to enhance the effectiveness of extracting entities like "Fault" and "Test method".

**Table 4.** Results comparison of different model.

Entity category	P	R	F1
Unit	0.908	0.908	0.908
Signal	0.833	0.894	0.863
Fault	0.820	0.790	0.804
Test method	0.780	0.832	0.805

#### 4.2. Relation extraction

The relationship extraction method combines rule-based and model-based approaches. The rule-based method relies on our understanding of fault data and knowledge logic, involving the manual creation of extraction rules, regular expressions, and templates. For instance, in equipment technical documentation, the composition relationship between fault units can be extracted directly based on the directory relationship of each composition unit. Similarly, if a sentence contains words such as "cause" or "lead to" between two fault entities, it suggests a causal relationship between them. Such rules enable accurate extraction of a large number of relationships in a simple and effective manner.

The deep learning method is also employed to extract relationships. This method converts relationship extraction into a binary classification problem, predicting whether a predefined relationship exists between two entities in a paragraph. The classifier features are extracted according to the following steps illustrated in Figure 5. Firstly, sentences containing entity pairs are located and their features are extracted using the "BERT+Convolution+Pooling" model. Paragraph-level features are then extracted through the "Convolution+Pooling" model, and concatenated with the sentence features that contain entity pairs to serve as input for the classifier model. For example, consider the following excerpt: "... the alarm system continuously reflects that the right engine speed is too high during the descent phase. The reason is that the oil level sensor control component fails, ...". Both sentences contain a fault entity. The features of these two sentences and paragraphs are concatenated, and then the classification model is used to determine whether a causal relationship exists between the two fault entities.

#### 4.3. Storage and Visual Display of Fault Knowledge Graph

Arangodb and neo4j are commonly used graph databases, but compared with neo4j, ArangoDB is an open-source distributed native multi-model database that boasts features like multi-model support, scalability, and high performance. Additionally, ArangoDB has a unified query language similar to SQL, called AQL, enabling rapid graphical queries that return nodes, edges, and their attributes. These precise characteristics make ArangoDB ideal for storing triplets of fault knowledge graphs.

Figure 6 depicts the local fault knowledge graph of an aircraft braking system, which includes critical information such as the system’s composition structure, signal parameters, related faults, and fault detection methods. This fault knowledge graph provides the essential knowledge support for fault diagnosis.

