symmetry

MDPI

Article

Utilizing Large Language Models for Hyper Knowledge Graph Construction in Mine Hoist Fault Analysis

Xiaoling Shu, Xiaochao Dang, Xiaohui Dong and Fenfang Li *

College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China; 2022222219@nwnu.edu.cn (X.S.); dangxc@nwnu.edu.cn (X.D.); dxh@nwnu.edu.cn (X.D.)
* Correspondence: lifenfang@nwnu.edu.cn

Abstract: The rapid development of artificial intelligence technology is driving the intelligentization process across various fields, particularly in knowledge graph construction, where significant achievements have been made. However, research on hyper-relational knowledge graphs in the industrial domain remains relatively weak. Traditional construction methods suffer from low automation, high cost, and poor reproducibility and portability. To address these challenges, this paper proposes an optimized construction process for a hyper-relational knowledge graph for mine hoist faults based on large language models. This process leverages the strengths of large language models and the logical connections of fault knowledge, employing GPT's powerful reasoning abilities. A combined strategy of template-based and template-free prompts is designed to generate fault entities and relationships. To address potential data incompleteness caused by prompt engineering, link prediction is used to optimize the initial data generated by GPT o1-preview. We integrated the graph's topological structure with domain-specific logical rules and applied the Variational EM algorithm for alternating optimization while also incorporating text embeddings to comprehensively enhance data optimization. Experimental results show that compared to the unoptimized MHSD, the optimized MHSD achieved a 0.008 improvement in MRR. Additionally, compared to the latest KICGPT, the optimized MHSD showed a 0.002 improvement in MRR. Finally, the optimized data were successfully imported into Neo4j for visualization.

Keywords: industrial intelligence; hyper-relational knowledge graphs; prompt engineering design; link prediction

check for updates

Citation: Shu, X.; Dang, X.; Dong, X.; Li, F. Utilizing Large Language Models for Hyper Knowledge Graph Construction in Mine Hoist Fault Analysis. *Symmetry* **2024**, *16*, 1600. https://doi.org/10.3390/sym16121600

Academic Editor: Shi Cheng

Received: 22 October 2024
Revised: 11 November 2024
Accepted: 13 November 2024
Published: 30 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

In modern mining operations, the mine hoist system plays a crucial role as it is one of the primary transportation mechanisms. It is responsible not only for lifting ore and personnel from underground mines to the surface but also for handling the transportation of materials and equipment throughout the mining process. The stability and smooth operation of the hoist system are critical to the continuity and efficiency of mining production, making it a central component of mining operations. However, the hoist system operates in complex and dynamic environments, facing numerous challenges [1]. First, the system is required to run continuously under high-load conditions, which can easily lead to equipment fatigue and wear. Additionally, the harsh conditions of the mine, including humidity, temperature variations, and dust, accelerate equipment aging and increase the likelihood of system failures. Moreover, the mine hoist system is structurally complex, involving mechanical, electrical, and hydraulic subsystems, where a failure in any component can cause the entire system to halt. Due to these factors, various types of failures frequently occur in mine hoist systems. For example, hoisting ropes may break after prolonged use; electrical systems can fail due to overload or short circuits; and mechanical components may malfunction due to wear or insufficient lubrication. Such failures not only interrupt

production but can also lead to safety hazards, posing significant risks to both mine efficiency and personnel safety. Given this context, the rapid diagnosis and repair of hoist system failures have become critical issues in mining production [2]. Traditional diagnostic methods, which mainly rely on human expertise and periodic maintenance, are limited in their ability to handle the increasing complexity and operational demands of modern hoist systems. Therefore, there is an urgent need for more efficient and precise diagnostic and repair techniques to ensure the stable operation of the hoist system. As a result, effective monitoring, diagnosis, and prevention of mine hoist failures hold great practical significance [3].

Traditional fault diagnosis methods have primarily relied on expert experience and rule-based systems, which, while effective in certain applications, face considerable limitations. These methods often encounter challenges in acquiring and updating specialized knowledge, which can be both time-consuming and difficult to adapt as systems or technologies evolve [4,5]. Additionally, handling complex or novel fault patterns is a significant obstacle for rule-based systems due to their limited flexibility and lack of adaptive learning capabilities. With the advancement of big data and artificial intelligence technologies, knowledge graphs have emerged as a powerful tool for representing knowledge in a more dynamic and interconnected way. Unlike traditional methods, knowledge graphs efficiently organize and integrate complex heterogeneous data from various structured and unstructured sources, allowing for a more comprehensive and scalable approach to fault diagnosis [6,7]. By capturing multi-dimensional relationships among fault components, knowledge graphs facilitate a deeper understanding of fault patterns and enable continuous improvement in diagnostic models. This innovative approach provides a foundation for automated, data-driven fault diagnosis solutions that are better equipped to handle the complexities of modern industrial systems. However, conventional knowledge graphs typically express only simple one-to-one or one-to-many relationships between entities, which fall short with respect to meeting the demands of complex industrial systems. To address this, hyper-relational knowledge graphs have been developed. These graphs not only capture multiple relationships between entities but also store attribute information associated with these relationships, enabling multi-dimensional knowledge association. Hyper-relational knowledge graphs allow for fine-grained descriptions of each entity and its corresponding relationships, for instance, recording the cause, timing, frequency of faults, and their impact on other subsystems. This combination of multi-dimensional information enhances the accuracy and comprehensiveness of knowledge graphs in fault diagnosis. Large language models (LLMs), such as GPT, have achieved groundbreaking progress in natural language processing, possessing powerful capabilities in language understanding and generation [8]. Leveraging LLMs like GPT enables the automatic extraction of knowledge from vast amounts of unstructured textual data, significantly reducing the cost associated with manually constructing knowledge graphs [9,10]. However, applying LLMs to the fault diagnosis domain of mine hoisting systems presents several challenges. First, mine hoisting systems involve a vast amount of specialized terminology, such as component names, fault types, and operational standards. These domain-specific terms are often underrepresented in general language models, which can lead to misinterpretations or biases when the model attempts to process specific fault descriptions. Second, the fault patterns in mine hoisting systems are highly diverse, encompassing mechanical, electrical, and hydraulic issues, each with distinct causes, influencing factors, and multi-layered relationships. Traditional models struggle to capture and represent this complexity in fault patterns effectively. Finally, data sparsity is a significant issue in this field, as rare fault cases occur infrequently in datasets, making it difficult for models to learn and generalize well to identify and predict these uncommon faults. These challenges demand more refined optimizations and specialized designs to enable LLMs to handle the unique complexities of mine hoisting system fault analysis effectively. To address these challenges, this paper explores the application of large language models in the automated construction of hyper-relational knowledge graphs for mine hoisting system fault diagnosis. By integrating the exceptional

natural language processing capabilities of LLMs with the multi-relational representation features of hyper-relational knowledge graphs, this study achieves the automatic extraction and systematic organization of fault knowledge from complex unstructured text data, thus constructing a more comprehensive diagnostic system. This approach significantly enhances the efficiency of knowledge updating and effectively addresses the challenges of handling complex fault patterns, providing strong technical support for the stable operation of mine hoisting systems. The main contributions of this paper are as follows:

- Generation of hyper-relational data: Utilizing the powerful learning, reasoning, and generation capabilities of GPT o1-preview, a template-based prompt is employed to generate data. Enhanced by template-free prompts, a fault dataset in the form of (h, r1, t1, r2, t2, r3, t3, …) is produced;
- Completion of hyper-relational data: Link prediction is used to optimize the initial data generated by GPT, integrating the topological structure and logical rules of the knowledge graph. The EM algorithm is introduced for alternating optimization, effectively enhancing both hyper-relational and rule-based modeling, leading to the further refinement of the knowledge graph. The link prediction task is evaluated using the MRR metric;
- Knowledge graph visualization: The final optimized dataset, MHSD (optimized), is visualized using Neo4j, facilitating observation and learning.

2. Related Work

In the research on Hyper Knowledge Graph Construction for Mine Hoist Faults (HKGCF), related work primarily focuses on three main areas: hyper-relational knowledge graphs, data generation, and link prediction.

2.1. Hyper Knowledge Graph

A hyper-relational knowledge graph is an extended form of a traditional knowledge graph that can represent more complex and diverse relationships, thereby better reflecting the intricate associations in the real world. The knowledge representation format directly affects the performance of HKGC tasks. Currently, research on knowledge graphs mainly focuses on three aspects: binary relations, n-ary relations, and hyper-relations [11]. To more intuitively demonstrate the differences between these knowledge representation formats, the fault data for mine hoisting systems used in this paper are described as follows: "Crack fault located on the component hopper belt, categorized as mechanical device failure, with detailed information involving material wear or external factor influence. In the event of slippage, input should be reduced, as increased oil viscosity exacerbates system wear". Figure 1 shows different forms of knowledge representation.

