

Article

# Automated Equipment Defect Knowledge Graph Construction for Power Grid Regulation

Wei Liu <sup>1,†</sup>, Yanghao Gu <sup>2,†</sup>, Zhiqiang Zeng <sup>3</sup>, Donglian Qi <sup>2,\*</sup>, Dezhi Li <sup>1</sup>, Yuanyuan Luo <sup>3</sup>, Qi Li <sup>2</sup> and Su Wei <sup>3</sup><sup>1</sup> State Grid Chongqing Electric Power Co., Ltd., Chongqing 400015, China<sup>2</sup> College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China<sup>3</sup> Electric Power Research Institute, State Grid Chongqing Electric Power Co., Ltd., Chongqing 400015, China

\* Correspondence: qidl@zju.edu.cn

† These authors contributed equally to this work.

**Abstract:** The normal operation of automated equipment is essential for power grid regulation, making the accurate identification and diagnosis of defects in this equipment highly significant. Constructing a knowledge graph for automated equipment defects offers an effective solution to challenges such as delayed reporting, low efficiency, and data omissions in manually recorded defects. To address this, we developed a framework for constructing an automated equipment defect knowledge graph by designing appropriate patterns and data layers. For knowledge extraction, we introduced two models: RoBERTa-BiLSTM for named entity recognition (NER) and ALBERT-BiGRU for relation extraction (RE), both of which demonstrated improved performance in their respective tasks. Additionally, we applied the KBGAT model for knowledge graph completion. Finally, Neo4j was used for storing, visualizing, and analyzing the knowledge graph, highlighting its significance in the operation of power grids and the advancement of digital power systems.

**Keywords:** defects in automated equipment; knowledge graph construction; knowledge processing; knowledge storage



**Citation:** Liu, W.; Gu, Y.; Zeng, Z.; Qi, D.; Li, D.; Luo, Y.; Li, Q.; Wei, S. Automated Equipment Defect Knowledge Graph Construction for Power Grid Regulation. *Electronics* **2024**, *13*, 4430. <https://doi.org/10.3390/electronics13224430>

Academic Editor: Dorin Petreus

Received: 18 October 2024

Revised: 7 November 2024

Accepted: 9 November 2024

Published: 12 November 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Nowadays, automated systems play a crucial role in industrial production [1]. They not only significantly improve production efficiency but also effectively reduce production costs, providing strong support for corporate sustainable development [2]. However, as the scale and complexity of automation systems continue to expand, equipment defect issues become more prominent [3]. If these issues are not properly managed, they can severely affect the stable operation and production efficiency of automated systems. The main characteristics of defects in automated equipment can be summarized as follows: high failure rates, diverse defect types, and complex maintenance and repair procedures.

Due to the above characteristics of defects in automated equipment, scientific and efficient defect management is vital for power grid regulation [4]. In general, defect management for power grid automation equipment is a comprehensive system. Currently, the defect management methods adopted mainly focus on a closed-loop approach [5], which can be summarized into four stages: “defect detecting, defect recording, defect rectifying, and operational verification”. Among these, the “defect recording” stage provides data support for subsequent defect rectification and is the most critical part of the entire automated equipment defect management system.

The common method for recording defects in automated equipment is manual handling by personnel. However, this manual management method has several issues. First of all, manual updating of defect reports can result in delays, making it difficult to update data in a timely manner. Aside from this, the efficiency of manually recording defect data is low, affecting the ability to quickly respond to and address defects. Moreover, manually entered information is often inconsistent, and data omissions or losses may occur.

Due to these shortcomings, numerous studies have been conducted in the power grid field to automate the defect recording process. Qing [6] proposed a quick response (QR) code-based defect management system, capitalizing on its large data capacity, high security, and robust anti-counterfeiting capabilities to enhance the safety, stability, and economic efficiency of equipment operations. Gao et al. [7] developed a big data analysis algorithm to create an automatic identification model for suspected familial defects in automated equipment, incorporating physical “ID” tracking to monitor the distribution of the same equipment, thereby improving defect identification accuracy. Zhang et al. [8] employed modern information platforms to enable equipment status evaluation and batch defect diagnosis through multidimensional analysis and multi-platform interaction, thus improving the quality and efficiency of defect analysis. Huang et al. [9] introduced an intelligent factory framework which integrates deep learning algorithms with IoT technology into defect detection systems, offering a more comprehensive solution for defect detection.

Knowledge graphs (KGs), as a structured knowledge representation method, organize complex information in the form of graphs, enabling computers to store, retrieve, and infer knowledge more effectively [10]. With the rapid development of artificial intelligence technologies, the application of KGs in the power industry has become a research hotspot [11]. The power system is a highly complex and dynamically changing system, involving a vast amount of equipment, operational procedures, safety standards, and real-time data. Traditional data processing methods face many challenges in handling this information, whereas knowledge graph technology, with its advantages in information organization, retrieval, and reasoning, provides a new solution to these problems. Meng et al. [12] developed a novel method to recognize power equipment entities based on bidirectional encoder representation from transformers (BERT), facilitating the construction of a power equipment fault knowledge graph. Tang et al. [13] introduced a method for constructing a power equipment KG by combining heterogeneous and multi-source data, thus enhancing the dispatching efficiency of power equipment. Cui [14] integrated KGs with the Internet of Things (IoT) to enable real-time monitoring of a power grid and its surrounding environment.

By applying KGs to the recording of automated equipment defects, the graph structure of KGs can be effectively utilized to analyze and process data, improving the accuracy of defect diagnosis and addressing the shortcomings of manual recording. However, existing electric power KGs primarily focus on the construction of KGs themselves, lacking further processing and refinement of the KGs.

