

Article

Knowledge Graphs for Railway Accident Profiling: Research and Applications

Xiaoqin Lian ¹, Zijie Wang ¹, Haoyang Yuan ¹, Chao Gao ^{1,*}, Zhibo Cheng ², Yanhua Wu ²
and Guangjing Zheng ¹

¹ School of Computer and Artificial Intelligence, Beijing Technology and Business University, Beijing 102488, China; lianxq@th.btbu.edu.cn (X.L.); 2431062125@st.btbu.edu.cn (Z.W.); 2330601027@st.btbu.edu.cn (H.Y.); 2531062212@st.btbu.edu.cn (G.Z.)

² Institute of Computing Technologies, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China; chengzhibo@rails.cn (Z.C.); wuyanhua@rails.cn (Y.W.)

* Correspondence: gaochao9158@btbu.edu.cn

Featured Application

The proposed railway accident knowledge graph and GraphRAG retrieval framework can support the structured organization and auxiliary retrieval of railway accident information, providing technical support for accident cause analysis, accident review, and railway safety management.

Abstract

In railway operations and safety oversight, vast amounts of accident-related data are recorded in unstructured textual formats, posing challenges for efficient information extraction and analysis. To address this, we constructed a knowledge graph from railway accident profile texts and integrated it with a Graph Retrieval-Augmented Generation (GraphRAG)-enhanced retrieval framework to support both basic and composite queries on railway accident information. First, railway accident profile texts were preprocessed, and Easy Data Augmentation (EDA) was used to expand samples of different types, thereby constructing a railway accident profile text dataset for subsequent knowledge extraction. We then introduced a deep learning model, RoBERTa-CNN-BiLSTM-CRF (RCBC), for automatic extraction of seven categories of entities, including accident IDs and causes. To extract semantic relations, we designed a prompt-based template leveraging large language models (LLMs). To mitigate information loss during entity extraction, a GCN-Attention-LLM (GA-LLM) model was further designed for knowledge graph completion. Experimental results show that RCBC achieves MicroF scores above 85% across entity extraction tasks, while GA-LLM attains an average Hits@3 of 82.84% in knowledge completion. In tests on basic and composite questions, LLM-GraphRAG outperformed both LLM and LLM-RAG in faithfulness, semantic similarity, context precision, and context recall. The resulting knowledge graph contains 1493 entities and 1832 relations. Combined with the retrieval framework, the system enables access to key railway accident information and offers technical support for intelligent railway safety management.

Keywords: railway accident; knowledge graph; GraphRAG; named entity recognition; large language model; intelligent risk analysis



Academic Editor: Krzysztof Zboński

Received: 7 August 2026

Revised: 6 September 2026

Accepted: 9 September 2026

Published: 11 September 2026

Copyright: © 2026 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](#).

1. Introduction

In recent years, China has proactively aligned with global technological trends by implementing strategies aimed at the digitalization and intelligent transformation of its high-speed rail system. These efforts have consistently fostered the in-depth application and innovation of relevant technologies in the high-speed railway sector [1]. However, with the continuous expansion of the railway network, the assessment and analysis of operational safety risks have become increasingly vital [2].

In the operation and safety supervision of railways, a vast amount of accident-related information is recorded in textual form. Effectively processing this textual data has become crucial for enhancing railway safety management. In recent years, academic research in railway safety has primarily focused on analyzing accident causality patterns, providing theoretical support for subsequent safety prevention and control measures. However, significant gaps remain in the critical area of digital extraction of accident information. Zhang et al. [3] identified the causal factors and their interrelationships in human factors during railway maintenance operations, aiming to analyze accident patterns and propose targeted preventive and control measures. Dai et al. [4] developed a comprehensive analytical framework for human factor-related safety accidents in the railway locomotive system, with the goal of identifying accident causal factors and their interrelationships. However, existing research predominantly concentrates on analyzing accident causal factors, with limited exploration of methods for the digital extraction of accident information.

Railway accident profile texts refer to records produced by relevant departments after railway transportation accidents, including basic accident information, the accident process, cause analysis, and handling information [5]. Such texts are typically unstructured, information-dense, and highly domain-specific. They constitute an important basis for railway safety supervision and a key data resource for accident analysis, risk assessment, and intelligent early warning. Railway accident profile texts contain relatively complete key information in a concise form and are therefore suitable for the digital extraction of critical information [6,7].

As a technical approach integrating semantic understanding and relation modeling, knowledge graphs can identify entities and their relations in text [8] and support the structured representation and semantic association of accident information [9]. Therefore, knowledge graph-based digital processing is well suited to the textual characteristics of railway accident data and can provide technical support for railway safety management. Knowledge graphs have been applied to a range of railway operation and maintenance tasks. In the equipment-failure domain, Lin et al. [10] proposed a knowledge graph construction method for high-speed railway turnout equipment faults, enabling structured representation and intelligent association of fault text. Li et al. [11] constructed a knowledge graph and application framework for railway transportation equipment faults and explored its use in intelligent diagnosis, risk management, and knowledge question answering. In passenger transport dispatching, Kong et al. [12] constructed a knowledge graph for high-speed railway passenger dispatching commands and proposed a closed-loop management process to support semantic parsing and intelligent application of dispatching commands. In railway construction, Zhu et al. [13] constructed a safety-quality-schedule knowledge graph for intelligent management of railway tunnel drilling and blasting. Song et al. [14] constructed knowledge graphs for subgrade, bridge, and tunnel design and combined deep learning with graph neural networks to support engineering scheme analysis and optimization.

Knowledge graph technologies have made progress in railway accident and related domains, but limitations remain in both data processing and practical application. At the data-processing level, railway accident profile texts contain dense professional terminology

and complex semantic logic, while accident-type samples are unevenly distributed; limited corpora may therefore affect how well knowledge extraction models learn different accident expressions. At the application level, existing accident-related knowledge graphs mainly focus on static knowledge storage, whereas graph completion and enhanced retrieval for complex associated information remain comparatively underexplored, limiting support for deeper semantic queries over railway accident information.

To address these limitations, this study proposes a digital knowledge-graph construction method for railway accident profile texts. First, the texts are organized and preprocessed, and Easy Data Augmentation (EDA) is used to expand the available samples and enrich the training corpus for subsequent entity and relation extraction. Second, entity extraction and relation extraction algorithms are used to extract knowledge from accident texts. A railway accident profile knowledge graph is then constructed in Neo4j, and a graph completion algorithm is used to improve graph structure. Finally, a Graph Retrieval-Augmented Generation (GraphRAG)-enhanced retrieval framework is introduced to support question answering for both basic and composite railway accident queries.

From a practical application perspective, the framework can transform key information distributed across accident texts into searchable structured knowledge, supporting rapid historical accident retrieval, cross-accident association analysis, and summaries of accident causes and outcomes, thereby providing auxiliary technical support for accident review and information retrieval by safety management personnel.

2. Materials and Methods

Knowledge graph construction approaches typically fall into three categories: knowledge-based top-down approaches, data-driven bottom-up approaches, and hybrid strategies that combine both [15]. Among these, the hybrid strategy merges the structural regularity of top-down methods with the flexible information mining capabilities of bottom-up approaches. Therefore, this study employs a hybrid knowledge graph construction method that integrates both top-down and bottom-up strategies. The research framework for constructing the railway accident profile knowledge graph is illustrated in Figure 1.