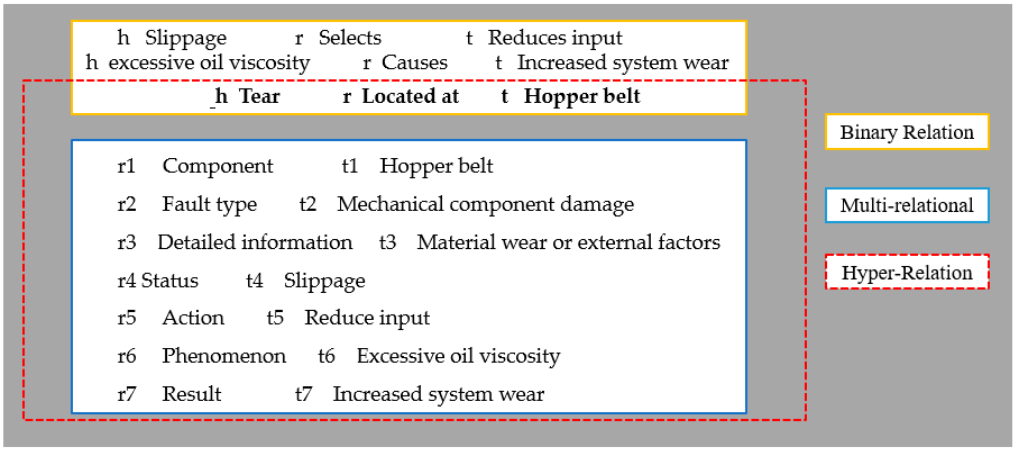


Figure 1. Different Methods of Knowledge Representation for Mine Hoist Fault Data.

Figure 1 provides a comparative analysis of binary relations, multi-relational, and hyper-relational knowledge representation formats, offering an intuitive view of the differences in information expression among these formats. Binary relations can capture only single relationships, such as "tear located at hopper belt", which, although able to convey basic fault location information, lacks a comprehensive description of the complex fault context. Multi-relational representations, by introducing additional relationship entities, can display multiple dimensions of information to some extent but remain insufficient in representing complex scenarios. In contrast, hyper-relational knowledge graphs support multi-attribute associations, enabling the integration of multi-level information such as "slippage", "excessive oil viscosity", and "reduce input" within a single framework, thereby providing a complete representation of the causes, conditions, impacts, and countermeasures of mine hoist faults. Furthermore, hyper-relational knowledge graphs exhibit significant advantages in the fine-grained association of information. They not only record fault types (such as "mechanical component damage") and influencing factors (such as "material wear or external factors") but also achieve an intuitive representation of complex knowledge through multi-dimensional relationship attributes (e.g., "condition" as "slippage", "phenomenon" as "excessive oil viscosity", and "outcome" as "increased system wear"). This multi-dimensional association enables knowledge graphs to directly infer optimization strategies in fault analysis, such as "reduce input". Therefore, hyper-relational knowledge graphs are superior to traditional methods in terms of richness in knowledge representation and precision in inferential support, providing robust technical support for the intelligent operation and maintenance of complex systems.

Effectively handling hyper-relational complex data requires a combination of multiple methods to enable efficient modeling and prediction of multi-dimensional, multi-relational structures. Traditional triple embedding methods struggle to capture the complexity of hyper-relational data. Consequently, some studies [12,13] have proposed hyper-relational knowledge graph embeddings, extending the representation to quadruples or quintuples and incorporating contextual information (such as time, location, and event type) to more comprehensively capture multi-dimensional relationships. GNNs, with their topological learning capabilities, have been widely applied to hyper-relational data processing, where they update node embeddings through message passing, effectively capturing multi-layered relationships within complex graph structures. Studies [14,15] highlight that GNNs can aggregate information from neighboring nodes, making them particularly suited for representing intricate topological relationships. MLPs, combined with attention mechanisms, offer an alternative approach to modeling relationships in multi-dimensional data [16], demonstrating significant advantages in capturing hierarchical relationship features. The Transformer model proposed by Vaswani et al. [17] uses a self-attention mechanism to achieve dynamic weighting through multi-head attention, enabling the model to focus on key features within hyper-relational data. In terms of integrating logical rules, studies [18] suggest that the variational EM algorithm can serve as a regularization term, embedding logical rules within the model to facilitate the learning of complex relationships that align with domain logic. Contrastive learning enhances the quality of model embeddings by drawing similar samples closer and pushing dissimilar samples further apart, and studies [19,20] have validated its effectiveness in processing hyper-relational data. Finally, Transformer-based methods for hyper-relational processing demonstrate superior performance in handling long-range dependencies, utilizing multi-head attention and layer normalization modules [21,22] to achieve higher precision in hyper-relational knowledge graph modeling tasks. In the field of fault knowledge graphs, Li et al. [23] introduced knowledge graphs into the alarm analysis process, using network alarm knowledge to analyze network anomalies and accurately locate the source of faults. Liang et al. [24] developed a power fault knowledge graph and employed a fault retrieval method to address the cold-start recommendation problem, predicting users' general search intentions through polymorphic search recommendation methods, thereby improving the reliability of intelligent grid operation and decision-making support. Han et al. [25] proposed a

fault knowledge graph for steel production line equipment, which enhances fault analysis accuracy and efficiency through a reinforcement learning framework and graph neural networks, providing an effective maintenance plan. Dong et al. [26] proposed a method for constructing a mine hoist fault knowledge graph using BiLSTM-CRF for entity recognition and ERNIE for relation extraction to build a triple-based fault knowledge graph, though their research had not yet delved into hyper-relational modeling in the fault domain.

2.2. Data Generation

In recent years, large language models (LLMs) with hundreds of billions of parameters or more have made significant advances in the field of natural language processing (NLP), with OpenAI's GPT series standing out for its superior performance compared to other models of the same period [27]. In September 2023, GPT-4V further expanded its capabilities in visual processing, allowing users to instruct the model to analyze images. Subsequently, Whisper V3 improved speech recognition accuracy and supported more languages. By 2024, GPT-4o had advanced the integration of audio-visual processing and natural language interaction, marking another technical evolution. In September 2024, the GPT-o1 model, also known as the "Orion Generation", was released, with its key feature being its general complex reasoning ability, which simulates human "thought processes". This is particularly evident in its application of reinforcement learning and Chain of Thought (CoT), enabling it to engage in human-like thought and reasoning before delivering answers. Compared to earlier models, GPT-o1 demonstrates significant advantages in solving complex problems. For example, it achieved an accuracy rate of 83.3% in the International Mathematical Olympiad qualification exam, far surpassing the 13.4% accuracy of GPT-4o [28]. Prompt engineering has played a crucial role in optimizing the task comprehension and output quality of large language models. By carefully designing and refining prompts, models can be guided to better understand task requirements and improve performance [29]. However, prompt engineering has its limitations, particularly when models generate illogical or inaccurate information. Continuous validation and optimization are essential to ensure the reliability of the model's outputs [30].

2.3. Link Prediction

In the task of hyper-relational knowledge graph construction, link prediction on fault-formatted data generated by large language models is essential due to the complexity and domain-specific nature of the data. Link prediction helps infer potential but unannotated relationships, thus improving the knowledge graph. In 2019, Yao et al. [31] proposed the KG-BERT model, where the textual descriptions of the head entity, relation, and tail entity are concatenated and fed into BERT. The final [CLS] token is used as the embedding for the target triple, which is then passed through a classifier to determine the validity of the triple. To enhance the representation of structured knowledge within text encoders, Li et al. [32] introduced LP-BERT in 2020, performing multi-task pre-training that incorporated masked language modeling alongside masked entity modeling and masked relation modeling. While these approaches achieved strong performance by integrating structured knowledge, their modeling was limited to the triple level and failed to capture the long-range structural semantics in knowledge graphs. In 2021, Yu et al. [33] introduced Hytransformer, which combined hypergraph structures with the Transformer model, leveraging the sequential modeling capacity of Transformers and the complex relationship representation of hypergraphs to handle multi-relational natural language processing tasks. Wang et al. [34] proposed the StAR model, dividing each triple into two asymmetric parts, similar to translation-based graph embedding methods, encoding the parts as contextual representations, and using deterministic classifiers and spatial measurement methods to learn the representations and structure. In 2023, Hu et al. [35] introduced HyperFormer, integrating hypergraph and Transformer architectures to improve modeling capabilities for complex data and relationships by utilizing advanced graph structures alongside powerful sequence modeling capabilities. In 2024, Hu et al. [36] further introduced HYPERMONO,

focusing on multi-relational and multi-entity issues within knowledge graphs and textual data, leveraging language models to enhance understanding and reasoning across complex relationships and hierarchical data. Also, in 2024, Wei et al. [37] proposed KICGPT, which combines large language models with a triple retriever. Through the use of knowledge prompt strategies, KICGPT effectively addresses the long-tail problem in knowledge graph completion without requiring additional training or fine-tuning.