To address these issues, we built upon the construction of automated equipment defect KGs by incorporating knowledge graph completion to enhance the practical usability of the constructed KG. To improve the effectiveness of the constructed knowledge graph, we applied AI-based methods. AI technologies provide several enhancements throughout the knowledge graph construction process. First, AI can increase the efficiency and accuracy of information extraction, accelerating the construction process. Secondly, by leveraging deep learning and graph inference techniques, AI can automatically identify the root causes of equipment defects, predict potential failure modes, and generate new knowledge relationships, further enhancing the intelligence of a knowledge graph. Furthermore, AI can utilize real-time equipment data and maintenance records to autonomously update and refine the knowledge graph, ensuring that it continuously reflects the equipment’s current operational status and potential risks and thereby addressing the timeliness issues associated with static knowledge graphs.

Based on the above discussion, the main contributions of this paper can be summarized as follows:

- A framework of the construction of automated equipment defect KGs is designed and implemented, effectively solving the problem of storing defect knowledge for automated equipment.

- In the construction process, the RoBERTa-BiLSTM model is used for named entity recognition (NER), and the ALBERT-BiGRU model is used for relation extraction (RE), with both achieving performance improvements in their respective tasks.
- We completed the constructed KG using the KBGAT algorithm [15] and stored and visualized it in the Neo4j graph database, resulting in a complete equipment automated defect KG.

## 2. Research Methodology

### 2.1. Overview of Construction Methods

In this section, we will provide an overall overview of the methods for constructing an automated equipment defect knowledge graph. Firstly, we need to determine the data source and preprocess the data, which will be detailed in Section 2.2. Secondly, we will develop an overall construction framework that includes the schema and construction steps of the knowledge graph, which will be demonstrated in Section 2.3. Finally, we will elaborate on each step and conduct specific experiments to validate the effectiveness in Sections 3 and 4.

### 2.2. Data Resource

The data used to construct knowledge graphs for automated defect equipment can be diverse, typically including unstructured and semi-structured data [16]. Unstructured data refer to information which does not adhere to a fixed format or unified data model, often presented in text form. Semi-structured data, meanwhile, fall between fully structured and unstructured formats, possessing some structural characteristics but lacking the rigid data models of traditional relational databases, typically appearing in formats such as JSON or tables. The knowledge graph data in this study primarily originate from unstructured text in selected chapters of e-books analyzing typical cases in automated systems, supplemented by semi-structured tabular data on defect records from a regional power grid's outage management system (OMS). OMS is an information system used in power grids and other industrial automation systems to monitor, control, and manage system operations [17]. In the context of defect management for automated equipment, an OMS provides functions such as defect data entry, intelligent diagnostics, and decision support for handling defects, improving both the efficiency and accuracy of defect management.

To better leverage the data collected for knowledge graph (KG) construction, we performed extensive data processing. For unstructured data, we filtered out unclear, incomplete, and overlapping sentences. After this preprocessing step, we retained a total of 1675 pieces of data for subsequent tasks. For semi-structured data, we parsed the information and classified its various elements to enrich the set of entities. This resulted in the extraction of 3294 entities, which will be used in the next named entity recognition (NER) task.

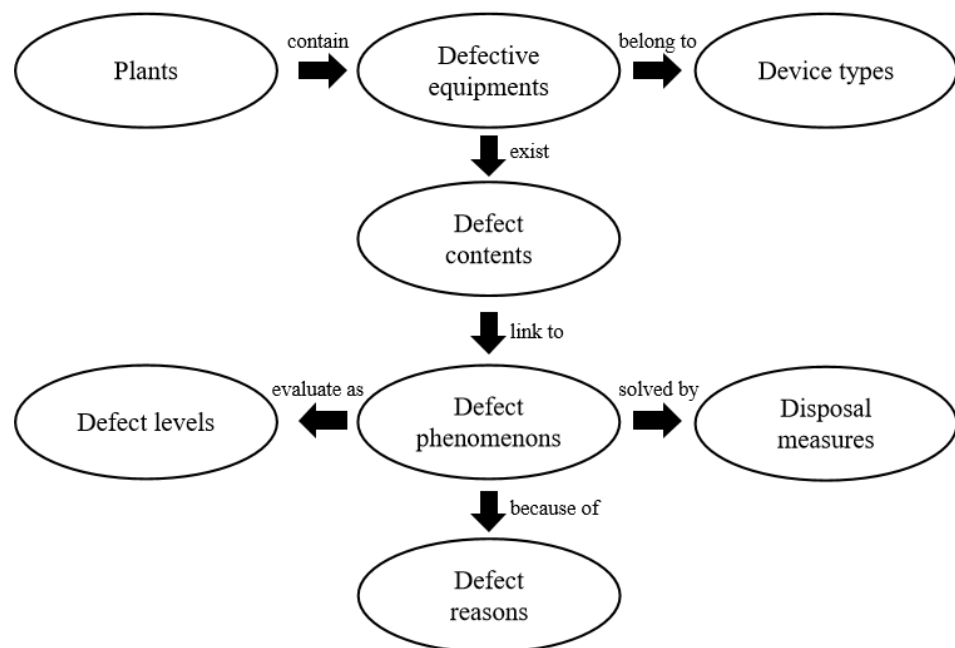
### 2.3. Overall Framework

There are three commonly used frameworks for constructing knowledge graphs: top-down, bottom-up, and hybrid [18]. In the top-down approach, structured data sources like encyclopedic websites are used to extract ontologies and schema information from high-quality data to populate the knowledge base [19]. Conversely, the bottom-up method involves extracting resource models from available public data using technical methods, adding higher-confidence information to the knowledge base. The hybrid method combines both approaches, using mutual mapping to update and iterate on the data, thereby improving the reliability of the knowledge graph.