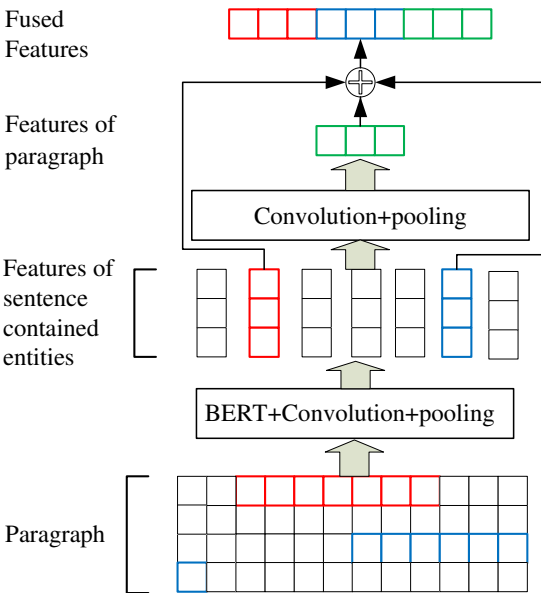


Figure 5. The framework of SP-LEAR

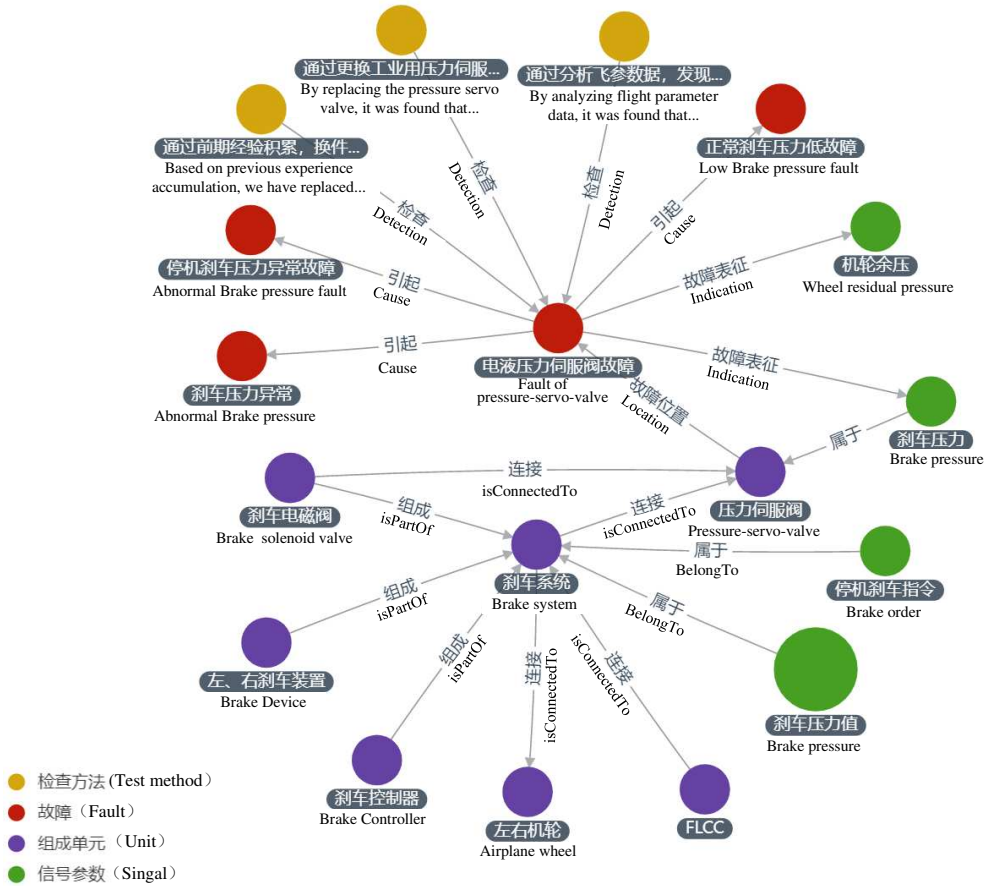


Figure 6. Visual display of fault knowledge graph

5. Question-answer System Based on Fault Knowledge Graph

To aid maintenance personnel in efficiently pinning down the cause of faults, an intelligent answer system was constructed, based on the fault knowledge graph. It comprises a question preprocessing module, a question analysis module, a graph retrieval module, and an answer generation module, as illustrated in Figure 7.

- The question preprocessing module performs tasks like spelling correction, word segmentation, and part-of-speech tagging on input questions, resulting in lower-level granular question information.
- The question analysis module, equipped with a word slot extraction unit and an intention recognition unit, extracts word slots and recognizes intentions of pre-processed questions. By using precise or fuzzy matching templates, this module analyzes and extracts relevant slot data from the problem, including entity slots, schema slots, attribute slots, condition slots, etc., which are predefined by intention template. For example: "How many components are there in the braking system?" This query yields extracted slots such as the entity slot "braking system," schema slot "unit," and condition slot "how many."
- The graph retrieval module retrieves pertinent fault information from the knowledge graph module based on the intention template garnered from the question analysis module. It translates intent template data into AQL query statements via rule heuristics and template matching, connecting to ArangoDB and retrieving entity names, relationship names, attribute names, attribute values, etc., from the database. At the same time, it also brings up relevant documents through entities, ensuring full knowledge traceability.
- The answer generation module converts and merges the database results based on answer templates, arranges the answers, sorts them, and returns an answer set and related documents.