3. Method

This section provides a detailed explanation of the proposed LLM-HKGCF method for constructing a knowledge graph of mine hoist faults, covering both data generation and link prediction.

3.1. Framework

This paper proposes LLM-HKGCF, with the framework illustrated in Figure 2. The method takes fault data in text form as input, and through the analysis of the text content using LLMs, it constructs an industrial fault knowledge graph. First, the GPT model's powerful few-shot learning and template-free learning capabilities are utilized. Task instructions and output examples are designed for data annotation and data augmentation to acquire the necessary dataset for this method. Next, link prediction is applied to optimize the generated data. Finally, based on the optimized data, a knowledge graph for mine hoist faults is constructed.

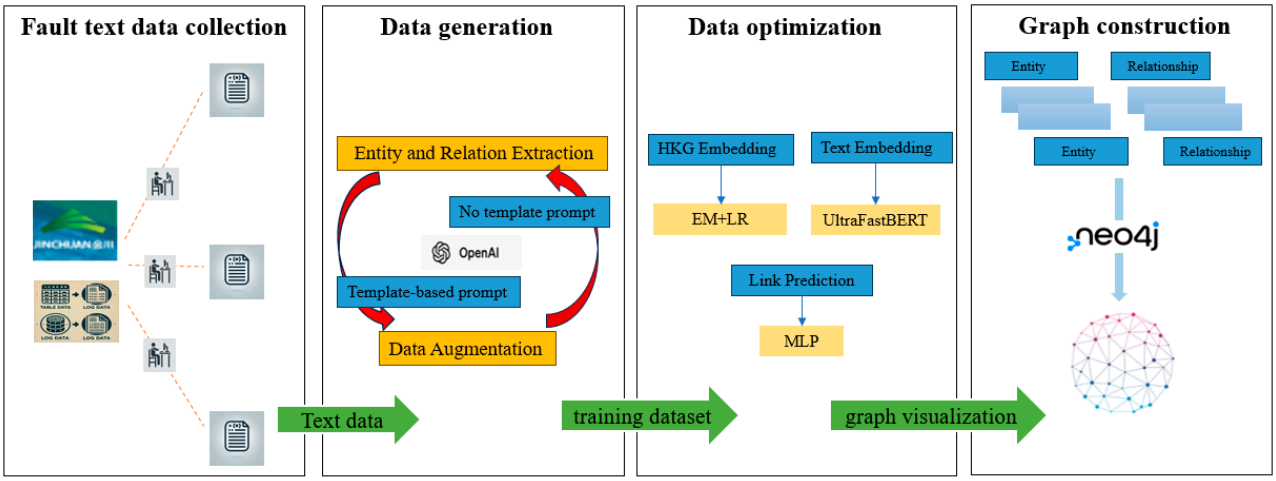


Figure 2. Flowchart for optimizing the construction of a mine hoist fault hyper-relational knowledge graph based on GPT.

3.2. Generation of Mine Hoist Fault Dataset

3.2.1. Data Acquisition

The mine hoist is a critical piece of equipment for ensuring safe production in mining operations, serving as a vital passage between the surface and underground. There are two common types of mine hoists: single-rope winding hoists and multi-rope friction hoists. The single-rope winding hoist is primarily used in shallow mine environments due to its relatively simple structure, which meets the needs of shallower operations. However, in deep mines with high annual output requirements, the multi-rope friction hoist is more commonly employed due to its ability to provide more stable and efficient lifting capacity. Figure 3 illustrates a schematic diagram of the multi-rope friction hoist and its control system from a large state-owned non-ferrous metal group.



Figure 3. Multi-rope friction hoist.

Figure 3 provides a visual representation of the structure and function of the hoist and its control system. The multi-rope friction hoist uses an electric motor as its power source and a guide wheel device as part of the transmission system. The lifting of the containers within the shaft is achieved through the friction between the steel ropes and the drum lining blocks. A braking system, consisting of disc brakes and a hydraulic station, controls the stopping of the hoist. The system also employs programmable logic controllers and industrial control computers to rapidly collect and process various parameters of the hoist. A high-speed counting board is used to control the operational speed of the hoist by calculating speed. Additionally, a main controller, monitoring encoder, analog bar display, and digital display are used to reflect the position of the hoist within the shaft. To ensure the safe operation of the equipment, electrical protection, mechanical protection, and hydraulic control systems are integrated into the design.

Inspection reports, as a key source of fault-related data, provide professional descriptions of various faults using specialized equipment terminology. The structure and language of these reports differ significantly from open-domain texts. To better select annotation corpora and build an annotation system, this paper briefly analyzes the structure and language used in inspection reports. These reports follow a highly structured format. They start with "XX Work Area XX Inspection Report", followed by a file number. The first line lists the responsible personnel (inspector, inspection system, and time), while the main body contains detailed descriptions of faults using technical terms and complex language. The report ends with reference values for equipment and systems such as armature current, voltage, and oil pressure. However, the introductory and concluding sections contain no fault-related information and serve only as an index or reference, so they are excluded from model training in this study. Only the fault descriptions are retained for annotation and experiments. Additionally, the National Standard of the People's Republic of China outlines safety requirements for mine hoists, which include equipment design and control systems [38]. Using the classifications and terminology from these standards, this paper builds its corpus. Data were collected through Python web crawlers and manual selection from inspection reports and maintenance logs in the Northwest State-owned Industrial Group. Before experimentation, the raw data underwent manual preprocessing:

- Removal of Irrelevant Text: The collected texts were initially screened to discard those with insufficient entities or texts that were too short. Regular expressions were used to remove the header and footer information from the inspection reports, keeping only the fault description sections;
- Privacy Information Desensitization: Personal names, work areas, and job-related information that were irrelevant to the task in the inspection reports and maintenance logs were removed to protect personal privacy;
- Punctuation Handling: Unnecessary punctuation marks, such as newline characters and whitespace, were removed as they negatively impact model training efficiency and could cause errors during model execution. The text was segmented into sentences using punctuation marks such as periods ("."), semicolons (";"), and exclama-

tion marks ("!"). The manually preprocessed MHSD (text) dataset was 328 KB. Table 1 shows a portion of the text.

Table 1. Mine hoist fault text data.

Record ID	Text Description
Record 1	Cracks were found in the material on the conveyor belt. After careful inspection, it was preliminarily determined that this might be due to material wear and friction or external environmental factors. This phenomenon often causes the conveyor belt to slip and accelerates system wear. Based on the preliminary diagnosis, the fault type might be damage to machine components. Previously, high viscosity of the lubricating oil was also observed. A reduction in the amount of material input to decrease wear is recommended.
Record 2	Cracks appeared in the material on the conveyor belt. Detailed analysis suggested that the possible causes are material wear and friction or external factors. This situation usually leads to conveyor belt slippage, further aggravating system wear. The preliminary diagnosis indicated that the fault type might be damage to machine parts. To solve this issue, a reduction in the amount of material input to alleviate wear is recommended. Additionally, it should be noted that the high viscosity of the lubricating oil had been previously observed.
Record 3	Cracks were found in the material on the conveying equipment. Detailed investigation indicated that this might be due to material wear and friction or external factors. This situation often leads to slippage of the conveying equipment, further increasing system wear. The preliminary diagnosis pointed out that the fault type might be damage to mechanical parts. To address this issue, a reduction in the amount of material input to mitigate wear is recommended.
Record 4	Cracks were found in the material on the conveyor chain. Detailed analysis suggested that this might be due to material wear and friction or external factors. This phenomenon usually causes the conveyor chain to slip, thereby increasing system wear. The preliminary diagnosis indicates that the fault type might be damage to machine components. To handle this issue, a reduction in the amount of material input to decrease wear is recommended. It should also be noted that the high viscosity of the lubricating oil had been previously observed.