Given the unstructured nature of defect text in power automation, we adopted a hybrid approach for KG construction. First, the schema of the KG is built using a top-down method. A schema is a domain-specific data model defining types of meaningful concepts and their attributes within that domain [20]. In simple terms, a schema is a "metadata" representation of knowledge, specifying the attributes of entities, the relationships between

them, and the constraints on both. The main purpose of a schema is to structure entities and clearly define their relationships.

A schema consists of entity types and relationship types. Entity types define a class of data instances with a common structure, while relationship types describe the associations between entities. Common schema construction often involves merely listing entity and relationship types, which may not clearly reflect logical relationships. In this paper, we constructed the schema by using a logical flowchart, creating a corresponding domain knowledge system. We predefined eight entity labels: “plants”, “defective equipment”, “device types”, “defect contents”, “defect phenomena”, “disposal measures”, “defect levels”, and “defect reasons”. After that, we designed seven relationship types between two entity labels separately: “contain”, “belong to”, “exist”, “link to”, “evaluate as”, “solved by”, and “because of”. Then, we used these entity labels and relation types to construct the whole schema, as shown in Figure 1.



**Figure 1.** Schema of defect knowledge graph.

Meanwhile, we applied a bottom-up approach for the data layer of the KG, consisting of named entity recognition (NER), relation extraction (RE), knowledge fusion, knowledge processing, knowledge updating, and knowledge storage. NER and RE, collectively referred to as knowledge extraction, are foundational to building a knowledge graph. Knowledge fusion integrates knowledge from various sources to form a unified, consistent representation. Knowledge processing involves refining and enhancing raw data or existing knowledge to improve its quality, accuracy, and usability. Knowledge updating is a continuous process in managing and maintaining the knowledge graph to ensure the data remain current, accurate, and relevant. Knowledge storage involves storing the constructed knowledge graph in a database for subsequent processing and updates. The overall framework is illustrated in Figure 2.

In this study, we pay attention to the stages of knowledge extraction, knowledge processing, and knowledge storage, which are crucial for the construction and application of automated equipment defect knowledge graphs. Figure 1 illustrates the schema which guided our knowledge extraction process, defining the entity types and relationship types which are essential for capturing the domain-specific knowledge. Figure 2 outlines the data layer, where we performed knowledge extraction using models like RoBERTa-BiLSTM for NER and ALBERT-BiGRU for RE. The extracted knowledge is then processed to enhance the graph’s completeness through techniques like knowledge graph completion, as detailed

in Section 4. Finally, the constructed knowledge graph is stored and visualized using the Neo4j graph database, as discussed in Section 5.

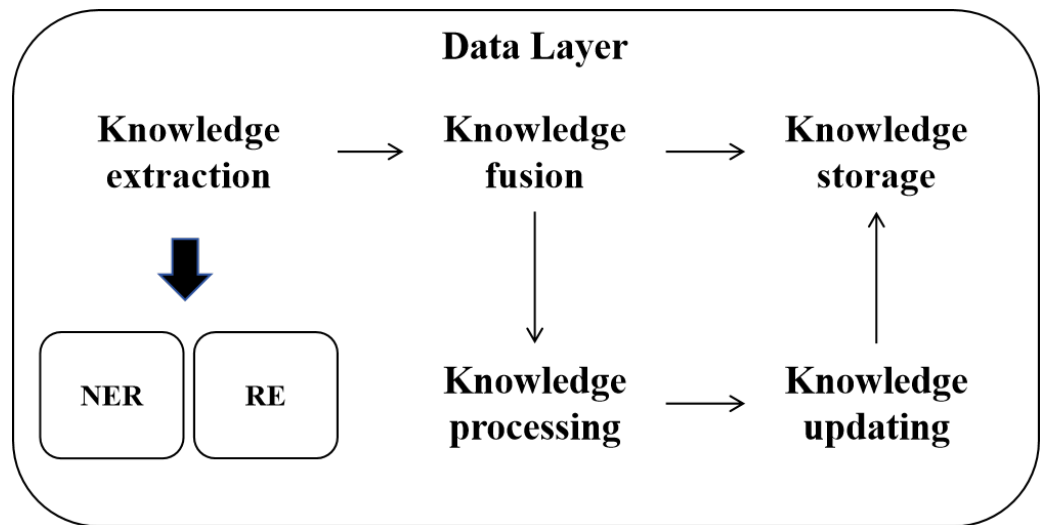


Figure 2. Data layer of defect knowledge graph.

### 3. Knowledge Extraction

#### 3.1. Named Entity Recognition

Named entity recognition (NER) [21] is a fundamental task in natural language processing (NLP) [22] which focuses on extracting entities with specific meanings such as names of people, locations, and other important information from unstructured text [23]. The primary objective of NER is to identify and capture key entities within natural language texts, thereby facilitating a deeper understanding of the text’s content. Before performing NER, unstructured text must undergo sequence labeling. We employed the BIO labeling scheme [23], where B (begin) marks the beginning of an entity, I (inside) indicates subsequent parts of the same entity, and O (outside) signifies tokens which do not belong to any entity, as shown in the example provided in Table 1. (B-DEV indicates the beginning of the device, while I-DEV indicates the inside of the device. Similarly, B-DEF indicates the beginning of a defect, while I-DEF indicates the inside of a defect.)