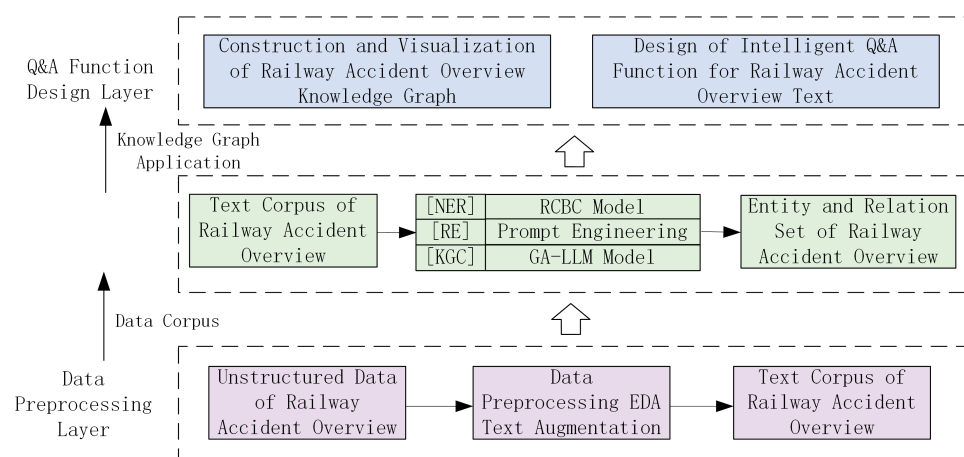


Figure 1. Research framework for the railway accident profile knowledge graph. Abbreviations: Q&A, question and answer; NER, named entity recognition; RE, relation extraction; KGC, knowledge graph completion; EDA, Easy Data Augmentation; RCBC, RoBERTa-CNN-BiLSTM-CRF; GA-LLM, GCN-Attention-LLM.

The three-layer structure of the railway accident profile knowledge graph construction framework consists of the data processing layer, the graph construction layer, and the management and application layer.

(1) Data Processing Layer

This layer is primarily responsible for preprocessing and augmenting railway accident profile texts. First, unstructured text records in comma-separated values (CSV) format are imported, and semantic annotation is used to systematically label concepts, entities, relations, and attributes. Then, to address the limited size of the original corpus and differences in the distribution of accident types, Easy Data Augmentation (EDA) is used to expand the training corpus [16,17]. Different augmentation strategies are adopted according to the structural and semantic characteristics of the three text segments. For the relatively standardized accident information and accident evaluation segments, replacement with entities of the same type is applied to key information such as accident IDs, responsible railway bureaus, railway lines, accident severity, and accident responsibility. For accident process segments, which contain denser railway terminology and tighter semantic logic, perturbations are mainly introduced outside named-entity spans. Common perturbations considered include random word deletion, insertion, homophone replacement, and word-order adjustment [16,17]. To minimize changes to core entities and the original semantic structure, the final augmentation strategy inserts stop words at non-named-entity positions. The augmented accident information, accident process, and accident evaluation segments are then randomly recombined to expand the training corpus.

(2) Graph Construction Layer

This layer relies on the constructed corpus to perform entity and relation extraction for the knowledge graph. First, the RoBERTa-CNN-BiLSTM-CRF (RCBC) model is developed to complete the entity extraction task. Then, large language models (LLMs) are employed to automate the extraction of relationships between entities through prompt engineering. Finally, the GCN-Attention-LLM (GA-LLM) model is designed to complete missing key entities.

(3) Management and Application Layer

This layer is mainly responsible for storing the constructed set of entities and relationships of railway accident profiles into the Neo4j graph database and generating a structured knowledge graph based on this, facilitating the visualization of the graph. Building upon this, GraphRAG technology is further integrated to construct a vectorized database of railway accident profiles, supporting question-and-answer functionality for both basic and composite types of railway accident-related queries.

2.1. Data Sources and Characteristics

The railway accident profile text dataset used in this study was obtained from the Institute of Computing Technologies, China Academy of Railway Sciences Corporation Limited. The structure of the accident profile data is presented in Table 1.

Table 1. Structure of railway accident profile text data.

Designation	Definition	Components
Accident Information Segment	Provides an overview of the accident	Accident ID, responsible railway bureau, railway line
Accident Process Segment	Describes the sequence of events leading to and during the accident	Accident cause, accident outcome
Accident Evaluation Segment	Describes classification and attribution of responsibility for the accident	Accident classification, accident responsibility

Railway accident profile texts consist of three components: accident information segments, accident process segments, and accident evaluation segments. Accident information segments describe basic accident information, including the accident ID, responsible railway bureau, and railway line. Accident process segments describe the accident occurrence process and contain accident causes and outcomes. Accident evaluation segments describe post-accident assessment information, including accident classification and accident responsibility.

The railway accident profile texts used in this study are internal railway-industry accident archives. Due to data-security, confidentiality, and authorization restrictions, these records cannot be collected at scale, openly shared, or provided to third parties in the same manner as general public corpora. Therefore, the study was conducted only within the authorized scope of data use, and the original accident archives are not publicly disclosed.

The original dataset contains 283 railway accident profile records covering 8 railway bureaus and 52 railway lines. Railway accidents are relatively infrequent events, and major accidents and some rare accident types occur even less frequently, which limits the number of available historical samples. In addition, data access and sharing restrictions constrain the amount of real data available for model training. Although the current dataset covers multiple railway bureaus and lines, accident types are unevenly distributed, and low-frequency or major accident scenarios remain underrepresented. Therefore, the reported results mainly reflect the terminology, segment structure, and entity-relation characteristics within the current data coverage and should not be used to infer the overall distribution of all railway accident types or model performance in uncovered scenarios.

All 283 accident records contain accident information, accident process, and accident evaluation segments. The 283 records are divided into training, validation, and test sets at a ratio of 70%:15%:15%, corresponding to 198, 42, and 43 records, respectively. The annotation schema contains seven entity types and seven relation types. Basic statistics of the dataset are shown in Table 2. This table describes the corpus size, source coverage, and composition of the knowledge annotation schema.

Table 2. Basic statistics of the railway accident profile dataset.

Statistic	Quantity	Description
Original Accident Records	283	Railway accident profile texts
Railway Bureaus Covered	8	Different railway bureaus
Railway Lines Covered	52	Different railway lines
Segment Types	3	Accident information, accident process, accident evaluation
Entity Types	7	Accident ID, responsible railway bureau, railway line, accident cause, accident outcome, accident severity, accident responsibility
Relation Types	7	Belongs to, includes, leads to, cause, outcome, classification, responsibility attribution

To reduce the influence of differences in formatting and terminology across accident records on knowledge extraction, dataset construction includes format standardization, removal of invalid information, and normalization of professional terminology. Seven core entity types are manually annotated using the beginning–inside–outside–end (BIOE) tagging scheme, with annotation conducted according to the information carried by the accident information, accident process, and accident evaluation segments. The annotations are manually reviewed. For samples with ambiguous entity boundaries or easily confused categories, decisions are made by considering contextual semantics, the accident causal chain, and the function of the text segment. For example, “emergency stop” may be annotated as an accident outcome in some contexts, but when the subsequent final outcome is “train delay”, it is treated as an intermediate factor in the causal chain and annotated as

an accident cause. The available annotation records do not contain independent duplicate annotations that can be used to compute inter-annotator agreement (IAA); therefore, no IAA value is reported. Annotation quality is controlled mainly through unified annotation guidelines and manual review.

In addition, different operating units may differ in recording practices, terminology, and level of detail, which may introduce reporting and sample-coverage biases. Format standardization, terminology normalization, and manual review can reduce differences in textual expression but cannot eliminate biases arising from archive coverage and accident-type composition.

2.2. Entity Extraction Method

Considering the dense professional terminology and context-dependent entity boundaries in railway accident profile texts, this study uses RCBC for entity extraction rather than directly applying a large language model. First, the RCBC components correspond explicitly to the processing stages of sequence labeling: RoBERTa provides contextual semantic representations, CNN supplements local compositional features, BiLSTM models bidirectional contextual dependencies, and CRF optimizes entity boundaries through label-transition constraints. Second, given the current dataset size, the training and inference costs of RCBC are relatively moderate, facilitating model training and result verification under a fixed label system. Finally, a large language model may generate entities that do not occur in the original text, whereas RCBC outputs are constrained by the input sequence and the BIOES-style label space and can therefore be traced more directly to the source text. Accordingly, the large language model is used for subsequent relation judgment, while entity extraction remains a supervised sequence-labeling task.

To construct the railway accident profile knowledge graph, this study defines seven core entities: responsible railway bureau, railway line, accident ID, accident cause, accident outcome, accident severity, and accident responsibility. For the entity extraction task, this study proposes the RCBC model, and its architecture is shown in Figure 2.