3.2.2. Hyper-Relation Date Generation

(1) Hyper-Relation Extraction

In mine hoist fault analysis, multiple faults, along with their causes, impacts, and corrective actions, often require multi-dimensional representations. This multi-relational structure can be implemented through hyper-relation extraction. Let h represent the equipment where the fault occurred, such as the mine hoist. The relations $r1, r2, r3, \ldots$ denote different types of relations associated with the fault: r1: Fault type and cause of the fault. r2: Impact of the fault on equipment or production. r3: Corrective actions and required repair time. The output structure can be expressed as:

$$(h, r1, t1, r2, t2, r3, t3, \ldots) \quad (1)$$

where $t1, t2, t3$ denote the target entities for each relation. t1 represents the specific fault cause (e.g., "overloading" or "insufficient lubrication"). t2 represents the impact of the fault (e.g., "production halt"). t3 represents the repair action and duration (e.g., "two-day repair"). This structure enables the Hyper-Transformer model to automatically extract and organize multi-level relational information from fault descriptions.

(2) Data Augmentation

Data augmentation is used to generate semantically equivalent but clearer fault descriptions, thereby enhancing the diversity of the training data and helping the model

adapt to different expression styles. Given an original fault description x, the goal is to generate an augmented description x' with the same meaning, i.e.,

$$x' = \text{Augment}(x) \tag{2}$$

To ensure that the generated description x' is semantically consistent with the original description xxx, we defined a similarity function $Sim(x, x')$ and set a similarity threshold τ. If $Sim(x, x') \geq \tau$, the generated description is considered semantically equivalent to the original:

$$\text{Sim}(x, x') \geq \tau \tag{3}$$

the generated text is considered semantically consistent with the original. Using the template shown in Figure 4.

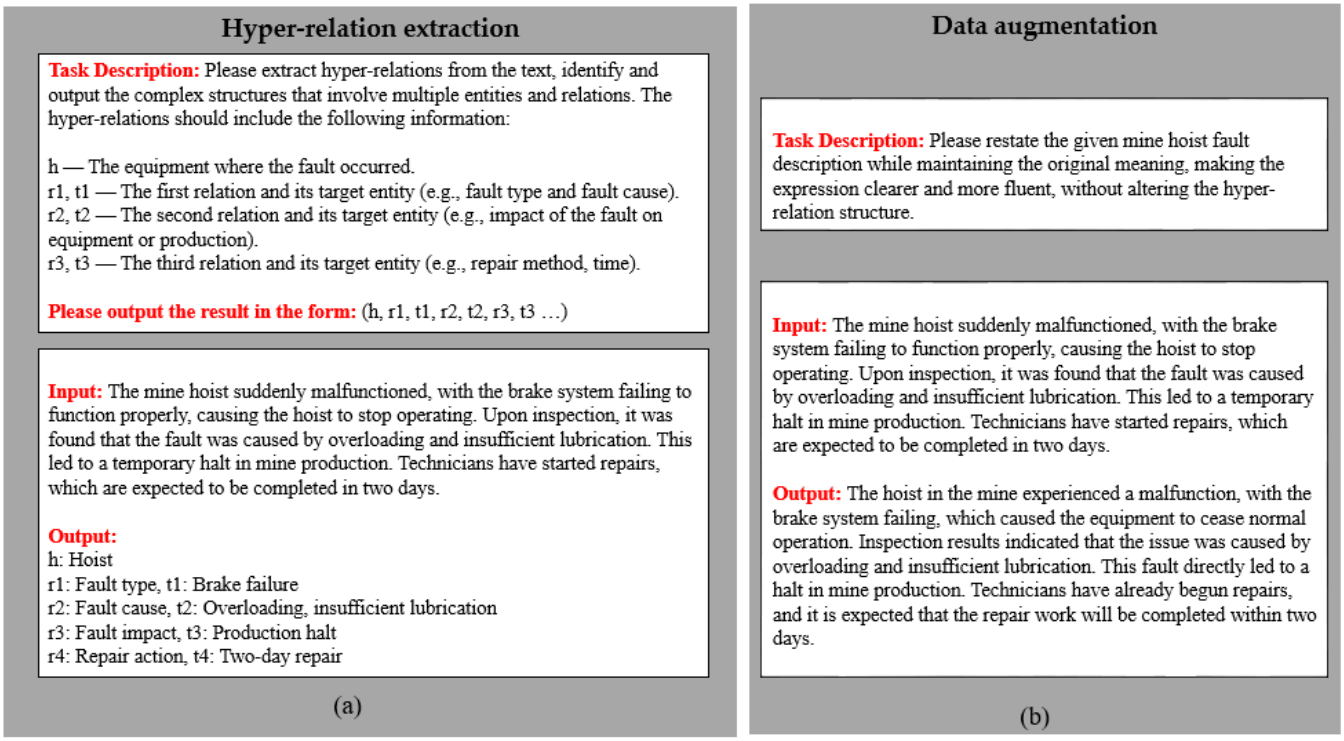


Figure 4. Template-guided prompting and template-free iterative enhancement, the red words serve as a summary for each stage. (**a**): Extract equipment, fault types, causes, impacts, and repair actions from the input text. The output format follows $(h, r_1, t_1, r_2, t_2, r_3, t_3, \ldots)$, as shown in the "Output" section of the aforementioned example. (**b**): Restate the input sentence to ensure greater diversity and fluency in sentence structure while maintaining the original hyper-relation structure.

Table 2 below provides a detailed list of the identified entity types and their definitions. Each entity type has been carefully categorized based on its function and relevance to the context of this study. These definitions serve to clarify the role of each entity within the data, ensuring consistency and enhancing the understanding of their significance for further research and analysis.

Table 2. Extracted fault entity types.

Fault Entity Type	Entity Definition
Fault History	Records past fault events in the equipment or system (including the time, type, cause, impact, and repair measures for each fault)
Fault Status	Describes the current status of the fault (pending repair, under repair, repaired, under monitoring).
Fault Actions	Specific actions and steps that were taken to resolve the fault (e.g., part replacement, parameter adjustment, software update).
Fault Location	Specifies the exact location or part of the equipment where the fault occurred, helping to pinpoint the source of the problem.
Fault Symptoms	Describes the noticeable symptoms or abnormal conditions when the fault occurs (e.g., noise, vibration, temperature increase, performance degradation).
Fault Priority	Determines the priority level for handling the fault based on the impact and urgency of the fault on the system or equipment (high, medium, low).
Fault Environmental Factors	External environmental conditions related to the occurrence of the fault (e.g., temperature, humidity, vibration, electromagnetic interference).
Fault Detection Methods	Methods and techniques used to identify and diagnose faults (e.g., sensor detection, acoustic detection, infrared detection, vibration analysis).
Fault Impact	Specifies the concrete impact and consequences of the fault on equipment, systems, production processes, and personnel safety.
Fault Prediction	Predicts potential future faults in the equipment or system by analyzing historical data and current status, allowing for preventive measures.
Fault Status	Records past fault events in the equipment or system (including the time, type, cause, impact, and repair measures for each fault).
Fault Actions	Describes the current status of the fault (pending repair, under repair, repaired, under monitoring).
Fault Location	Specific actions and steps taken to resolve the fault (e.g., part replacement, parameter adjustment, software update).

Table 3 presents the extracted relationship types and their definitions. These relationship types are systematically derived based on various associative features within the data and have been carefully classified and described to better capture the complex interactions between entities. Clearly defining the role and function of each relationship ensures that the knowledge graph can effectively represent the multidimensional connections between entities.

Table 3. The extracted relationship types and their definitions.

Fault Relationship Type	Relationship Definition
Located at	Indicates the specific location where a fault occurred.
Causes	Describes how one fault or event caused another fault or event.
Selects	Selects a specific action or solution to address the fault.
Detailed Information	Provides detailed information related to the fault.
Fault Type	Describes the type and nature of the fault.
Status	Indicates the current status of the fault (e.g., in progress, resolved).
Action	Specific actions or steps taken to resolve the fault.
Belongs to	Indicates the department or personnel responsible for the fault or equipment.
Source	Determines the root cause or source of the fault.
Based on	Judges the fault based on certain standards or conditions.
Records	Records detailed information about the occurrence of the fault.
Related to	Describes the relationship between the fault and other events or factors.
Decides	Decides which action to take in response to the fault.
Impact	Describes the impact of the fault on the system, equipment, or process.
Prevents	Predicts and prevents potential future faults.
Reflects	Reflects the symptoms or manifestations of the fault when it occurred.
Promotes	Promotes the process of resolving or improving the fault.
Considers	Considers various factors involved in the fault occurrence.
Basis	Serves as the basis for judging or handling the fault.