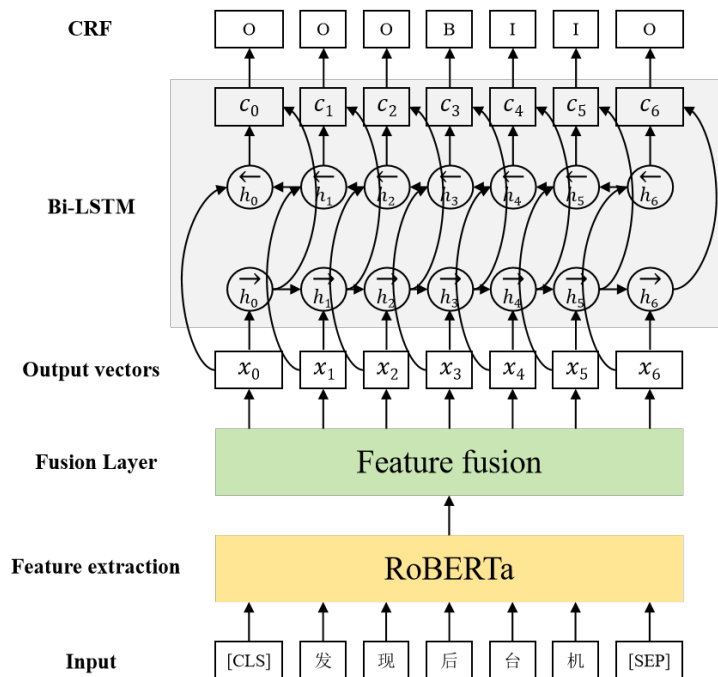
Table 1. Example of BIO annotation.

Sentence	Annotation
发	O
现	O
后	B-DEV
台	I-DEV
机	I-DEV
遥	B-DEF
测	I-DEF
刷	I-DEF
新	I-DEF
不	I-DEF
正	I-DEF
常	I-DEF

The sentence “发现后台机遥测刷新不正常” means “discovered abnormal telemetry refresh in the backend machine” in English.

Considering the characteristics of automated defect text, we employed the RoBERTa-BiLSTM model for named entity recognition (NER) on automated equipment defect data. RoBERTa [24], developed by Facebook AI Research (FAIR), is a pretrained language rep-

resentation model based on the BERT architecture. Compared with traditional BERT, RoBERTa more effectively captures contextual information and introduces a dynamic masking strategy where a random subset of tokens is selected for prediction in each iteration, improving the fixed masking mechanism of BERT. Bidirectional long short-term memory (Bi-LSTM) [25], a specialized recurrent neural network (RNN), combines LSTM units with bidirectional processing, providing robust representation capabilities to capture complex patterns in sequential data. In this model, RoBERTa is used for text feature extraction in the NER task, while Bi-LSTM is applied for sequence labeling. The overall architecture of the model is illustrated in Figure 3. In Figure 3, the sentence “发现后台机” represents the input sentence, which means “discovered backend machine” in English.



**Figure 3.** Structure diagram of RoBERTa-BiLSTM.

Figure 3 illustrates the architecture of the RoBERTa-BiLSTM model employed for NER in the context of automated equipment defect data. The model consists of two main components: the RoBERTa encoder and the BiLSTM layer with a conditional random field (CRF) layer on top. The RoBERTa encoder, pretrained on a large corpus, provides a deep contextualized representation of the input text. After feature fusion in the fusion layer, the representation is then fed into the BiLSTM layer, which captures sequential dependencies within the text. The BiLSTM’s forward and backward passes allow it to consider the context from both directions, enhancing the model’s ability to understand the surrounding text for each token. Finally, the CRF layer is used to predict the most likely sequence of labels, taking into account the transitions between different entities. This model architecture leverages the strengths of both deep contextualized word representations and sequential modeling, which is crucial for accurately identifying entities in unstructured text.

### 3.2. Relation Extraction

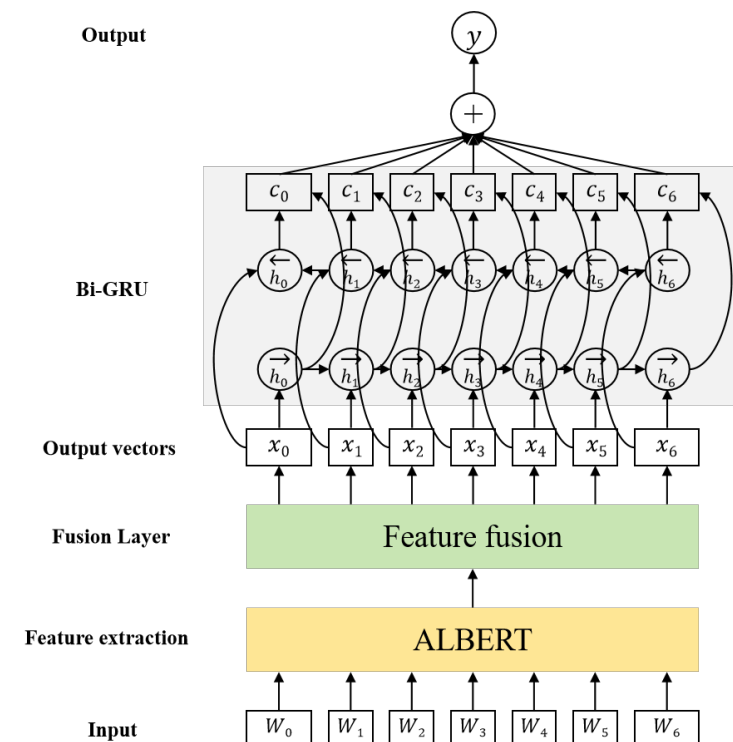
Relation extraction (RE) is a key task in natural language processing (NLP) and one of the fundamental components of information extraction [26]. Its goal is to identify relationships between entities (usually noun phrases or pronouns) within textual data. Building on named entity recognition (NER), RE forms the foundation for enabling intelligent systems to understand natural language and construct knowledge bases, playing a crucial role in enhancing a machine’s semantic understanding. Similar to the pre-labeling required

for NER, RE also necessitates predefined relationships between entities. We list all the predefined relationship types in Table 2.