RCBC consists of RoBERTa, CNN, BiLSTM, and CRF. RoBERTa is used to obtain contextual semantic representations, CNN supplements local compositional features within entities, BiLSTM bidirectionally encodes the text sequence to model contextual dependencies, and CRF performs global decoding with label-transition constraints. These modules correspond to semantic representation, local feature extraction, contextual modeling, and label constraints, respectively, and are jointly used to identify the seven entity types.

This study first takes the railway accident profile text as input and converts it into a sequence of tokens containing textual information, where n represents the number of Token = [Token₁, Token₂, ..., Token _{n}]. The Token sequence is then fed into the RoBERTa and CNN modules, embedded into sequences $E = [E_1, E_2, \dots, E_n]$ and $T = [T_1, T_2, \dots, T_n]$, respectively. Sequences E and T are subsequently concatenated to obtain the final output sequence $h = [h_1, h_2, \dots, h_n]$, with each element h_i in the sequence calculated as shown in Equation (1).

$$h_i = \text{Concat}(E_i, T_i), \quad 1 \leq i \leq n \quad (1)$$

Here, E_i represents the i -th vector in the embedded sequence E , and T_i represents the i -th vector in the embedded sequence T .

Next, this study uses a bidirectional long short-term memory network (BiLSTM) to perform both forward and backward LSTM computations on the sequence h , obtaining the forward sequence $\vec{h}$ and the backward sequence $\overleftarrow{h}$, and then concatenates them to obtain the sequence $H = [\vec{h}, \overleftarrow{h}]$. Finally, this study uses the sequence H as input to the CRF model, calculates the conditional probabilities of each possible label sequence, and selects the sequence with the highest probability as the final output.

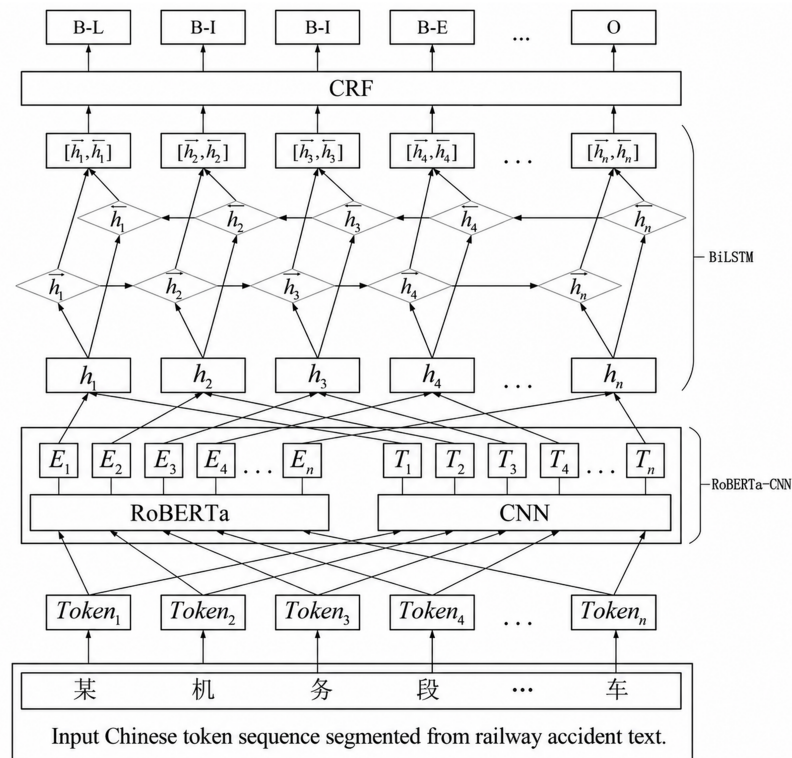


Figure 2. Architecture of the RCBC fusion model. The Chinese text shows a source-language token example.

2.3. Relation Extraction Method

The number of labeled samples available for relation extraction from railway accident profile texts is limited, while the relation types have relatively explicit domain-specific semantic constraints. Compared with supervised relation-classification models that require larger labeled datasets, large language models can use task descriptions and candidate entities to perform relation judgment under limited-data conditions [18]. Therefore, relation extraction is formulated as a constrained semantic-matching task: the accident text and recognized entities are provided as input, the selectable relation types and output format are restricted, and the large language model performs the relation judgment.

After completing the entity extraction task, this study introduces an LLM-based relation extraction method utilizing prompt engineering templates to further identify semantic relations between entities. This method constructs a structured prompt engineering template and interfaces with a large-scale pre-trained language model to achieve the automatic recognition and classification of entity relations. The prompt template designed in this study consists of three parts: the accident profile text input section, the entity extraction result section, and the entity-relation definition section. Its structure is shown in Figure 3.

In the accident profile text input section, each accident description is first separated and then provided sequentially to the model. The entity information and corresponding entity categories in the entity extraction result section are obtained from the RCBC entity extraction output. The entity-relation definition section standardizes the semantic relations between entities according to the predefined relation types. Seven relation types are defined in this study, as shown in Table 3.

To improve the reproducibility of the relation extraction procedure, Table 4 specifies the structured prompt template used for relation extraction based on the three-part framework described above. The accident text and recognized entity set are dynamically replaced for each accident, while the relation definitions, task constraints, and output format remain fixed.

Accident Overview Text Input Segment	[CLS]11*****07. On **** **, 2022, a passenger train of XX Bureau traveled to XX section. Sudden mountain floods collapsed the right cutting slope, and debris flow overflowed onto the track, causing the auxiliary locomotive to hit the debris flow and derail...												
Entity Extraction Result Segment	<table><tr><td>11*****07</td></tr><tr><td>Accident ID</td></tr></table>	11*****07	Accident ID	<table><tr><td>** Bureau</td></tr><tr><td>Affiliated Railway Bureau</td></tr></table>	** Bureau	Affiliated Railway Bureau	...	<table><tr><td>Sudden mountain flood</td></tr><tr><td>Accident Cause</td></tr></table>	Sudden mountain flood	Accident Cause	<table><tr><td>Deraiment</td></tr><tr><td>Accident Consequence</td></tr></table>	Deraiment	Accident Consequence
11*****07													
Accident ID													
** Bureau													
Affiliated Railway Bureau													
Sudden mountain flood													
Accident Cause													
Deraiment													
Accident Consequence													
Entity Relation Definition Segment	<table><tr><td>Affiliated Route -> Accident ID</td></tr><tr><td>Include Relation</td></tr></table>		Affiliated Route -> Accident ID	Include Relation	...	<table><tr><td>Accident Cause -> Accident Consequence</td></tr><tr><td>Cause-and-Effect Relation</td></tr></table>		Accident Cause -> Accident Consequence	Cause-and-Effect Relation				
Affiliated Route -> Accident ID													
Include Relation													
Accident Cause -> Accident Consequence													
Cause-and-Effect Relation													

Figure 3. Diagram of the Relation Extraction Prompt Template. Asterisks denote anonymized identifying information.

Table 3. Definitions of Entity Relations in the Railway Accident Profile Knowledge Graph.

Head Entity	Relation	Tail Entity
Responsible Railway Bureau	Belongs To	Railway Line
Railway Line	Includes	Accident ID
Accident Cause	Leads To	Accident Outcome
Accident ID	Cause	Accident Cause
Accident ID	Outcome	Accident Outcome
Accident ID	Classification	Accident Severity
Accident ID	Responsibility Attribution	Accident Responsibility

Table 4. Structured prompt template for relation extraction.