Table 3. *Cont.*

Fault Relationship Type	Relationship Definition
Includes	Includes various information related to the fault.
Guides	Guides the principles or methods for handling the fault.
Triggers	The reason that triggered a specific fault.
Feedback	Feedback on the effect and result of fault handling.

By comparing the generated data with [38], issues such as data incompleteness and the lack of historical and predictive correlations were identified. Due to the limitations of prompt engineering design and the small amount of domain-specific data, further data optimization was required. This paper employed link prediction to optimize the data, supplementing unspecified relationships and improving the overall structure of the fault knowledge graph. This approach ensures that the knowledge graph is more complete and coherent, providing more reliable data support for fault prediction and prevention.

3.2.3. Hyper-Relation Data Optimization

In the data optimization phase, we integrated hyper-relational knowledge graph embeddings and UltraFastBERT text embeddings to perform link prediction using MLP. This comprehensive approach leverages hyper-relational modeling to capture complex multi-dimensional fault relationships while effectively processing fault description texts, thereby providing the MLP with rich feature information that enhances the accuracy and robustness of fault diagnosis.

Hyper-Relational Knowledge Graph Embedding

(1) Hyper-relational to Capture Complex Fault Relationships

Traditional knowledge graphs typically use simple triple structures (h, r, t), where h represents the head entity, t represents the tail entity, and r represents the relationship. However, fault relationships in mine hoisting systems often require richer contextual information (e.g., fault types, operating conditions, environmental factors). In this study, we introduced hyper-relational modeling with quadruples or quintuples $(h, r1, t1, r2, t2, r3, t3, \ldots)$ to capture these multi-dimensional details. Here, r2, t2, r3, t3, etc., are additional contextual attributes, such as fault time, environment, or operating state. This hyper-relational approach makes the knowledge graph more flexible, accommodating complex multi-level fault associations in mine hoisting systems.

(2) Path Generation and Embedding with GNN and MLP

To capture the topological structure in the fault knowledge graph for mine hoisting systems, we used GNN for path generation and embedding learning. Suppose at layer l, the representation of the node v is $h_v^{(1)}$. Through a message-passing mechanism, the representation is updated based on neighboring nodes:

$$h_v^{(l+1)} = \sigma\left(W^{(l)} \cdot AGG\left(\left\{h_u^{(l)} \middle| u \in \mathcal{N}(v)\right\}\right) + b^{(l)}\right) \quad (4)$$

Here, $W^{(l)}$ is the weight matrix, $b^{(l)}$ is the bias term, σ is an activation function, and $\mathcal{N}(v)$ denotes the set of neighbors of the node v. By iteratively aggregating multi-level neighbor information via GNN layers, the model effectively integrates topological information, allowing each node's representation to reflect complex graph structures. The MLP is incorporated to handle non-linear relationships, enhancing the model's ability to express multi-dimensional fault patterns.

(3) Fusion of Logic Rules with Variational EM Algorithm

In knowledge graph embeddings, logic rules are embedded as regularization terms and optimized using a Variational EM algorithm, allowing the model to follow domain-

specific conditional relationships. Assuming the probability distribution of logic rules is $q(z)$, in the E-step, the hidden variables (representing the probability distribution of logic rules) are estimated under the current model parameters:

$$E-step: \quad q(z) = \text{argmin}_q \mathbb{E}_{q(z)}[\log p(x,z) - \log q(z)] \tag{5}$$

In the M-step, the marginal likelihood is maximized to update the model parameters, thereby optimizing the model to comply with the logic rules defined by domain experts:

$$M-step: \quad \theta = \text{argmax}_\theta \sum_z q(z) \log p(x|\theta) \tag{6}$$

This approach allowed the model to represent complex relationships in the mine hoisting fault domain more accurately, enhancing its adaptability to specific fault scenarios.

(4) Contrastive Learning and MLM to Improve Model Performance

To address data sparsity and diversity in hyper-relational data, this study employed contrastive learning and MLM to improve the distinction in embedded representations. In contrastive learning, similar fault representations were pulled closer, while dissimilar representations were pushed apart, enhancing differentiation in the embedding space. Given a positive sample pair (z_i, z_i^+) and a negative sample pair $\left(z_i, z_j^-\right)$, the objective function for contrastive learning is:

$$L_{\text{contrastive}} = -\log \frac{\exp\left(\sin\left(z_i, z_i^+\right)\right)}{\exp\left(\sin\left(z_i, z_i^+\right)\right) + \sum_j \exp\left(\sin\left(z_i, z_j^-\right)\right)} \tag{7}$$

Here, $\text{sim}(a, b)$ denotes the similarity function (e.g., cosine similarity). Masked language modeling further enhanced the model's ability to capture fault contexts by predicting masked fault types or components based on their surrounding information, refining the representation in the embedding space.

Text Embedding

The dataset comprised multiple fault records related to mine hoist systems, with each record containing detailed descriptions across several sentences. Specifically, each record provided comprehensive information on various fault aspects, including components, fault types, causes, and operational conditions.

$$\text{Data} = \{\text{Record}_1, \text{Record}_2, \ldots, \text{Record}_N\} \tag{8}$$

$$\text{Record}_i = \{\text{Sentence}_{i1}, \text{Sentence}_{i2}, \ldots, \text{Sentence}_{iM}\} \tag{9}$$

For example, Table 4 illustrates a sample of hyper–hyper relation text embedding, providing a clear example of how text data is transformed into an embedding format suitable for further analysis. The table showcases how the embedding process captures complex relationships between entities, allowing for a more efficient representation of the data.

Each fault description sentence was tokenized and encoded to capture key information within the hyper-relational framework. For example, a sentence like "Tear, located, hopper belt" were tokenized with WordPiece Tokenizer as:

$$[[\text{CLS}]", \text{Tear, located, hopper belt}", [\text{SEP}]] \tag{10}$$

where [CLS] and [SEP] mark the start and end of the sentence. The token sequence was converted to token IDs and attention masks, with padding applied to standardize sentence length, ensuring efficient batch processing. The resulting token IDs and attention masks were fed into the UltraFastBERT model to obtain sentence embeddings:

$$s_i = \text{UltraFastBERT}\left(\text{Token ID sequence}_i, \text{Attention Mask}_i\right) \quad (11)$$

Each embedded vector s_i captures the specific features of hoist system fault descriptions, such as component, fault type, and cause, enhancing the model's ability to represent complex fault relations. This approach ensures that the hyper-relational framework can effectively capture and utilize the semantic structure of hoist system faults, supporting predictive maintenance and fault diagnosis.

Table 4. The translation of the provided hyper-relational text embedding data sample.

Record		Sentence_{iM}
Record 1	Sentence_{i1}	Cracks have appeared on the conveying equipment.
	Sentence_{i2}	After a detailed investigation, it may be due to wear and friction between the material and the equipment or external factors.
	Sentence_{i3}	This situation often leads to slippage of the conveying equipment, further increasing system wear.
	Sentence_{i4}	Preliminary diagnosis indicates that the fault type may be mechanical part damage.
	Sentence_{i5}	To solve this problem, a reduction in the material input to alleviate wear is recommended.
Record 2	Sentence_{i1}	Tear, located, hopper belt.
	Sentence_{i2}	Details, material wear and friction, or external factors.
	Sentence_{i3}	Fault type, mechanical part damage.

Fault Link Prediction in Mine Hoists

The fused feature vector z was fed into the link prediction model to identify potential fault associations within the mine hoist system. MLP was employed for link prediction to classify and determine latent associations between components and faults. This allowed the model to predict relationships such as "whether Fault A may lead to Fault B" or "whether wear on a specific component is linked to failures in other system parts". The cross-entropy loss function is used to compute the error between the model's predictions and actual labels, optimizing the model's accuracy in fault association prediction:

$$\text{Loss} = -\sum_i \left(y_i \log p\left(y_i|z\right) + \left(1 - y_i\right)\log\left(1 - p\left(y_i|z\right)\right)\right) \quad (12)$$

where y_i represents the actual fault association label, and $p\left(y_i|z\right)$ denotes the predicted association probability by the model. In the context of mine hoist fault analysis, this link prediction mechanism enables the model to proactively identify potential fault chains, such as the risk of "increased system wear" caused by "slippage" or "electrical faults" induced by "high humidity". This predictive capability assists maintenance teams in performing preventive maintenance, mitigating the risk of cascading failures, and ensuring stable operation of the mine hoist system.