**Table 2.** Entity relation categories.

Head Entity	Tail Entity	Relationship
Device types	Defect types	exist
Defect types	Defect contents	link to
Defect types	Defect levels	belong to
Defect types	Disposal measures	decide
Defect contents	Defect phenomena	link to
Defect phenomena	Defect reasons	cause

We employed ALBERT-BiGRU to address the relation extraction task. A Lite BERT (ALBERT) [27], developed by Google, was designed to reduce the number of parameters while maintaining or even enhancing performance. ALBERT reduces parameters through cross-layer sharing and proposes sentence order prediction (SOP), a new training method which replaces BERT’s next sentence prediction (NSP) and helps the model capture richer semantic information. Furthermore, with fewer parameters, ALBERT loads and runs faster, making it particularly suited for the computational environment in this study. The bidirectional gated recurrent unit (Bi-GRU) is a type of RNN which combines GRU mechanisms with bidirectional processing [28]. Compared with Bi-LSTM, Bi-GRU offers faster training speeds and fewer parameters, making it more appropriate for resource-constrained relation extraction tasks. We integrated ALBERT with Bi-GRU, following a similar approach to that used for NER. ALBERT serves as the feature extractor, while Bi-GRU handles sequence modeling, enhancing the features extracted by ALBERT to better capture the complex patterns between entities and their relationships. The model’s overall architecture is illustrated in Figure 4.



**Figure 4.** Structure diagram of ALBERT-BiGRU.

Figure 4 depicts the ALBERT-BiGRU model structure used for RE tasks within the automated equipment defect dataset. The model is composed of two primary sections—the

ALBERT module and the BiGRU layer—followed by a fully connected layer for classification. ALBERT, a lightweight variant of BERT, was utilized here for its efficiency and performance, providing a compact yet rich representation of the input text through its parameter-sharing mechanism. This representation is then processed by the BiGRU layer which, similar to the BiLSTM, processes the sequence in both directions to capture comprehensive contextual information. The key difference between BiGRU and BiLSTM is the use of the gated recurrent unit (GRU), which simplifies the model architecture while maintaining effectiveness in sequence modeling. The output from the BiGRU layer is then fed into a series of fully connected layers designed to classify the relationship between the entities. This architecture is pivotal in accurately extracting and classifying relationships within the text, which is essential for constructing a comprehensive knowledge graph.

### 3.3. Experiments and Results

#### 3.3.1. Datasets

To ensure that the experimental set-up aligned with the primary focus of our study—automated equipment defect knowledge graph construction—we carefully curated a dataset which captured the complexity and diversity of real-world scenarios. This dataset was derived from a comprehensive subset of previously mentioned data sources, which included unstructured text from e-books analyzing typical cases in automated systems and semi-structured tabular data from a regional power grid’s outage management system (OMS). This combination offered a balanced representation of both the theoretical and practical aspects of defect management.

After segmenting lengthy sentences to improve model training efficiency, we obtained a total of 6785 text entries. These entries were manually annotated using the BIO method to ensure accurate, high-quality labels for model training. The detailed experimental dataset of entities can be found in Appendix A Table A1. The dataset was split into training, validation, and test sets, with careful consideration to ensure the model’s ability to generalize effectively to unseen data, which is critical for real-world applications.

A training-validation-test split of 8:1:1 is a widely adopted practice in machine learning. It strikes a balance between providing a large enough training set for the model to learn complex patterns and maintaining sufficient data for validation and testing. The training set, comprising 80% of the data, enables the model to learn from a diverse range of examples. The validation set, constituting 10% of the data, is used to fine-tune the model parameters and prevent overfitting. The test set, which is also 10% of the data, serves as an unbiased evaluation of the model’s performance on unseen data, offering a realistic measure of its effectiveness in practical scenarios.

#### 3.3.2. Model Parameter Settings

The experiments related to NER and RE were conducted on an Ubuntu 18.04 operating system utilizing four NVIDIA 3090 24 G GPUs with CUDA version 11.7 and employing PyTorch framework version 1.12.1. The training hyperparameters for both tasks are detailed in Table 3. For the evaluation metrics, precision, recall, and F1 score were used for both tasks [29].

**Table 3.** Hyperparameters configuration of NER and RE.

Hyperparameters	NER	RE
Number of epochs	10	20
Batch size	64	64
Learning rate	$1 \times 10^{-5}$	$1 \times 10^{-5}$
Max length of sequence	512	512
Optimizer	Adam	Adam



### 3.3.3. Comparison Experiment of NER

To assess the effectiveness of the RoBERTa-BiLSTM model adopted for the NER task in automated defect equipment text, experiments were conducted to compare several widely used deep learning models. The experimental results, obtained under consistent conditions, are presented in Table 4.

**Table 4.** Comparison of different NER models.

Model	Precision	Recall	F1 Score
GCN [30]	78.5	77.6	78.0
LSTM-CRF [31]	80.2	79.7	79.9
BiLSTM-CNN-CRF [32]	83.4	82.1	82.7
BERT [33]	86.8	84.9	85.8
<b>Ours</b>	<b>88.2</b>	<b>87.3</b>	<b>87.7</b>

The comparative analysis revealed that the RoBERTa-BiLSTM architecture achieved outstanding performance on the automated equipment defect dataset, attaining optimal results in precision, recall, and F1 score. Compared with traditional BERT-CRF methods, the proposed model improved the precision, recall, and F1 score by 1.61%, 2.83%, and 2.21%, respectively, providing substantial evidence for its effectiveness.

To further validate the effectiveness of our proposed model, we conducted ablation experiments. The results of the ablation experiments are presented in Table 5.

**Table 5.** Ablation experiments for the NER task.

Index	RoBERTa	BERT	BiLSTM	F1 Score
1	✓		✓	87.7
2	✓			86.9
3		✓	✓	86.5
4		✓		85.8

The symbol '✓' indicates that the model is adopted.