Module	Prompt Content
Role Definition	You are an expert in railway accident analysis. Based on the given railway accident profile text and recognized entity set, determine the semantic relations between entities and output the results as structured triples.
Accident Profile Text Input	Accident profile text: {Accident_Text}
Entity Extraction Results	Recognized entities and their types: {Entity_List}. Relation inference may only use entities in this entity set.
Relation Definitions	Candidate relation types: (1) Belongs To: administrative affiliation between railway line and responsible railway bureau; (2) Includes: inclusion relation between railway line and accident ID; (3) Leads To: causal relation between accident cause and accident outcome; (4) Cause: correspondence between accident ID and accident cause; (5) Outcome: correspondence between accident ID and accident outcome; (6) Classification: determination relation between accident ID and accident severity; (7) Responsibility Attribution: determination relation between accident ID and accident responsibility.
Candidate Entity Constraint	Only entities in the recognized entity set may be used; entities not present in the text must not be added, rewritten, or inferred.
Relation Type Constraint	The relation type must be selected from the seven predefined relation types above. No new relation category may be generated. A triple is output only when the accident text supports the corresponding relation.
Output Constraint	Each triple is output only once. Relations not supported by the source text must not be added from general knowledge. If no relation can be determined between entities, no triple is output.
Output Format	Uniformly output as: (head entity, relation type, tail entity). Return only the triple list without additional explanation.

The prompt template constrains relation extraction through candidate-entity constraints, relation-type constraints, and output-format constraints. Candidate-entity constraints restrict relation inference to the entities identified by RCBC; relation-type constraints require the model to select only from the seven predefined relations; and output constraints

specify the triple format, deduplication rule, and conditions under which a relation may be returned. These constraints reduce uncertainty from open-ended generation and allow the extracted triples to be used directly in subsequent knowledge graph construction.

Because large language model outputs may be affected by changes in prompt expression, the task description, organization of the candidate entity set, relation definitions, and output format are fixed during relation extraction to control additional variation introduced by prompt formulation. It should be noted that this study focuses on the constraining role of structured prompts in domain-specific relation extraction and does not further conduct a systematic sensitivity analysis under different prompt wording, numbers of examples, or example orders. Therefore, the reported results correspond to the current fixed prompt-template setting.

Railway accident profile texts also contain domain-specific terms such as shunting routes and switch locking. A general-purpose large language model may therefore still misinterpret railway-specific semantics. To reduce the influence of domain-semantic bias on relation extraction, this study constrains the model's relation judgment by providing the original accident text and limiting the candidate entity set and predefined relation types.

2.4. Entity Completion Method

During entity extraction from railway accident profile texts, core entities such as accident causes and accident outcomes are often embedded in accident process segments with complex semantic logic. Differences in textual expression may cause incomplete entity-boundary recognition or entity omissions, thereby affecting knowledge graph completeness. To provide candidate completions for such missing entities, this study designs the GA-LLM model shown in Figure 4.

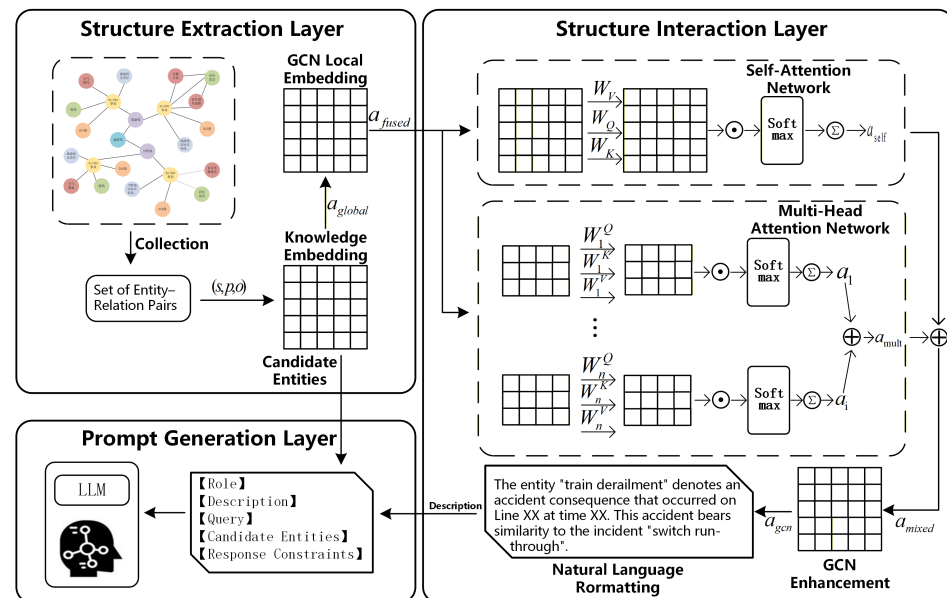


Figure 4. Structure of the GA-LLM model. Ellipses indicate additional elements of the same type.

GA-LLM consists of a structure extraction layer, a structure interaction layer, and a prompt generation layer. The structure extraction layer uses knowledge embeddings to represent head entities, relations, and candidate entities and to form a candidate entity set. The structure interaction layer combines attention mechanisms with GCN to aggregate entity-neighborhood and relation information. The prompt generation layer organizes the structural information as natural-language prompts and constrains the large language

model to select a completion entity from the candidate set. The three modules perform candidate generation, structural information fusion, and constrained completion, respectively.

(1) Structure Extraction Layer

The structure extraction layer is designed to generate entity and relation embedding representations that integrate global semantic features with local structural information. The data, after entity and relation extraction, is organized in the form of triples, where each triple is represented as $(s, p, o) \in T$, with s representing the head entity, p representing the relation, and o representing the tail entity.

This study uses a Knowledge Graph Embedding (KGE) model to learn vector representations of entities and relations. The model is pretrained on the existing knowledge graph to obtain low-dimensional semantic representations. For an incomplete triple, matching scores are calculated for potential completion entities according to the known head or tail entity and its associated relation, and entities whose scores exceed a preset threshold are selected to form the candidate set. Because SimKGC can represent semantic associations between entities and relations and can be used for candidate matching in incomplete triples, SimKGC is used for knowledge graph embedding. The resulting global embedding is calculated as shown in Equation (2).

$$a_{global} = \text{KGE}(s, p, o) \quad (2)$$

Here, KGE refers to the SimKGC embedding model. s and o denote the vector embeddings of the head and tail entities, respectively, while p denotes the embedding vector of the relation.

To focus on the subgraph structure of an entity's neighborhood, this study introduces a GCN (Graph Convolutional Network) to capture the surrounding information of entities and uses a weighted summation to obtain the fused vector a_{fused} , which is calculated as shown in Equation (3).

$$a_{fused} = \alpha a_{global} + (1 - \alpha) \text{ReLU}(\tilde{A} W_g a_{global}) \quad (3)$$

Here, α denotes the fusion weight, a_{global} represents the global embedding, ReLU is the activation function, $\tilde{A}$ denotes the normalized adjacency matrix, and W_g is the weight matrix of the GCN.

In addition, SimKGC is also employed to construct the candidate entity set. By applying a similarity threshold, entities that meet the criterion are selected as candidates to support the design of prompt strategies and effectively constrain the generation space of the large language model. The matching scores of candidate entities are measured using cosine similarity, as shown in Equation (4).

$$\theta(s, p, o) = \cos(a_{sp}, a_o) \quad (4)$$

Here, $\theta(s, p, o)$ denotes the score of the triple (s, p, o) , a_{sp} represents the embedding vector obtained by learning the representations of the head entity s and the relation p , and a_o represents the embedding vector of the tail entity.

(2) Structure Interaction Layer

The structure interaction layer combines self-attention and multi-head attention to perform weighted interaction over entity and relation representations and model associations between different nodes. For nodes i and j , their embedding vectors are a_i and

a_j , respectively. The attention weight between the two nodes is calculated as shown in Equation (5).

$$a_{ij} = \text{softmax}\left(\frac{(a_i W_Q)(a_j W_K)^T}{\sqrt{d}}\right) \quad (5)$$

Here, d denotes the embedding dimension, while W_Q and W_K represent the query and key matrices, respectively. The softmax function normalizes the attention weights into a probability distribution so that the attention weights of node i over its neighboring nodes sum to 1.

In the self-attention mechanism, the vector a_{self} represents the weighted aggregation of a node's neighboring feature vectors, where a_{ij} denotes the attention weight between nodes. Its computation is shown in Equation (6).

$$a_{self} = \sum_j a_{ij}(a_j W_V) \quad (6)$$

Here, the matrix W_V denotes a projection transformation that maps the original embeddings into a task-specific feature space, thereby enhancing the semantic adaptability of the representations.