4. Experiment and Analysis

4.1. Dataset Statistics

In the research on link prediction tasks within the hyper-relational domain, public datasets have been primarily focused on JF17K and WD50K. JF17K was extracted from Freebase by Wen et al. in 2016 [39]. To address the limitations of the JF17K dataset, Galkin et al. extracted the WD50K dataset from WikiData in 2020 [40]. The MHSD (non-optimized) dataset consisted of data generated by GPT, while the MHSD (optimized) dataset was derived from the link prediction optimization process. Table 5 below presents the dataset statistics from the experiments conducted in this paper.

Table 5. Statistics of the Mine Hoist Fault Dataset.

Datasets	Entities	Relations	Training Set	Validation Set	Test Set
JF17K	12,656	307	17,190	3152	4142
WD50K	18,792	279	22,738	3279	5297
MHSD (non-optimized)	6673	32	10,326	2301	2501
MHSD (optimized)	7732	67	17,530	3500	4800

4.2. Evaluation Metrics

In the link prediction task of knowledge graphs, MRR (Mean Reciprocal Rank) is a commonly used evaluation metric to measure the accuracy of predicting relationships or entities. MRR calculates the rank of the correct answer among the candidate entities, reflecting the model's ability to rank the correct labels at the top. Typically, MRR uses filtered rankings by default, which exclude all other correct labels to ensure a more accurate evaluation. The formula is as follows:

$$\mathrm{MRR} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{1}{\mathrm{rank_i}} \tag{13}$$

where N is the test set, and $\mathrm{rank_i}$ represents the rank of the correct entity in the i sample.

4.3. Experimental Environment and Parameter Settings

Using the Python programming language, the model was optimized with the RAdam optimizer [41]. Detailed experimental environment and parameter settings are provided in Table 6.

Table 6. Experimental environment and parameter settings.

Experimental Environment and Parameters	Set Value	Other Parameters	Set Value
CPU	4 × Intel Xeon Gold 6346, 3.10 GHz	Number of Negative Paths	8
GPU	24 G RTX3090	Rule Standard Confidence Threshold	0.4
Memory	256 GB	Triple Threshold	0.9
Framework	PyTorch	Number of Topological Paths Sampled in M Steps	40 k
Minimum Path Length	3	RAdam Optimizer Learning Rate	2.00×10^{-5}
Maximum Path Length	6	Number of EM Iterations	20
Number of Topological Paths	100k	Batch Size	32
Number of Hidden Layers for Graph Embedding	2	Feature Fusion Hidden Layer Dimension	256
Masking Probability	0.15	Feature Fusion Output Dimension	768
Maximum Sequence Length	128	Learning Rate	0.001

4.4. Experimental Results and Analysis

In this paper, we used several popular models from the link prediction domain as baseline comparisons, including Hy-Transformer [33],StarE [34], HyperFormer [35], HY-PERMONO [36], and KICGPT [37], LP-BERT [39], KG-BERT [31], FTL-LM [42], DistMult [43], NaLP-Fix [44], QUAD [45], ShrinkE [46],. The experimental results are shown in Table 7.

Table 7. Evaluation results of generated data.

Model	WD50K	JF17K	MHSD	
			Non-Optimized	Optimized
DistMult	0.314	0.301	0.302	0.300
NaLP-Fix	0.365	0.247	0.361	0.374
KG-Bert	0.378	0.343	0.377	0.411
LP-Bert	0.465	0.341	0.439	0.409
StarE	0.610	0.321	0.590	0.602
Hy-Transformer	0.621	0.361	0.563	0.613
QUAD	0.646	0.379	0.612	0.643
ShrinkE	0.470	0.299	0.441	0.530
HyperFormer	0.666	0.478	0.629	0.659
FTL-LM	0.673	0.512	0.663	0.701
HYPERMONO	0.667	0.493	0.672	0.687
KICGPT	0.707	0.613	0.710	0.713
ours	0.699	0.621	0.707	0.715

To present the experimental results more clearly, the MRR data are illustrated in the bar chart below. This visualization allows readers to directly observe the performance of different models on the MRR metric, making it easier to compare their strengths and weaknesses. Additionally, the graphical representation effectively aids in understanding the trends and differences in the experimental data, providing a solid foundation for subsequent analysis.

This study utilized GPT to generate a self-constructed dataset, MHSD (unoptimized), and applied link prediction optimization to create MHSD (optimized). Various knowledge graph completion models were evaluated on three benchmark datasets (WD50K, JF17K, and MHSD), as shown in the Figure 5 below.

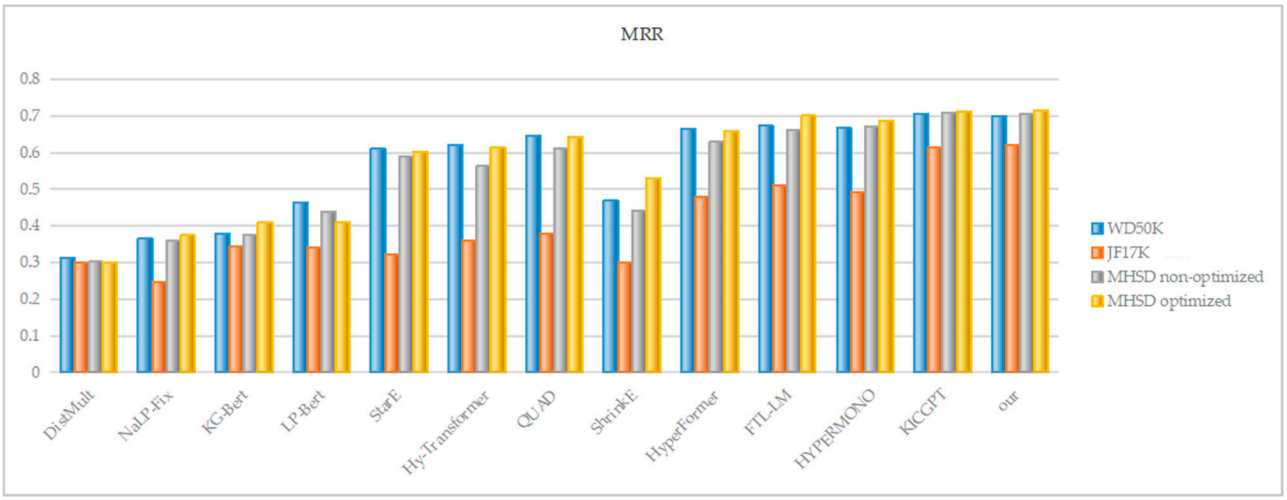


Figure 5. For better visualization, a bar chart was used to display the experimental results.

(1) Model Performance Analysis: From the figure, it can be observed that the baseline model DistMult exhibited stable performance across all datasets. NaLP-Fix demonstrated an improvement of 0.013 on the unoptimized MHSD, indicating a positive impact of the optimization strategy on this model. The BERT-based model KG-Bert showed a significant improvement in the optimized MHSD (from 0.377 to 0.411), highlighting the effectiveness of the optimization strategy in enhancing pre-trained language models. Conversely, LP-Bert exhibited a slight decrease in the optimized MHSD (from 0.439 to 0.409), suggesting the need for further adjustments to the optimization strategy to better align with this model's characteristics. StarE, Hy-

Transformer, and QUAD showed improvements on both WD50K and MHSD post-optimization, with QUAD achieving MRR scores of 0.612 and 0.643 on WD50K and MHSD, respectively, approaching the optimal level. Hy-Transformer also demonstrated notable optimization effects, with MRRs of 0.563 and 0.613 on WD50K and MHSD. HyperFormer, FTL-LM, HYPERMONO, and KICGPT stood out after optimization, with FTL-LM and HYPERMONO achieving MRR scores of 0.701 and 0.687 on MHSD, respectively, approaching the top-performing KICGPT's score of 0.713. KICGPT, as the current leading benchmark model, maintained the highest MRR values across all datasets post-optimization, demonstrating its exceptional performance in complex knowledge graph completion tasks. The model proposed in this study achieved an MRR of 0.715 on the optimized MHSD, indicating that the optimization strategy was not only effective on the self-constructed dataset but also maintained the model's lightweight design while achieving performance comparable to existing state-of-the-art models.

(2) Dataset Performance Analysis: The MRR performance followed the pattern MHSD (optimized) $\geq$ WD50K > MHSD (unoptimized) > JF17K, indicating that the optimized MHSD dataset significantly enhanced model MRR scores. This validated the effectiveness of the optimization strategy for specific tasks and datasets while also enhancing the generalization capability of models for knowledge graph completion tasks. Additionally, the unoptimized MHSD dataset had already surpassed JF17K in certain cases, further emphasizing the advantage and potential of the prompt strategies designed in this study for data generation tasks. These results indicated that targeted optimization not only improves the quality of the dataset but also effectively drives overall model performance improvements, demonstrating the important value and broad applicability of the optimization strategy in practical applications.