As is shown in Table 5, model 2 removed BiLSTM, which led to a 0.91% decrease in the F1 score. Model 3 used BERT-BiLSTM, and the F1 score was reduced by 1.37%. Model 4 used BERT separately, which reduced the F1 score by 2.17%.

### 3.3.4. Comparison Experiment of RE

To validate the effectiveness of the ALBERT-BiGRU model selected for the RE task in automated defect equipment text, comparisons were made with several traditional deep network models. The experimental results are shown in Table 6. Table 6 presents a comparison of different relation extraction (RE) models using three primary metrics: precision, recall, and F1 score. These metrics were selected as they are standard in evaluating the performance of classification models, particularly in natural language processing tasks such as RE. Precision measures the accuracy of the positive predictions, recall indicates the model's ability to find all relevant instances, and the F1 score provides a harmonic mean of precision and recall, offering a balance between the two.

**Table 6.** Comparison of different RE models.

Model	Precision	Recall	F1 Score
BiLSTM [25]	61.1	59.8	60.4
C-GCN [34]	63.2	60.6	61.9
BERT [33]	67.5	66.4	66.9
SpERT [35]	72.7	71.2	71.9
<b>Ours</b>	<b>75.2</b>	<b>72.0</b>	<b>73.6</b>

The comparative analysis indicates that the ALBERT-BiGRU architecture achieved excellent performance on the automated equipment defect dataset, obtaining optimal results for precision and F1 score. Compared with the SpERT model proposed by Markus Eberts, the adopted model improved the precision and F1 score by 3.44% and 2.36%, respectively, providing substantial evidence for its effectiveness.

Similarly, we conducted ablation experiments for RE tasks as well. The results of the ablation experiments are presented in Table 7.

**Table 7.** Ablation experiments for RE task.

Index	AIBERT	SpERT	BiGRU	F1 Score
1	✓		✓	73.6
2	✓			72.5
3		✓	✓	72.7
4		✓		71.9

The symbol '✓' indicates that the model is adopted.

As is shown in Table 7, model 2 removed BiGRU, which led to a 1.49% decrease in the F1 score. Model 3 used SpERT-BiGRU, and the F1 score was reduced by 1.22%. Model 4 used SpERT separately, which reduced the F1 score by 2.31%.

## 4. Knowledge Processing

### 4.1. Knowledge Graph Completion

Knowledge processing refers to the process of collecting, organizing, analyzing, transforming, and applying knowledge, aimed at enhancing its value and usability. Knowledge graph completion (KGC) is a technique in knowledge processing which fills in missing triples in a knowledge graph, thereby making it more complete. In constructing knowledge graphs for automated equipment defects, many rules are predefined manually, which can lead to gaps and omissions. Therefore, incorporating a knowledge completion step during the graph construction process is highly significant.

There are various methods for knowledge graph completion, including rule-based methods, tensor decomposition-based methods, and translation model-based methods (such as the Trans series). We adopted deep learning-based methods, planning to experiment with a series of deep learning models to identify the most effective one through comparison.

### 4.2. Model Parameter Settings

The experimental environment in this section is consistent with that described in Section 3, with some of the training hyperparameters provided in Table 8.

**Table 8.** Hyperparameter configuration of KGC.

Hyperparameters	KGC
Number of epochs	200
Batch size	32
Learning rate	$1 \times 10^{-4}$
Dropout	0.3

We used HITS@3, HITS@10, and MRR as evaluation metrics for the KGC task. HITS@ $n$  represents the average proportion of triples ranked at or below  $n$  in link prediction, where higher values are preferable. The mean reciprocal rank (MRR) measures the average reciprocal rank of correctly predicted triples in the overall prediction results. A higher MRR indicates better model performance. MRR can be expressed as follows:

$$MRR = \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{1}{rank_i} \quad (1)$$

The parameter  $|S|$  denotes the number of triples, while  $rank_i$  refers to the predicted ranking of the  $i$ th triplet link.

#### 4.3. Experiments and Results

To evaluate the effectiveness of KGC models for the equipment defect knowledge graph discussed in this paper, several mainstream deep learning models were compared, yielding the experimental results displayed in Table 9.

**Table 9.** Comparison of different KGC models.

Model	HITS@3	HITS@10	MRR
ConvE [36]	32.7	48.6	0.325
ConvKB [37]	31.2	45.9	0.314
R-GCN [38]	15.6	27.7	0.159
HAKA [39]	<b>46.4</b>	58.9	0.478
<b>KBGAT [15]</b>	44.2	<b>61.1</b>	<b>0.493</b>

The comparative analysis revealed that the HAKA and KBGAT models each excelled in different performance metrics. After careful consideration, the HITS@10 and MRR metrics were deemed more influential for the KGC task, leading to the selection of the KBGAT model for completing the equipment defect knowledge graph.

The KBGAT model leverages graph neural networks to effectively model entities and relationships while utilizing an attention mechanism to emphasize crucial information within the graph. By adjusting the weights based on the importance of different relationships, the attention mechanism enhances the model's completion performance, particularly in complex knowledge graphs. Additionally, the use of graph neural networks allows KBGAT to efficiently propagate information throughout the knowledge graph, capturing intricate dependencies between entities and relationships. In KGC tasks, KBGAT can predict missing triples, fill in gaps, and uncover new relationships hidden within a graph.

## 5. Knowledge Storage and Application

### 5.1. Knowledge Graph Storage and Visualization

This study employed the Neo4j graph database to store and visualize the constructed knowledge graph. Neo4j, a high-performance and network-oriented NoSQL graph database, excels at processing highly connected data. Using the proposed method for constructing the knowledge graph of automated equipment defects, a total of 11,351 entity nodes and 19,164 entity relationship edges were created and stored in the Neo4j database through the Py2neo module in Python 3.8 [40].