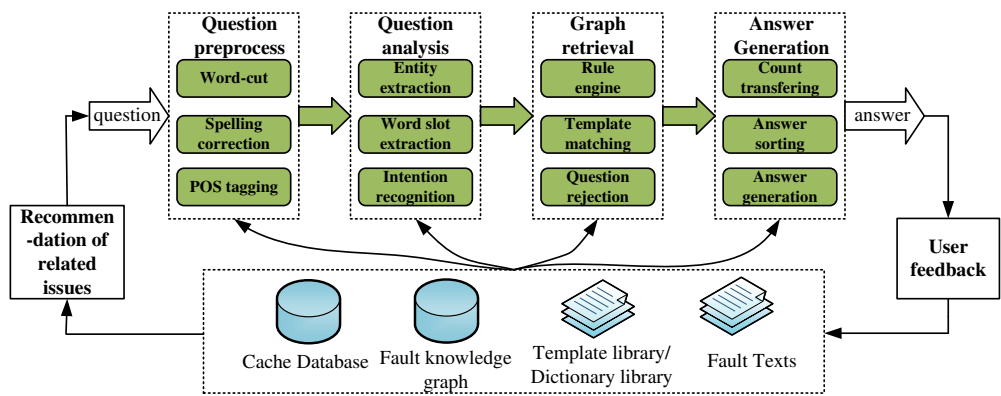


Figure 7. The framework of intelligent answer system

With the technical modules outlined above, we have developed an interactive interface for our fault knowledge question-answering system (as depicted in Figure 8). By entering the statement "The hydraulic solenoid valve has malfunctioned," the system returns accurate test methods for diagnosing and resolving this fault. This feature enables engineers to promptly access fault reasons and evaluate corrective measures by following the provided test method. Additionally, the system can extensively trace documents associated with answers, ensuring answer results' interpretability. Based on entity similarities, the system can suggest relevant questions and comprehensively address the completeness of answers.