Multi-head attention computes multiple attention heads in parallel and combines entity and relation features from different projection spaces. The i -th attention head is calculated as shown in Equation (7).

$$head_i = \text{softmax}\left(\frac{(a_r W_i^Q)(a_r W_i^K)^T}{\sqrt{d}}\right) a_r W_i^V \quad (7)$$

Here, a_r denotes the input features, and W_i^Q , W_i^K , and W_i^V represent the projection matrices of the i -th attention head.

Finally, the outputs of all attention heads are integrated via vector concatenation followed by a linear transformation to produce a unified multi-head attention representation a_{multi} , as computed in Equation (8).

$$a_{multi} = \text{Concat}(a_1, a_2, \dots, a_n) W^O \quad (8)$$

Here, Concat denotes the vector concatenation operation, n indicates the number of attention heads, and W^O denotes the output fusion matrix.

Upon completing the weighted integration of the self-attention and multi-head attention mechanisms, the fused attention representation a_{mixed} is obtained, as computed in Equation (9).

$$a_{mixed} = \gamma a_{self} + (1 - \gamma) a_{multi} \quad (9)$$

Here, γ denotes the fusion weight, a_{self} represents the self-attention representation, and a_{multi} represents the multi-head attention representation.

A GCN layer is then added to the attention representation to aggregate neighboring-node information and supplement local graph-structural features, as shown in Equation (10).

$$a_{gcn} = \text{ReLU}\left(\tilde{A} a_{mixed} W_g\right) \quad (10)$$

Here, ReLU denotes the activation function, W_g represents the GCN weight matrix, and $\tilde{A}$ denotes the normalized adjacency matrix.

After attention and GCN processing, a structural representation integrating entity attributes, relation types, and neighboring-node information is obtained. A predefined

template then maps this structured information into a natural-language description that is used as supplementary context during prompt generation. The description is generated as shown in Equation (11).

$$Des = T(\text{Name}(a_i), \text{Join}(A), \text{Join}(R)) \quad (11)$$

Here, a_i denotes a known entity, and $\text{Name}(a_i)$ denotes its name. $\text{Join}(A)$ represents the operation of combining the attribute information in set A , while $\text{Join}(R)$ represents the combination of the relation types in set R .

(3) Prompt Generation Layer

The prompt generation layer aims to convert structured knowledge graph information into natural language expressions, constructing guiding prompts to support large language models (LLMs) in performing knowledge completion tasks.

This study uses a structure-driven prompt template containing five components: role definition, entity description, question formulation, candidate entities, and answer format. The role definition constrains the task scope; the entity description provides the known entity and its local structural information; question formulation presents the missing content as an incomplete triple; candidate entities restrict the selectable answer space; and the answer format standardizes model output for subsequent parsing and verification.

The structure-driven prompt used for graph completion is specified as follows. The role definition is: "You are an expert in railway-accident knowledge graph completion. Based on the known entity and relation, complete the incomplete triple and select the most reasonable answer from the candidate entity set." The entity-description component provides the head entity and its type, the relation and its definition, local structural information centered on the head entity, and entities semantically similar to the head entity. The question component uses: "Given the incomplete triple: {head entity}-{relation}-{entity to complete}. Which tail entity most likely corresponds to {entity to complete}?" The candidate component provides the candidate tail-entity set. The answer must be semantically consistent with "{head entity}-{relation}", match the relation pattern reflected by the local structure, and must not contain an entity outside the candidate set. The output format is standardized as "Answer: {candidate entity name}." These candidate constraints limit the completion scope to the candidate set and reduce uncertainty from open-ended generation.

2.5. Evaluation Metrics

To evaluate RCBC entity extraction performance on railway accident profile texts, this study uses micro-averaging and adopts Micro Precision (MicroP), Micro Recall (MicroR), and MicroF as evaluation metrics. MicroP is the proportion of predicted positive samples that are true positives and reflects prediction precision, as shown in Equation (12).

$$\text{MicroP} = \frac{TP}{TP + FP} \times 100\% \quad (12)$$

Here, TP denotes the number of samples correctly predicted as a given class, and FP denotes the number of samples from other classes incorrectly predicted as that class.

MicroR measures the proportion of actual positive samples that are correctly identified and reflects coverage of true samples, as shown in Equation (13).

$$\text{MicroR} = \frac{TP}{TP + FN} \times 100\% \quad (13)$$

Here, FN denotes the number of samples that belong to the class and are incorrectly predicted as another class.

MicroF is the harmonic mean of MicroP and MicroR and balances precision and recall, as shown in Equation (14).

$$\text{MicroF} = \frac{2 \times \text{MicroP} \times \text{MicroR}}{\text{MicroP} + \text{MicroR}} \times 100\% \quad (14)$$

To evaluate GA-LLM knowledge graph completion, Hits@k and Mean Reciprocal Rank (MRR) are used as the main metrics. Hits@k measures whether the correct entity appears among the top- k candidate entities, as shown in Equation (15).

$$\text{Hits@k} = \frac{1}{N} \sum_{i=1}^N I(r_i \leq k) \quad (15)$$

Here, N denotes the total number of test samples, r_i denotes the rank of the correct entity for the i -th test sample, and I is an indicator function that equals 1 when the condition is satisfied and 0 otherwise.

MRR measures the overall ranking quality of the correct entities, as shown in Equation (16).

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^N \frac{1}{r_i} \quad (16)$$

A higher MRR indicates that the correct entities tend to be ranked closer to the top of the candidate list.

For the question-answering function, this study uses the RAGAS evaluation framework [19]. Faithfulness measures the consistency between the generated answer and the retrieved context; semantic similarity measures the semantic proximity between the generated answer and the reference answer; context precision evaluates the relevance of the retrieved context to the question; and context recall evaluates the coverage of information required to answer the question.

3. Results

This study takes railway accident profile texts as the core data source and utilizes the augmented text as a corpus to conduct a series of experiments, including entity extraction, relation extraction, and knowledge graph completion. A domain-specific knowledge graph is then constructed to evaluate the effectiveness of the proposed approach.

3.1. Analysis of Entity Extraction Experimental Results

In the entity extraction task, RCBC uses RoBERTa as the backbone and integrates CNN, BiLSTM, and CRF modules. The maximum sequence length is set to 512, and the model is trained for 30 epochs. The RoBERTa learning rate is 3×10^{-5} , while the CNN, BiLSTM, and CRF learning rates are 1.5×10^{-4} . The Adam optimizer is used for parameter updates. The numbers of entities extracted by RCBC are shown in Table 5.

To further analyze the roles of key RCBC components, ablation results are shown in Figure 5. The comparison examines the effects of the CNN and BiLSTM modules on entity extraction performance. With RoBERTa and CRF kept unchanged, the complete RCBC model achieves higher MicroP, MicroR, and MicroF values than both RoBERTa-BiLSTM-CRF and RoBERTa-CNN-CRF. The results indicate that CNN and BiLSTM provide complementary contributions through local compositional features and contextual dependencies, respectively.

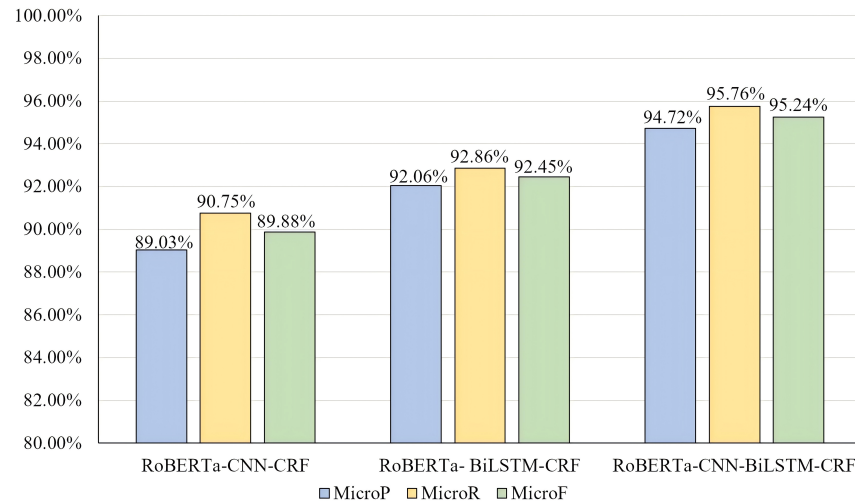


Figure 5. RCBC model ablation results.