The data stored in Neo4j could be visualized, with Figure 5 providing a partial visualization of the equipment defect knowledge graph.

### 5.2. Application of Knowledge Graph

The proposed automated equipment defect knowledge graph has broad application prospects across multiple fields and scenarios. Three typical application scenarios include the following:

- **Intelligent fault diagnosis of substation equipment:** By integrating deep learning technology with the knowledge graph, a semantic graph which fuses multimodal information from substation equipment can be developed, enabling autonomous fault warnings and diagnosis and thereby enhancing the efficiency of power grid operation, maintenance, and management.
- **Intelligent transformer operation and inspection:** In the domain of power equipment operation and inspection, the knowledge graph facilitates machine comprehension of expert knowledge, covering multiple fields and ultimately improving equipment management efficiency.



- We proposed a construction framework for the automated equipment defect knowledge graph, primarily employing a top-down approach for building the schema and a bottom-up approach for constructing the data layer.
- To address the unstructured nature of defect texts for automated equipment, the RoBERTa-BiLSTM and ALBERT-BiGRU models were constructed for named entity recognition and relationship extraction, achieving improvements in their respective task metrics.
- Based on knowledge extraction, the extracted data underwent knowledge processing, mainly realized through knowledge graph completion techniques. Through comparative analysis, the KBGAT model demonstrated better performance for knowledge graph completion.
- Finally, the KG constructed in this study was stored and visualized using the Neo4j graph database, and the application prospects and value of the automated equipment defect knowledge graph were analyzed.

**Author Contributions:** Conceptualization, W.L. and Y.G.; methodology, W.L. and Y.G.; software, W.L. and Z.Z.; validation, W.L., Y.G. and D.L.; formal analysis, W.L. and Z.Z.; investigation, Y.G. and D.L.; resources, W.L. and Y.G.; data curation, W.L.; writing—original draft preparation, W.L. and Y.G.; writing—review and editing, W.L. and Y.G.; visualization, Y.L.; supervision, D.Q.; project administration, S.W.; funding acquisition, Q.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by the Technology Project of State Grid Corporation of China. (Grant No. 5108-202327070A-1-1-ZN).

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** Authors Wei Liu and Dezhi Li were employed by the company State Grid Chongqing Electric Power Co., Ltd. Zhiqiang Zeng, Yuanyuan Luo and Su Wei were employed by the company Electric Power Research Institute, State Grid Chongqing Electric Power Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

KG	Knowledge graph
QR	Quick response
IoT	Internet of Things
OMS	Outage management system
NER	Named entity recognition
RE	Relationship extraction
NLP	Natural language processing
FAIR	Facebook AI Research
BERT	Bidirectional encoder representation from transformers
Bi-LSTM	Bidirectional long short-term memory
RNN	Recurrent neural network
SOP	Sentence order prediction
NSP	Next sentence prediction
Bi-GRU	Bidirectional gated recurrent unit
KGC	Knowledge graph completion
MRR	Mean reciprocal rank
NoSQL	Not Only Structured Query Language

## Appendix A

**Table A1.** Detailed experimental dataset of entities.

Entity Label	Entity Example	Total Number of Entities
Plants	35 kV 茨竹变电站 (35 kV Cizhu Substation)	683
Defective equipment	后台机 (backend machine)	2187
Device types	计算机设备 (computer equipment)	40
Defect contents	子站测控装置及二次回路故障 (fault of substation measurement and control device and secondary circuit)	2913
Defect phenomena	遥测刷新不正常 (abnormal telemetry refresh)	3538
Defect levels	一般 (normal)	4
Disposal measures	修改电能量采集终端时间配置 (modify the time configuration of the electric energy collection terminal)	1550
Defect reasons	监控系统故障 (monitoring system malfunction)	436

## References

- Huang, X.; Qi, D.; Chen, Y.; Yan, Y.; Yang, S.; Wang, Y.; Wang, X. Distributed Self-Triggered Privacy-Preserving Secondary Control of VSG-Based AC Microgrids. *IEEE Trans. Smart Grid* **2024**. [[CrossRef](#)]
- Parasuraman, R.; Mouloua, M.; Molloy, R.; Hilburn, B. Monitoring of automated systems. In *Automation and Human Performance*; CRC Press: Boca Raton, FL, USA, 2018; pp. 91–115.
- de Oliveira Hansen, J.P.; da Silva, E.R.; Bilberg, A.; Bro, C. Design and development of automation equipment based on digital twins and virtual commissioning. *Procedia CIRP* **2021**, *104*, 1167–1172. [[CrossRef](#)]
- Sarathkumar, D.; Srinivasan, M.; Stonier, A.A.; Samikannu, R.; Dasari, N.R.; Raj, R.A. A technical review on classification of various faults in smart grid systems. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2021; p. 012152.
- Tong, X.; Liang, Z.; Li, N.; Zhao, B.; Li, K.; Zhang, G. Comprehensive Analysis and Evaluation of Shandong Power Grid Station Automation Equipment. In Proceedings of the 2020 Asia Energy and Electrical Engineering Symposium (AEEES), Hangzhou, China, 18–20 December 2020; pp. 632–637.
- Qing, L. Research on equipment defect management system based on QR code. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2019; p. 052058.
- Gao, Q.; Zhong, C.; Wang, Y.; Wang, P.; Yu, Z.; Zhang, J. Defect analysis of the same batch of substation equipment based on big data analysis algorithm. In *IOP Conference Series: Earth and Environmental Science*; IOP Publishing: Bristol, UK, 2021; p. 022093.
- Zhang, Z.; Yang, Q.; Xie, S.; Rao, Z.; Zhou, G.; Zhang, K. Application of Intelligent Management System for Defects of Substation Equipment. In Proceedings of the 2024 36th Chinese Control and Decision Conference (CCDC), Xi'an, China, 25–27 May 2024; pp. 2192–2196.
- Huang, D.C.; Lin, C.F.; Chen, C.Y.; Sze, J.R. The Internet technology for defect detection system with deep learning method in smart factory. In Proceedings of the 2018 4th International Conference on Information Management (ICIM), London, UK, 25–27 May 2018; pp. 98–102.
- Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Philip, S.Y. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 494–514. [[CrossRef](#)]
- Wang, J.; Wang, X.; Ma, C.; Kou, L. A survey on the development status and application prospects of knowledge graph in smart grids. *IET Gener. Transm. Distrib.* **2021**, *15*, 383–407. [[CrossRef](#)]
- Meng, F.; Yang, S.; Wang, J.; Xia, L.; Liu, H. Creating knowledge graph of electric power equipment faults based on BERT-BiLSTM-CRF model. *J. Electr. Eng. Technol.* **2022**, *17*, 2507–2516. [[CrossRef](#)]
- Tang, Y.; Liu, T.; Liu, G.; Li, J.; Dai, R.; Yuan, C. Enhancement of power equipment management using knowledge graph. In Proceedings of the 2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia), Kuala Lumpur, Malaysia, 21–24 May 2019; pp. 905–910.
- Cui, B. Electric device abnormal detection based on IoT and knowledge graph. In Proceedings of the 2019 IEEE International Conference on Energy Internet (ICEI), Chengdu, China, 11–13 October 2019; pp. 217–220.