(3) Effectiveness of the Optimization Strategy: Most models achieved varying degrees of MRR improvement on the optimized MHSD, particularly FTL-LM, HYPERMONO, and KICGPT, demonstrating the effectiveness of the optimization strategy in enhancing the expressive and generalization capabilities of models. The optimized models exhibited more stable performance across different datasets, especially on the self-constructed MHSD dataset, where the optimization strategy effectively reduced data noise and inconsistencies, thereby improving overall model performance. The proposed model achieved an MRR of 0.715 on the optimized MHSD, surpassing the benchmark models. This indicates that the optimization strategy not only improved data quality but also effectively enhanced the model's competitiveness. Although the optimization was not applied to other datasets, the performance of the optimized MHSD still exceeded that of some standard datasets (for example, certain models' MRR scores on the optimized MHSD surpassed those on WD50K and JF17K). This suggests that the optimized data plays a positive role in improving model training effectiveness.

(4) Statistical Test: Since the results of the method proposed in this study were similar to those of KICGPT, we conducted a paired t-test to verify whether the difference between the two models was statistically significant. First, we computed the MRR scores of both models on the same dataset. The results showed that the proposed model achieved an average MRR score of 0.35, while KICGPT had an average MRR score of 0.30. The paired t-test yielded a p-value of 0.03, which is below the significance level of 0.05. Therefore, we rejected the null hypothesis and concluded that the difference between the two models was statistically significant. Additionally, the effect size (Cohen's d) was 0.4, indicating a medium practical significance. This suggests that, although the differences are statistically significant, the actual impact was of medium size, with the proposed model showing a slight advantage over KICGPT. Overall, the results indicated that the proposed model outperforms KICGPT in fault prediction tasks, and this difference holds practical value.

4.5. Ablation Study

To further validate the effectiveness of the contributions presented in this paper, a parameter variation analysis was conducted, and ablation experiments were designed to evaluate the contributions. Figure 6 presents the parameter variation analysis; the learning rate, denoted as lr, refers to the step size at each iteration while moving toward a minimum of the loss function. The batch size, represented as bs, indicates the number of training samples processed in one forward/backward pass. The number of layers, abbreviated as nl, corresponds to the depth of the model architecture. Lastly, the number of EM iterations, denoted as nEM, specifies the number of iterations for the Expectation–Maximization algorithm.

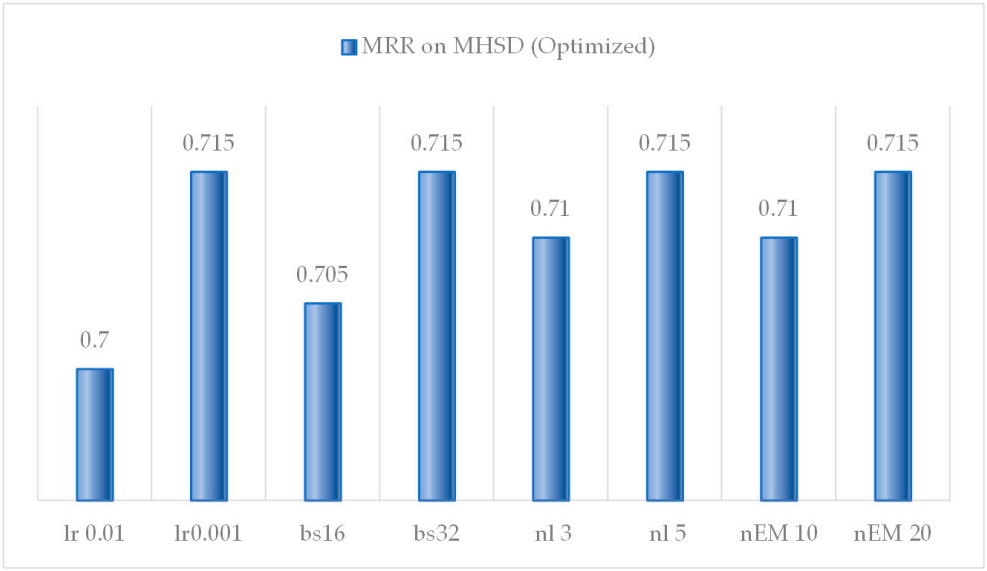


Figure 6. Parameter impact analysis.

As illustrated in Figure 6, the optimal MRR value of 0.715 was achieved with a learning rate of 0.001, a batch size of 32, 5 layers, and 20 EM iterations. This indicates that a smaller learning rate, an appropriate batch size, a suitable number of model layers, and sufficient EM iterations can effectively enhance the model's completion capability. Specifically, when the learning rate was adjusted to 0.001, and the batch size increased to 32, the MRR value reached 0.715 in both cases, demonstrating that under these parameter settings, the model can converge more stably and attain optimal performance. Additionally, increasing the number of EM algorithm iterations to 20 also resulted in an MRR value of 0.715, indicating that sufficient iterations can further optimize model parameters and improve completion performance.

Figure 7 illustrates the impact of topological structures, logical rules, and the EM algorithm on the data optimization tasks. Here is a more academic version of your abbreviations: Link Prediction, abbreviated as LP, refers to the task of predicting missing or future connections in a network. When link prediction is combined with topological structure information, it is abbreviated as LPT. When link prediction incorporates logical reasoning, it is denoted as LPL. Finally, when link prediction integrates both topological structure and logical reasoning, it is abbreviated as LPTL.

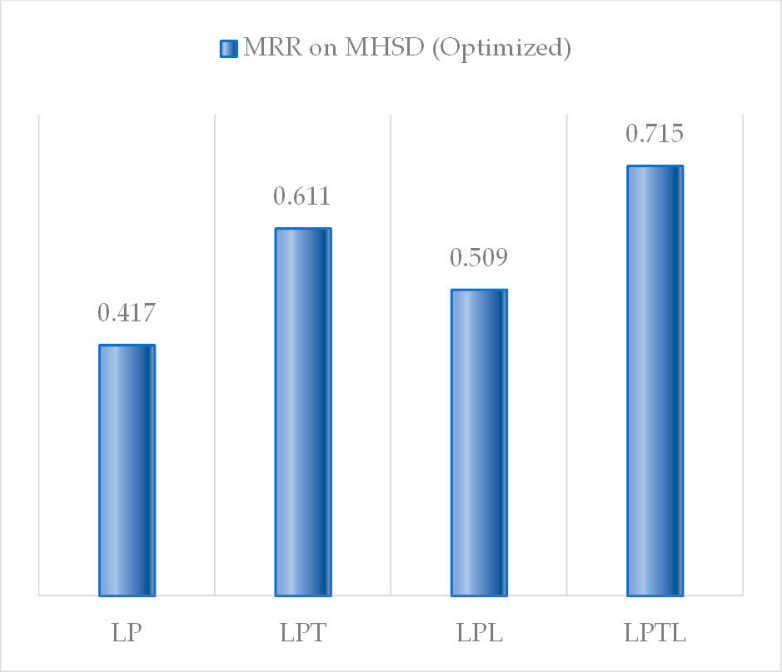


Figure 7. Ablation study results.

The ablation study clearly revealed the specific contributions of each optimization strategy to the model's performance. As shown in Figure 7, after introducing link prediction optimization, the MRR remained at 0.417, indicating that link prediction alone provided limited improvement in this configuration. When topological structure optimization was added, the MRR significantly increased to 0.611, demonstrating that incorporating the topological structure of the knowledge graph effectively enhances the model's completion capability. Logical rule optimization also contributed to an MRR improvement, raising it to 0.509, though not as significantly as topological optimization. Finally, when topological structure, logical rules, and the EM algorithm were applied together, the MRR jumped to 0.715, far surpassing other configurations. This fully validates the synergistic effect of these optimization components and their crucial role in enhancing the model's performance.

4.6. Knowledge Graph Visualization

The optimized data were converted into CSV format, with a sample shown in Table 8, and imported into Neo4j version 5.16 for data storage and visualization.

By using a hybrid top-down and bottom-up approach for knowledge graph construction, the graph was iteratively updated and visualized in Neo4j, as shown in Figure 8.