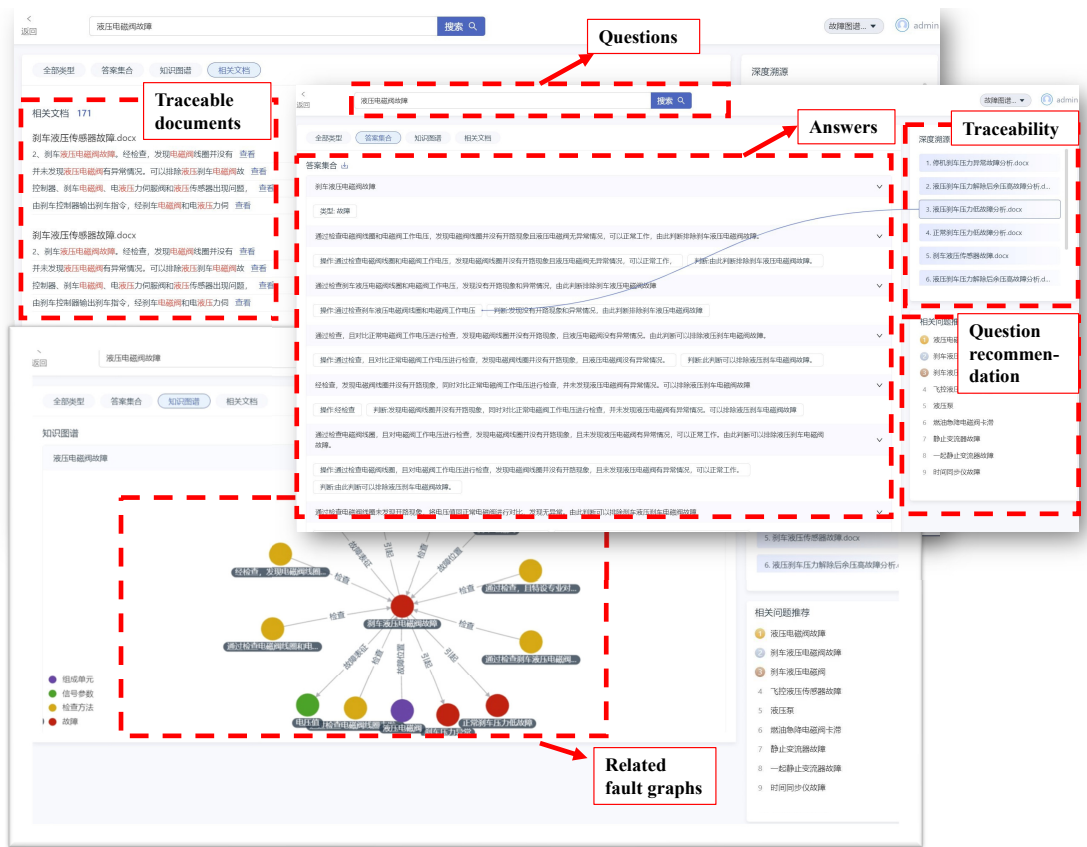


Figure 8. The application of fault knowledge graphs

6. Conclusions and discussion

This paper explores the construction and application methods of aircraft fault knowledge graphs. Firstly, based on the knowledge needs of aircraft fault diagnosis, we comprehensively sorted knowledge elements and constructed a schema layer for the aircraft fault knowledge graph. Secondly, we conducted a preliminary investigation into the method of extracting fault knowledge from structured and unstructured data. Using this approach, we constructed the fault knowledge graph for a specific type of aircraft and achieved the storage and visualization display of that fault knowledge graph. Finally, we developed a fault knowledge Q&A system based on the fault knowledge graph, which can accurately answer maintenance engineers’ questions and trace the documents where the answers are located. This system provides an effective means for maintenance engineers to quickly acquire knowledge related to faults. Practice has demonstrated that with knowledge graph technology, massive structured and unstructured fault data can be fully exploited, correlated, and integrated into a fault knowledge base, thereby facilitating the sharing and the further application of fault knowledge. It can be predicted that knowledge graphs represent an efficient means for aircraft fault knowledge management, improving diagnostic efficiency and accuracy. However, to further improve the quality of fault knowledge graphs and intelligent question answering systems, efforts should be focused in the following areas.

(a) The current exploration applies traditional knowledge graphs to the field of fault diagnosis. However, conventional knowledge graphs mainly construct static knowledge bases focusing on nouns, describing "what is" and what "relationships exist". In contrast, aircraft fault knowledge involves dynamic predicate events requiring dynamic knowledge such as "why and how" to be described. For instance, the fault phenomenon "right engine speed too high during the descent phase" is an event rather than a noun. Empirical evidence demonstrates that if it leads to some potential problems if the event is purely treated as a static entity; on one hand, these "event entities" typically comprise

short sentences with fuzzy boundaries and low recognition accuracy and recall using commonly used entity recognition methods; on the other, finer-grained knowledge embedded within the event may be missed entirely. Therefore, to overcome these difficulties, traditional knowledge graph models need to integrate events, so as to provide a more comprehensive description of fault events and their logical relationships, in turn improving recognition accuracy and recall.

(b) ChatGPT has showcased its powerful information integration and dialogue capabilities. Undoubtedly, technologies related ChatGPT can also be implemented in the field of aircraft fault diagnosis, empowering a more robust interactive diagnostic assistant. However, ChatGPT is opaque and lacks traceable knowledge, sometimes fabricating facts, which will bring practical problems for maintenance personnel. Therefore, our next step will combine knowledge graph technology with ChatGPT technology to create a more efficient and traceable interactive diagnostic system that generates reliable answers for engineers. A viable approach is to first retrieve relevant knowledge elements from the fault knowledge graph, then use them to prompt a natural language generative model, generating dependable answers.

**Funding:** This work was supported by the National Natural Science Foundation of China (No. 72201276) and China Postdoctoral Science Foundation funded Project (No.2021M693941).