15. Nathani, D.; Chauhan, J.; Sharma, C.; Kaul, M. Learning attention-based embeddings for relation prediction in knowledge graphs. *arXiv* **2019**, arXiv:1906.01195.
16. Kalyanpur, A.; Boguraev, B.K.; Patwardhan, S.; Murdock, J.W.; Lally, A.; Welty, C.; Prager, J.M.; Coppola, B.; Fokoue-Nkoutche, A.; Zhang, L.; et al. Structured data and inference in DeepQA. *IBM J. Res. Dev.* **2012**, *56*, 10–11. [[CrossRef](#)]
17. Jensen, M.; Sel, C.; Franke, U.; Holm, H.; Nordström, L. Availability of a SCADA/OMS/DMS system—A case study. In Proceedings of the 2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe), Barcelona, Spain, 6–8 December 2010; pp. 1–8.
18. Haibo, S.; Sunxin, L.; Weiyue, T.; Li, L. Construction of Knowledge Graph of Power Communication Planning based on Deep Learning. In Proceedings of the 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 4–6 March 2022; pp. 843–851.
19. Hofer, M.; Obraczka, D.; Saeedi, A.; Köpcke, H.; Rahm, E. Construction of Knowledge Graphs: Current State and Challenges. *Information* **2024**, *15*, 509. [[CrossRef](#)]
20. Zhou, D.; Zhou, B.; Zheng, Z.; Soylyu, A.; Savkovic, O.; Kostylev, E.V.; Kharlamov, E. Schere: Schema reshaping for enhancing knowledge graph construction. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, San Francisco, CA, USA, 7–11 November 2022; pp. 5074–5078.
21. Marrero, M.; Urbano, J.; Sánchez-Cuadrado, S.; Morato, J.; Gómez-Berbís, J.M. Named entity recognition: Fallacies, challenges and opportunities. *Comput. Stand. Interfaces* **2013**, *35*, 482–489. [[CrossRef](#)]
22. Kang, Y.; Cai, Z.; Tan, C.W.; Huang, Q.; Liu, H. Natural language processing (NLP) in management research: A literature review. *J. Manag. Anal.* **2020**, *7*, 139–172. [[CrossRef](#)]
23. Li, J.; Sun, A.; Han, J.; Li, C. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 50–70. [[CrossRef](#)]
24. Liu, Y. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
25. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
26. Pawar, S.; Palshikar, G.K.; Bhattacharyya, P. Relation extraction: A survey. *arXiv* **2017**, arXiv:1712.05191.
27. Lan, Z. Albert: A lite bert for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
28. Li, X.; Ma, X.; Xiao, F.; Xiao, C.; Wang, F.; Zhang, S. Time-series production forecasting method based on the integration of Bidirectional Gated Recurrent Unit (Bi-GRU) network and Sparrow Search Algorithm (SSA). *J. Pet. Sci. Eng.* **2022**, *208*, 109309. [[CrossRef](#)]
29. Zhou, Z.H. *Machine Learning*; Springer Nature: Berlin/Heidelberg, Germany, 2021.
30. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
31. Zeng, D.; Sun, C.; Lin, L.; Liu, B. LSTM-CRF for drug-named entity recognition. *Entropy* **2017**, *19*, 283. [[CrossRef](#)]
32. Ma, X. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. *arXiv* **2016**, arXiv:1603.01354.
33. Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
34. Zhang, Y.; Qi, P.; Manning, C.D. Graph convolution over pruned dependency trees improves relation extraction. *arXiv* **2018**, arXiv:1809.10185.
35. Eberts, M.; Ulges, A. Span-based Joint Entity and Relation Extraction with Transformer Pre-training. *arXiv* **2019**, arXiv:1909.07755.
36. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2d knowledge graph embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
37. Nguyen, D.Q.; Nguyen, T.D.; Nguyen, D.Q.; Phung, D. A novel embedding model for knowledge base completion based on convolutional neural network. *arXiv* **2017**, arXiv:1712.02121.
38. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Greece, 3–7 June 2018; pp. 593–607.
39. Zhang, Z.; Cai, J.; Zhang, Y.; Wang, J. Learning hierarchy-aware knowledge graph embeddings for link prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 2–7 February 2020; pp. 3065–3072.
40. Gupta, S. *Building Web Applications with Python and Neo4j*; Packt Publishing Ltd.: Birmingham, UK, 2015.

**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.