Table 5. Summary of extracted entity counts.

Entity Type	Count
Responsible Railway Bureau	8
Railway Line	52
Accident ID	268
Accident Cause	352
Accident Outcome	274
Accident Severity	247
Accident Responsibility	292

Using the RCBC model, a total of 1493 entities were extracted, comprising 8 responsible railway bureau entities, 52 railway line entities, 268 accident ID entities, 352 accident cause entities, 274 accident outcome entities, 247 accident severity entities, and 292 accident responsibility entities.

To comprehensively evaluate model performance, MicroP, MicroR, and MicroF are used as the main evaluation metrics. After RCBC is trained on the augmented text data, the entity recognition results are shown in Table 6.

Table 6. Entity extraction results on the railway accident profile dataset.

Entity Type	MicroP	MicroR	MicroF
Responsible Railway Bureau	90.52%	89.90%	90.21%
Railway Line	91.35%	90.87%	91.11%
Accident ID	91.76%	91.32%	91.54%
Accident Cause	87.01%	86.45%	86.73%
Accident Outcome	86.77%	87.00%	86.50%
Accident Severity	90.96%	90.50%	90.73%
Accident Responsibility	90.10%	89.60%	89.85%

The results show that RCBC performs well on the current dataset, with MicroF exceeding 90% for most entity categories. Accident Cause and Accident Outcome have MicroF values of 86.73% and 86.50%, respectively, which are lower than those of the other categories. These descriptive entities generally occur in accident process segments, where professional terminology is denser, semantic logic is more complex, and a single accident may involve multiple contributing factors, making entity boundaries and contextual dependencies more difficult to determine. In addition, terminology usage, causal-chain length, and description

style vary across accident scenarios. Even after EDA, the current corpus cannot cover all complex expressions, and missed entities or incomplete boundary recognition may still occur. In contrast, responsible railway bureau, railway line, accident ID, and accident severity entities have more standardized expressions and clearer boundaries.

3.2. Analysis of Knowledge Graph Completion Experimental Results

To further evaluate GA-LLM on railway accident profile knowledge graph completion, this study compares it with four representative methods: TransE [20], DistMult, KE-GCN [21], and LLM-Prompt. TransE and DistMult represent conventional knowledge-embedding methods, KE-GCN represents a graph-structure-enhanced method, and LLM-Prompt represents a prompt-based LLM completion method. Hits@1, Hits@5, Hits@10, and MRR are used to evaluate overall candidate-ranking performance. The comparison is shown in Table 7.

Table 7. Comparison of knowledge graph completion models for railway accident profiles.

Model	Hits@1	Hits@5	Hits@10	MRR
TransE	62.35%	75.12%	80.06%	61.29%
DistMult	65.78%	78.45%	82.91%	71.55%
KE-GCN	70.23%	82.67%	86.54%	75.82%
LLM-Prompt	72.56%	83.10%	87.02%	77.12%
GA-LLM	81.39%	89.76%	92.45%	84.64%

GA-LLM achieves the highest values on all four metrics. TransE and DistMult rely primarily on matching entity and relation embeddings and have relatively limited joint use of complex graph structure and domain semantics. KE-GCN uses neighborhood structural information but remains limited in modeling complex domain semantics. LLM-Prompt relies on large language model semantic judgment but lacks explicit graph-structural constraints. GA-LLM combines candidate-entity screening, graph-structural information, and large language model semantic judgment and therefore achieves the best overall candidate-ranking performance on the current dataset.

For the category-level graph completion experiment, the augmented text records are used to construct the graph completion dataset. Accident IDs are unique identifiers for individual accidents and are therefore excluded from completion. The GA-LLM completion results for the remaining six entity categories are shown in Table 8.

Table 8. Knowledge graph completion results for railway accident profiles.

Entity Type	Knowledge Graph Completion Metrics			
	MRR	Hits@1	Hits@3	Hits@10
Responsible Railway Bureau	79.91%	67.91%	88.55%	93.46%
Railway Line	80.16%	68.43%	88.97%	93.89%
Accident Cause	78.12%	65.91%	86.91%	92.33%
Accident Outcome	78.34%	66.23%	84.12%	92.57%
Accident Severity	66.32%	56.01%	75.48%	80.11%
Accident Responsibility	66.91%	56.42%	73.01%	80.76%

The category-level results show differences in candidate hit and ranking performance across entity types. The first four categories achieve relatively better results, whereas Accident Severity and Accident Responsibility still have greater room for improvement.

Hits@k and MRR reflect candidate-entity hit and ranking performance and do not imply that completion results have been factually verified. Although GA-LLM narrows the

completion space through candidate entity sets, graph-structural information, and relation-level semantic constraints, incorrect candidate selection may still occur. For key information such as accident causes, accident severity, and accident responsibility, completion results should be checked against original accident records and graph context and treated as candidate knowledge.

4. Discussion

4.1. Knowledge Graph Visualization Application

This study uses Neo4j as the storage and visualization platform for railway accident profile knowledge. The integrated results of entity extraction, relation extraction, and graph completion are uniformly represented as “entity–relation–entity” triples and are used to construct the railway accident knowledge graph. Graph storage and visualization are implemented in Neo4j to support complex relation modeling. Given that railway accident profile records are updated dynamically, the graph design supports incremental updates and periodic maintenance and expansion through batch processing. The resulting railway accident profile knowledge graph is shown in Figure 6.

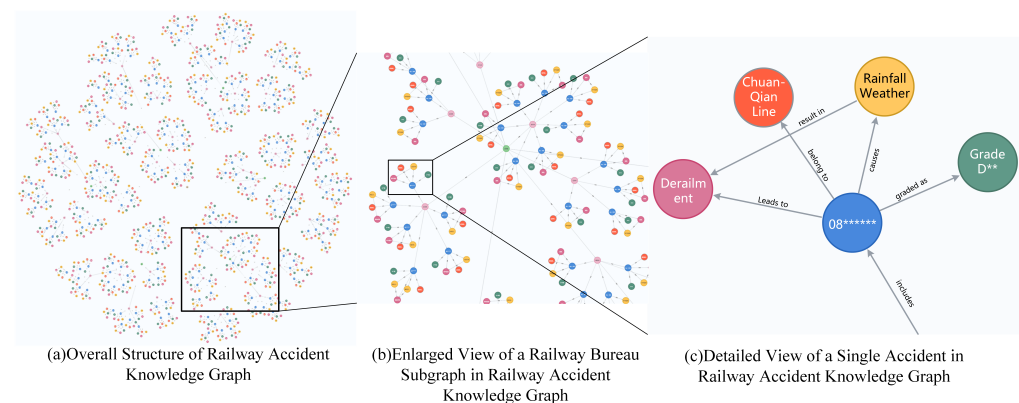


Figure 6. Schematic of the railway accident profile knowledge graph. Chinese labels show source-language entities; asterisks denote anonymized identifying information.

Figure 6a shows the overall knowledge graph network; Figure 6b provides an enlarged local view of nodes and relations associated with a railway bureau; and Figure 6c further presents entities and relations within a single accident. Different colors indicate different entity types, and edges represent semantic relations between entities.

4.2. Knowledge Graph Retrieval Application

Retrieval-Augmented Generation (RAG) combines external knowledge retrieval with large language model generation by retrieving information related to a question before answer generation and providing the retrieved information as context for the model [22,23]. GraphRAG further incorporates entity relations and graph structure from knowledge graphs, allowing retrieval to use both textual semantics and structured associations [24–26]. Given the explicit entity relations and cross-accident associations in railway accident profile texts, this study applies GraphRAG to retrieval and question answering over the constructed knowledge graph.