## References

1. Che, C.; Wang, H.; Fu, Q.; Ni, X. Combining Multiple Deep Learning Algorithms for Prognostic and Health Management of Aircraft. *Aerospace Science and Technology* **2019**, *94*, 105423. doi:10.1016/j.ast.2019.105423.
2. Yu, X.; BeiHang, Q.L.; Hu, X. Aircraft fault diagnosis system research based on the combination of CBR and FTA. 2015 First International Conference on Reliability Systems Engineering (ICRSE). IEEE, 2015, pp. 1–6.
3. Deng, W.; Wen, B.; Zhou, J.; Wang, J.; Chen, Z. The study of aircraft fault diagnosis method based on the integration of case and rule reasoning. *Proceedings of 2014 Prognostics and System Health Management Conference, PHM 2014* **2014**, pp. 271–274. doi:10.1109/PHM.2014.6988177.
4. Niu, G.; Li, H. IETM centered intelligent maintenance system integrating fuzzy semantic inference and data fusion. *Microelectronics Reliability* **2017**. doi:10.1016/j.microrel.2017.03.015.
5. Kabir, S. An overview of Fault Tree Analysis and its application in model based dependability analysis. *Expert Systems with Applications* **2017**, *77*. doi:10.1016/j.eswa.2017.01.058.
6. Pan, K.; Liu, H.; Gou, X.; Huang, R.; Ye, D.; Wang, H.; Glowacz, A.; Kong, J. Towards a Systematic Description of Fault Tree Analysis Studies Using Informetric Mapping. *Sustainability* **2022**, *14*, 11430. doi:10.3390/su141811430.
7. Ma, L. Research on the Fault Diagnostic of the Aircraft Cross-Linking Systems. *Journal of Physics: Conference Series* **2022**, *2220*, 012013. doi:10.1088/1742-6596/2220/1/012013.
8. Xu, C.; Li, J.; Cheng, X. Comprehensive Learning Particle Swarm Optimized Fuzzy Petri Net for Motor-Bearing Fault Diagnosis. *Machines* **2022**, *10*, 1022. doi:10.3390/machines10111022.
9. Guo, Y.; Wang, J.; Chen, H.; Li, G.; Huang, R.; Yuan, Y.; Ahmad, T.; Sun, S. An expert rule-based fault diagnosis strategy for variable refrigerant flow air conditioning systems. *Applied Thermal Engineering* **2018**, *149*. doi:10.1016/j.applthermaleng.2018.12.132.
10. Wang, Z.; Li, S.; He, W.; Yang, R.; Feng, Z.; Sun, G. A New Topology-Switching Strategy for Fault Diagnosis of Multi-Agent Systems Based on Belief Rule Base. *Entropy* **2022**, *24*, 1591. doi:10.3390/e24111591.
11. Bu, X.; Nie, H.; Zhang, Z.; Zhang, Q. An Industrial Fault Diagnostic System Based on a Cubic Dynamic Uncertain Causality Graph. *Sensors* **2022**, *22*. doi:10.3390/s22114118.
12. Shen, B.; Xu, L.; Cai, H.; Yu, H.; Hu, P.; Jiang, L.; Guo, J. Enhancing Context-Aware Reactive Planning for Unexpected Situations of On-Orbit Spacecraft. *IEEE Transactions on Aerospace and Electronic Systems* **2022**, *58*, 4965–4983. doi:10.1109/TAES.2022.3172022.
13. Yan, Q.; Wang, H. Double-Layer Q-Learning-Based Joint Decision-Making of Dual Resource-Constrained Aircraft Assembly Scheduling and Flexible Preventive Maintenance. *IEEE Transactions on Aerospace and Electronic Systems* **2022**, *PP*, 1–18. doi:10.1109/TAES.2022.3211247.

14. Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amato, C.; Melo, G.d.; Gutierrez, C.; Kirrane, S.; Gayo, J.E.L.; Navigli, R.; Neumaier, S.; others. Knowledge graphs. *ACM Computing Surveys (CSUR)* **2021**, *54*, 1–37.
15. Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Philip, S.Y. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems* **2021**, *33*, 494–514.
16. Nickel, M.; Murphy, K.; Tresp, V.; Gabrilovich, E. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE* **2015**, *104*, 11–33.
17. Zhang, D.; Jia, Q.; Yang, S.; Han, X.; Xu, C.; Liu, X.; Xie, Y. Traditional Chinese Medicine Automated Diagnosis Based on Knowledge Graph Reasoning. *Computers, Materials and Continua* **2022**, *71*. doi:10.32604/cmc.2022.017295.
18. Mao, Y. Summary and Evaluation of the Application of Knowledge Graphs in Education 2007–2020. *Discrete Dynamics in Nature and Society* **2021**, *2021*, 1–10. doi:10.1155/2021/6304109.
19. Wu, H.; Chang, Y.; Li, J.; Zhu, X. Financial fraud risk analysis based on audit information knowledge graph. *Procedia Computer Science* **2022**, *199*, 780–787.
20. Yuan, L.; Huang, Z.; Zhao, W.; Stakhiyevich, P. Interpreting and predicting social commerce intention based on knowledge graph analysis. *Electronic Commerce Research* **2020**, *20*. doi:10.1007/s10660-019-09392-1.
21. Pandithawatta, S.; Ahn, S.; Rameezdeen, R.; Chow, C.W.K.; Gorjian, N.; Kim, T.W. Development of a Knowledge Graph for Automatic Job Hazard Analysis: The Schema. *Sensors* **2023**, *23*. doi:10.3390/s23083893.
22. Yin, Z.; Shi, L.; Yuan, Y.; Tan, X.; Xu, S. A Study on a Knowledge Graph Construction Method of Safety Reports for Process Industries. *Processes* **2023**, *11*. doi:10.3390/pr11010146.
23. Yahya, M.; Breslin, J.; Ali, M.I. Semantic Web and Knowledge Graphs for Industry 4.0. *Applied Sciences* **2021**, *11*, 5110. doi:10.3390/app11115110.
24. Li, M.; Ni, Z.; Tian, L.; Hu, Y.; Shen, J.; Wang, Y. Research on Hierarchical Knowledge Graphs of Data, Information, and Knowledge Based on Multiple Data Sources. *Applied Sciences* **2023**, *13*. doi:10.3390/app13084783.
25. Deng, J.; Wang, T.; Wang, Z.; Zhou, J.; Cheng, L. Research on Event Logic Knowledge Graph Construction Method of Robot Transmission System Fault Diagnosis. *IEEE Access* **2022**, *10*, 1–1. doi:10.1109/ACCESS.2022.3150409.
26. Liu, L.; Wang, b.; Ma, F.; Zheng, Q.; Yao, L.; Zhang, C.; Mohamed, M. A Concurrent Fault Diagnosis Method of Transformer Based on Graph Convolutional Network and Knowledge Graph. *Frontiers in Energy Research* **2022**, *10*. doi:10.3389/fenrg.2022.837553.
27. Zhao, H.; Pan, Y.; Yang, F. Research on Information Extraction of Technical Documents and Construction of Domain Knowledge Graph. *IEEE Access* **2020**, *8*, 168087–168098. doi:10.1109/ACCESS.2020.3024070.
28. Jaradeh, M.Y.; Singh, K.; Stocker, M.; Both, A.; Auer, S. Information extraction pipelines for knowledge graphs. *Knowledge and Information Systems* **2023**. doi:10.1007/s10115-022-01826-x.
29. Qiao, B.; Zou, Z.; Huang, Y.; Fang, K.; Zhu, X.; Chen, Y. A joint model for entity and relation extraction based on BERT. *Neural Computing and Applications* **2022**, *34*. doi:10.1007/s00521-021-05815-z.
30. Agrawal, A.; Tripathi, S.; Vardhan, M.; Sihag, V.; Choudhary, G.; Dragoni, N. BERT-Based Transfer-Learning Approach for Nested Named-Entity Recognition Using Joint Labeling. *Applied Sciences* **2022**, *12*, 976;. doi:10.3390/app12030976.
31. Zhang, B.; Hu, Y.; Xu, D.; Li, M.; Li, M. SKG-Learning: a deep learning model for sentiment knowledge graph construction in social networks. *Neural Computing and Applications* **2022**, *34*. doi:10.1007/s00521-022-07028-4.
32. Qi, Z.; Wang, H.; Zhang, H. A Dual-Store Structure for Knowledge Graphs. *IEEE Transactions on Knowledge and Data Engineering* **2021**, *PP*, 1–1. doi:10.1109/TKDE.2021.3093200.
33. Szeremeta, Ł.; Tomaszuk, D. Document-oriented RDF graph store. *Studia Informatica* **2017**, *38*, 31–43.
34. Yu, H.; Li, H.; Mao, D.; Cai, Q. A domain knowledge graph construction method based on Wikipedia. *Journal of Information Science* **2020**, *47*, 016555152093251. doi:10.1177/0165551520932510.
35. Lin, J.; Zhao, Y.; Huang, W.; Liu, C.; Pu, H. Domain knowledge graph-based research progress of knowledge representation. *Neural Computing and Applications* **2021**, *33*. doi:10.1007/s00521-020-05057-5.
36. Baclawski, K.; Bennett, M.; Berg-Cross, G.; Schneider, T.; Sharma, R.; Singer, J.; Sriram, R. Ontology summit 2020 communiqué: Knowledge graphs. *Applied Ontology* **2021**, *16*, 1–19. doi:10.3233/AO-210249.

37. Liu, B.; Wu, J.; Yao, L.; Ding, Z. Ontology-based Fault Diagnosis: A Decade in Review. ICCMS 2019: Proceedings of the 11th International Conference on Computer Modeling and Simulation, 2019, pp. 112–116. doi:10.1145/3307363.3307381.
38. Lira Nunez, D.; Borsato, M. OntoProg: An ontology-based model for implementing Prognostics Health Management in mechanical machines. *Advanced Engineering Informatics* **2018**, *38*, 746–759. doi:10.1016/j.aei.2018.10.006.
39. Su, Y.; Liang, X.; Wang, H.; Wang, J.; Pecht, M. A Maintenance and Troubleshooting Method Based on Integrated Information and System Principles. *IEEE Access* **2019**, *PP*, 1–1. doi:10.1109/ACCESS.2019.2915327.
40. Tang, X.; Xiao, M.; Hu, B.; Pan, D. Exchanging deep knowledge for fault diagnosis using ontologies. *International Journal of Reasoning-based Intelligent Systems* **2020**, *12*, 117. doi:10.1504/IJRIS.2020.106805.
41. li, J.; Sun, A.; Han, R.; Li, C. A Survey on Deep Learning for Named Entity Recognition. *IEEE Transactions on Knowledge and Data Engineering* **2020**, *PP*, 1–1. doi:10.1109/TKDE.2020.2981314.
42. Kenton, J.D.M.W.C.; Toutanova, L.K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT, 2019, pp. 4171–4186.
43. Li, X.; Feng, J.; Meng, Y.; Han, Q.; Wu, F.; Li, J. A Unified MRC Framework for Named Entity Recognition. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 5849–5859.
44. Yang, P.; Cong, X.; Sun, Z.; Liu, X. Enhanced Language Representation with Label Knowledge for Span Extraction. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 4623–4635.
45. Wei, Z.; Su, J.; Wang, Y.; Tian, Y.; Chang, Y. A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; Association for Computational Linguistics: Online, 2020; pp. 1476–1488. doi:10.18653/v1/2020.acl-main.136.

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