Based on the constructed railway accident profile knowledge graph, a GraphRAG-based question-answering architecture is designed, as shown in Figure 7. The framework contains three stages: data preprocessing and text chunking, community clustering and vector-index construction, and semantic retrieval and answer generation.

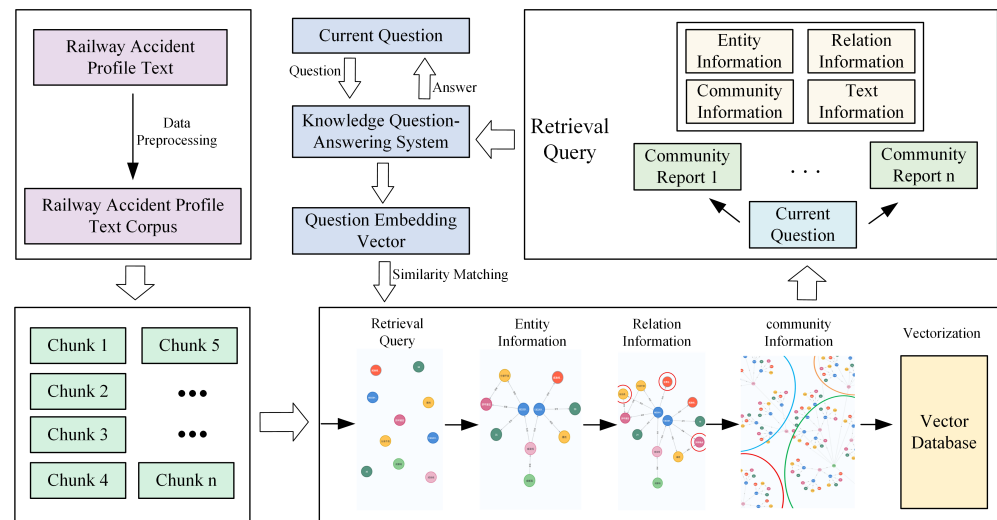


Figure 7. Architecture of the railway accident profile text retrieval function. Chinese text shows source-language examples from the railway accident corpus.

The framework first segments the preprocessed accident texts by individual accident and performs entity extraction, relation extraction, and graph completion. The Leiden community detection algorithm [27] is then used to cluster the knowledge graph and derive thematic subgraphs and community summaries. A graph-text fusion embedding mechanism is used to construct a vector index [28]. During querying, the user question is semantically parsed and vectorized; relevant text chunks, thematic subgraphs, and community summaries are retrieved from the index; and the retrieved results and original question are jointly provided to the large language model for answer generation.

4.3. Qualitative Analysis of Railway Accident Profile Question-Answering Results

To illustrate the practical output of GraphRAG for composite questions, this study selects statistical and association-analysis scenarios involving multiple accident records, including accident-cause statistics, accident-outcome analysis, railway-line distribution, and accident-severity statistics. Examples are shown in Table 9. To satisfy data-security requirements, identifying information such as railway bureaus and railway lines is anonymized, and the answers are presented in summarized form.

Table 9. Examples of composite railway accident profile question-answering results.

Question Type	Example Question	Answer Result
Accident Cause Statistics	What are the most common causes of accidents recorded in ** Bureau?	Common causes of railway accidents in ** Bureau mainly include continuous rainfall, track-circuit shunting failure, rule violations, and improper handling by operating personnel.
Accident Outcome Analysis	What are the common accident outcomes in ** Bureau?	Common accident outcomes in ** Bureau mainly include vehicle derailment, switch damage, traffic interruption, and equipment damage.

Table 9. *Cont.*

Question Type	Example Question	Answer Result
Railway-Line Distribution	In the current accident records of ** Bureau, which lines have more recorded accidents?	According to the current accident records, accidents in ** Bureau are mainly distributed on ** Line, ** Line, ** Line, and ** Line.
Accident Severity Statistics	What is the distribution of accident severity in the current accident records of ** Bureau?	According to the current records, D10 accidents occur most frequently in ** Bureau, followed by D3 accidents.

Note: ** denotes anonymized identifying information.

These examples show that the question-answering function can jointly retrieve and summarize accident causes, accident outcomes, railway-line distribution, and accident severity across multiple records, thereby illustrating how GraphRAG supports composite queries.

4.4. Quantitative Analysis of Railway Accident Profile Question-Answering Results

To further evaluate the question-answering function, the four metrics defined above are used to compare LLM, LLM-RAG, and LLM-GraphRAG on both basic and composite questions. The results are shown in Table 10.

Table 10. Performance comparison of question-answering methods on basic and composite questions.

Question Type	Evaluation Metric	LLM	LLM-RAG	LLM-GraphRAG
Basic	Faithfulness	71.42%	78.15%	82.36%
Basic	Semantic Similarity	73.08%	79.64%	83.02%
Basic	Context Precision	72.31%	77.58%	81.67%
Basic	Context Recall	74.25%	81.12%	87.84%
Composite	Faithfulness	67.85%	73.94%	79.68%
Composite	Semantic Similarity	70.12%	76.08%	78.43%
Composite	Context Precision	68.47%	74.36%	76.92%
Composite	Context Recall	71.35%	78.64%	82.57%

LLM-GraphRAG obtains higher faithfulness, semantic similarity, context precision, and context recall than both LLM and LLM-RAG for basic and composite questions, while the overall scores for composite questions are lower than those for basic questions. These results indicate that, under the current end-to-end evaluation setting, LLM-GraphRAG achieves higher answer-quality and retrieved-context scores than LLM and LLM-RAG.

No dedicated experiments were conducted to assess retrieval latency or query-perturbation robustness for the retriever alone. Therefore, this study does not make further inferences about GraphRAG's real-time response performance or stability under perturbation.

5. Conclusions

This study develops a knowledge graph construction workflow for the structured processing and knowledge modeling of railway accident profile texts, including data augmentation, entity extraction, relation extraction, and graph completion. RCBC is used to identify seven core entity types, an LLM is used for entity-relation extraction under fixed prompt conditions, and GA-LLM is used for the candidate completion of missing entities. Experimental results show that the MicroF values of RCBC exceed 85% across the seven entity extraction tasks and that GA-LLM achieves an average Hits@3 of 82.84% across six

entity completion categories. These results indicate that the workflow can support core entity recognition and candidate knowledge completion under the current dataset and provide a basis for constructing railway accident profile knowledge graphs.

In addition, this study develops an LLM-GraphRAG-based question-answering function for railway accident profiles, enabling semantic retrieval and answer generation over accident knowledge. Under the current test settings for both basic and composite questions, LLM-GraphRAG achieves higher faithfulness, semantic similarity, context precision, and context recall than both LLM and LLM-RAG. These results suggest that incorporating graph-structured knowledge can improve retrieved-context relevance and information coverage under the current test setting, thereby providing more complete context for answer generation.

Overall, this study establishes a knowledge graph construction and retrieval method for railway accident profile texts, supporting the structured organization of accident information, entity-relation association, and auxiliary historical accident retrieval. Through RCBC entity extraction, prompt-based relation extraction, GA-LLM knowledge completion, and LLM-GraphRAG question answering, the study provides a feasible approach to knowledge modeling and information utilization for railway accident texts. In practical applications, relation-extraction, knowledge-completion, and question-answering outputs produced using LLMs should be verified against the original accident records, especially when they involve accident causes, accident severity, or accident responsibility. Such outputs should not be used as the sole basis for accident classification, responsibility determination, or safety decision-making.

Railway accidents are generally low-frequency events, and major accidents and some rare accident types occur even less frequently, resulting in an objectively limited number of real historical samples. At the same time, industry data-security requirements, confidentiality requirements, and authorization constraints limit the scale of data aggregation and sharing. Therefore, the experiments in this study are mainly validated on the currently available data. The current results show that the constructed knowledge graph and retrieval framework can support the structured organization of railway accident information, historical accident retrieval, and accident review, providing technical support for railway accident knowledge management and information utilization.

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

Funding: This research was funded by the National Natural Science Foundation of China (NSFC)—China State Railway Group Joint Fund for Basic Research, grant number U2268202; the Technology Research and Development Plan Program of China State Railway Group Co., Ltd., grant number K2024S005(JB); and the Foundation of China Academy of Railway Sciences Co., Ltd., grant number 2024YJ259. The Article Processing Charge (APC) was funded by the Technology Research and Development Plan Program of China State Railway Group Co., Ltd.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of these data. The data used in this study were obtained from the Institute of Computing Technologies, China Academy of Railway Sciences Corporation Limited. The data are not publicly available due to confidentiality and data-use restrictions and cannot be shared with third parties.

Conflicts of Interest: Authors Zhibo Cheng and Yanhua Wu were employed by the company China Academy of Railway Sciences Corporation Limited. The authors declare that this study received funding from China State Railway Group Co., Ltd. and China Academy of Railway Sciences 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:

NER	Named Entity Recognition
RCBC	RoBERTa-CNN-BiLSTM-CRF
LLM	Large Language Model
RAG	Retrieval-Augmented Generation
GraphRAG	Graph Retrieval-Augmented Generation
EDA	Easy Data Augmentation
GCN	Graph Convolutional Network
KGE	Knowledge Graph Embedding
MRR	Mean Reciprocal Rank
GA-LLM	GCN-Attention-LLM

References

1. Yin, F.; Mei, S. The “14th Five-Year Plan” for scientific and technological innovation in the transport sector was released. *J. Waterw. Harb.* **2022**, *43*, 239.
2. Ren, C.; Zhang, B. Research on railway network data security risk assessment methods. *Railw. Signal. Commun.* **2025**, *61*, 45–50. [[CrossRef](#)]
3. Zhang, Y.; Fu, J.; Li, Z.; Meng, S.; Wu, Z.; Xie, D. Causation network model and prevention measures for human-factor safety accidents in railway maintenance operations. *J. Railw. Sci. Eng.* **2025**, *22*, 5119–5131. [[CrossRef](#)]
4. Dai, C.; Liu, J.; Li, X.; Zhang, Y. Research on human factors analysis methods for accidents in railway locomotive system. *Railw. Transp. Econ.* **2025**, *47*, 205–214. [[CrossRef](#)]
5. Xu, Y. Analysis of Rule-Violating Behaviors in Accidents in the Railway Traffic Service System Based on HFACS-Bayesian Network. Master’s Thesis, China Academy of Railway Sciences, Beijing, China, 2023.
6. Liu, J.; Schmid, F.; Li, K.; Zheng, W. A knowledge graph-based approach for exploring railway operational accidents. *Reliab. Eng. Syst. Saf.* **2021**, *207*, 107352. [[CrossRef](#)]
7. Hughes, P.; Robinson, R.; Figueres-Esteban, M.; Van Gulijk, C. Extracting safety information from multi-lingual accident reports using an ontology-based approach. *Saf. Sci.* **2019**, *118*, 288–297. [[CrossRef](#)]
8. Tian, L.; Zhang, J.; Zhang, J.H.; Zhou, W.; Zhou, X. Knowledge graph survey: Representation, construction, reasoning and knowledge hypergraph theory. *J. Comput. Appl.* **2021**, *41*, 2161–2186. [[CrossRef](#)]
9. Zhong, L.; Wu, J.; Li, Q.; Peng, H.; Wu, X. A comprehensive survey on automatic knowledge graph construction. *ACM Comput. Surv.* **2024**, *56*, 94. [[CrossRef](#)]
10. Lin, H.; Zhao, Z.; Lu, R.; Bai, W.; Hu, N.; Lu, R. Construction and application of a knowledge graph for fault handling of high-speed railway turnout equipment. *J. China Railw. Soc.* **2024**, *46*, 73–80. [[CrossRef](#)]
11. Li, X.; Liu, C.; Dai, M.; Li, G.; Shi, W. Construction and application of knowledge graph for railway transportation equipment faults. *Railw. Comput. Appl.* **2023**, *32*, 20–24.
12. Kong, Q.; Miao, Z.; Feng, X.; Yang, G.; Long, X. Research on the closed-loop management scheme of high-speed railway passenger transport dispatching commands based on knowledge graph. *Railw. Transp. Econ.* **2023**, *45*, 89–98. [[CrossRef](#)]
13. Zhu, Q.; Wang, S.; Ding, Y.; Zeng, H.; Zhang, L.; Guo, Y.; Li, H.; Wang, W.; Song, S.; Hao, R.; et al. A method of safety-quality-schedule knowledge graph for intelligent management of drilling and blasting construction of railway tunnels. *Geomat. Inf. Sci. Wuhan Univ.* **2022**, *47*, 1155–1164. [[CrossRef](#)]
14. Song, W.; Yao, H.; Zhong, J.; Xiang, Z.; Xie, H.; Bai, H.; Wu, J. Research and application of key technologies for intelligent decision-making of engineering schemes for critical structures driven by knowledge graph. *Railw. Stand. Des.* **2025**, *69*, 73–82.
15. Pu, T.; Tan, Y.; Peng, G.; Xu, H.; Zhang, Z. Construction and application of knowledge graph in the electric power field. *Power Syst. Technol.* **2021**, *45*, 2080–2091. [[CrossRef](#)]
16. Pellicer, L.F.A.O.; Ferreira, T.M.; Costa, A.H.R. Data augmentation techniques in natural language processing. *Appl. Soft Comput.* **2023**, *132*, 109803. [[CrossRef](#)]

17. Wei, J.; Zou, K. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 6382–6388. [[CrossRef](#)]
18. Wadhwa, S.; Amir, S.; Wallace, B.C. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada, 9–14 July 2023; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 15566–15589. [[CrossRef](#)] [[PubMed](#)]
19. Es, S.; James, J.; Espinosa Anke, L.; Schockaert, S. RAGAS: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, St. Julians, Malta, 17–22 March 2024; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 150–158. [[CrossRef](#)]
20. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2787–2795.
21. Yu, D.; Yang, Y.; Zhang, R.; Wu, Y. Knowledge embedding based graph convolutional network. In *Proceedings of the Web Conference 2021, Virtual Event/Ljubljana, Slovenia, 19–23 April 2021*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 1619–1628. [[CrossRef](#)]
22. Dong, Y.; Wang, S.; Zheng, H.; Chen, J.; Zhang, Z.; Wang, C. Advanced RAG models with graph structures: Optimizing complex knowledge reasoning and text generation. In *Proceedings of the 2024 5th International Symposium on Computer Engineering and Intelligent Communications (ISCEIC)*; IEEE: Piscataway, NJ, USA, 2024; pp. 626–630. [[CrossRef](#)]
23. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.T.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.
24. Peng, B.; Zhu, Y.; Liu, Y.; Bo, X.; Shi, H.; Hong, C.; Zhang, Y.; Tang, S. Graph retrieval-augmented generation: A survey. *ACM Trans. Inf. Syst.* **2026**, *44*, 35:1–35:52. [[CrossRef](#)]
25. Hu, Y.; Lei, Z.; Zhang, Z.; Pan, B.; Ling, C.; Zhao, L. GRAG: Graph retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, NM, USA, 29 April–4 May 2025*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 4145–4157. [[CrossRef](#)]
26. Wu, J.; Zhu, J.; Qi, Y.; Chen, J.; Xu, M.; Menolascina, F.; Jin, Y.; Grau, V. Medical Graph RAG: Evidence-based medical large language model via graph retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vienna, Austria, 27 July–1 August 2025; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 28443–28467. [[CrossRef](#)]
27. Traag, V.A.; Waltman, L.; van Eck, N.J. From Louvain to Leiden: Guaranteeing well-connected communities. *Sci. Rep.* **2019**, *9*, 5233. [[CrossRef](#)] [[PubMed](#)]
28. Sarmah, B.; Mehta, D.; Hall, B.; Rao, R.; Patel, S.; Pasquali, S. HybridRAG: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. In *Proceedings of the 5th ACM International Conference on AI in Finance*, Brooklyn, NY, USA, 14–17 November 2024; Association for Computing Machinery: New York, NY, USA, 2024; pp. 608–616. [[CrossRef](#)]

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.