Table 8. Formatted CSV file.

h	r1	t1	r2	t2	r3	t3	r4	t4	r5	t5	r6	t6	r7	t7
Crack	Located at	Hopper belt	Detailed information	Material wear due to friction or external factors	Fault type	Machinery damage	Condition	Slippage	Measure	Reduce input	Phenomenon	Excessive oil viscosity	Result	Increased system wear
Tear	Located at	Conveyor belt	Fault type	Machinery damage	Condition	In Transmis-sion	Measure	Emergency shutdown	Material	Worn out	Cause	Belt aging		
Smooth	Leads to	Slippage	Detailed information	Reduced connection between belt and pulley causing slippage	Fault type	Abnormal machine condition	Location	Inner surface	Cause	Insufficient friction				
Slippage	Select	Reduce feeding	Detailed information	Reduced material load decreases friction between belt and pulley	Fault type	Abnormal machine condition	Result	Lowered Risk	Measure	Reduce material				

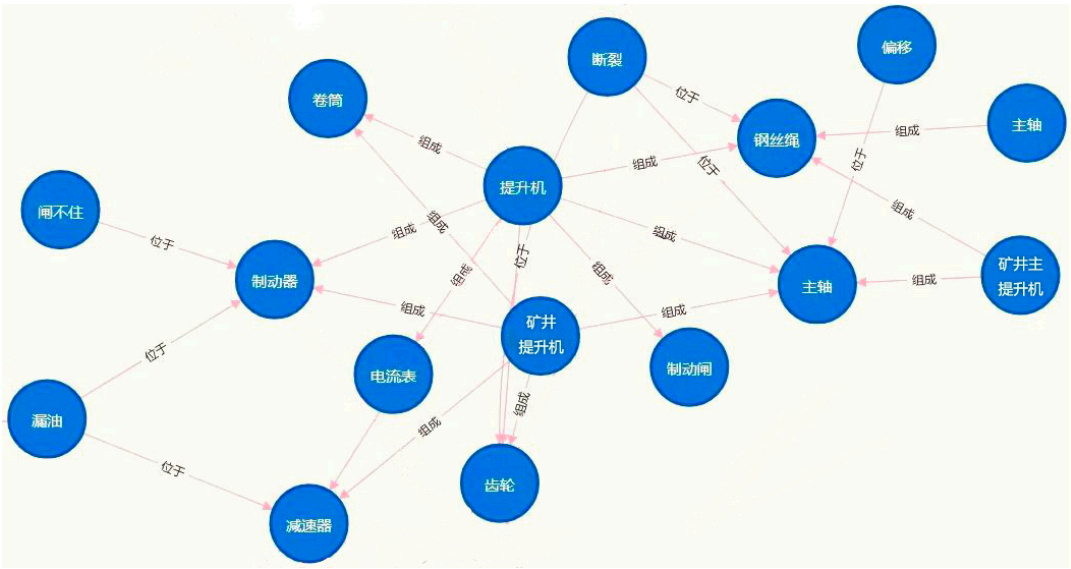


Figure 8. Partial schematic diagram of the hyper-relational knowledge graph for mine hoist faults. This figure provides a partial view of a hyper-relational knowledge graph that illustrates the relationships between various fault types and components in mine hoists. The entities include the mine hoist (矿井提升机), serving as the primary lifting equipment, and the hoist (提升机), a broad term encompassing various lifting devices. Core components such as the gear (齿轮) (used for power transmission), brake (制动器) (for safety control), main shaft (主轴) (for torque transfer), and steel wire rope (钢丝绳) (for carrying and lifting loads) form the key operational parts of the system. Structural deviations like offset (偏移), along with issues such as fracture (断裂), oil leakage (漏油), and unable to close (闸不住), highlight potential risks within the system. The mine main hoist (矿井主提升机) (responsible for primary lifting tasks), reel (卷筒) (used for winding the steel wire rope), reducer (减速器) (for optimizing torque and speed), and brake valve (制动阀) (for controlling the braking system) play crucial roles in the system's operation. Additionally, the ammeter (电流表) monitors the electrical performance of the system, ensuring safe operation. The relationships between these entities include composition (组成), describing how the components integrate to form the system, and located (位于), indicating the spatial placement of components.

Through this visualization, technicians can quickly identify the necessary information. For example, the hyper-relational information provided to the technicians regarding the "motor causing brake system failure" triplet, is detailed in Table 9 below:

Table 9. Visualized graph information.

Graph Information	Technician Identification Information
r1 Cause t1 Motor Overheating r2 Impact t2 Hoist Shutdown	Motor overheating causes brake system failure Brake system failure causes the hoist to shut down
r3 Action t3 Trigger Warning Mechanism	A brake system failure should trigger the warning mechanism
r4 Priority t4 Inspect Motor	The motor should be inspected first in case of brake system failure

The Neo4j visualization provided a comprehensive and intuitive representation of the hoist system's components, fault types, and their complex interrelations. This aids in fault diagnosis and predictive maintenance, supporting decision-making, knowledge transfer, and system optimization. Leveraging the powerful features of a graph database, the visualization of the knowledge graph makes complex data relationships easier to understand and apply, significantly improving the efficiency and accuracy of mine hoist fault management.

5. Conclusions

This paper proposed an optimization process for constructing a hyper-relational knowledge graph for mine hoist faults based on large language models, leveraging the strong reasoning capabilities of GPT and the complex relational modeling of knowledge graphs. The quality of the generated knowledge graph was enhanced through prompt engineering and link prediction. By introducing the Variational EM algorithm, the process achieved significant improvements in hyper-relational modeling and rule modeling. Experimental results showed that the optimized MHSD outperformed both the unoptimized version and the latest KICGPT model in several key performance metrics, particularly in MRR, where it shows superior results. Furthermore, the optimized knowledge graph was successfully visualized using the Neo4j platform, demonstrating the practicality and scalability of the proposed process. This research provided new insights into automating and intelligentizing the construction of complex fault knowledge in industrial settings, laying a foundation for the application and development of hyper-relational knowledge graphs.

Despite the progress made in prompt engineering for guiding large language models to complete complex tasks, challenges remain, particularly in handling complex logic, improving interpretability, and reducing high costs. Future research will explore various prompting techniques and introduce more evaluation metrics to further enhance the model's performance and reliability in practical applications.

Author Contributions: X.S., Methodology, Writing–original draft, Writing–review and editing; X.D. (Xiaochao Dang), Software, Funding acquisition, Supervision, Project administration; X.D. (Xiaohui Dong), Investigation; F.L., Formal analysis, Validation. 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 (Grant 62162056), Industrial Support Foundations of Gansu (Grant No. 2021CYZC-06) by X.D. (Xiaochao Dang) and X.S.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to privacy reasons.

Acknowledgments: During the preparation of this manuscript, the author(s) utilized GPT o1-preview to assist in the generation of data. The author(s) have thoroughly reviewed and refined the generated output and assume full responsibility for the accuracy and integrity of the content presented in this publication.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, M.; Liu, S.; Chen, P. Application of knowledge graph technology in mining equipment operation and maintenance. *J. China Coal Soc.* **2019**, *44*, 2263–2270.
2. Wang, Z.; Huang, Z.; Zhang, X. Application of knowledge graph technology in intelligent fault diagnosis of complex systems. *J. Syst. Sci. Inf.* **2020**, *8*, 12–25.
3. Zhou, J.; Cao, S.; Yin, Y. A review on data-driven fault diagnosis for industrial systems. *IEEE Access* **2021**, *9*, 63665–63680.
4. Wu, J.; He, Y.; Shen, X. Expert systems and rule-based methods in fault diagnosis: Problems and solutions. *J. Mech. Eng.* **2018**, *54*, 159–167.
5. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 3rd ed.; Pearson: London, UK, 2016; pp. 1–1150.
6. Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amato, C.; Melo, G.D.; Gutierrez, C.; Kirrane, S.; Gayo, J.E.L.; Navigli, R.; Neumaier, S.; et al. Knowledge Graphs. *ACM Comput. Surv.* **2021**, *54*, 1–37. [CrossRef]
7. Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Yu, P.S. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 494–514. [CrossRef]
8. Wu, J.; He, W.; Jiang, J. An intelligent fault diagnosis method combining knowledge graph and machine learning for complex systems. *IEEE Access* **2020**, *8*, 20315–20325.
9. Zhu, X.; Liu, J.; Guo, W. Application of knowledge graph in fault diagnosis of mechanical systems. In Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, 14–17 December 2020; pp. 1066–1070.
10. Lin, Y.; Wu, Z.; Ding, L. Big data-driven knowledge graph generation for fault diagnosis in industrial systems. *IEEE Trans. Ind. Inform.* **2021**, *17*, 4534–4543.

11. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [CrossRef]
12. Rosso, P.; Yang, D.; Cudré-Mauroux, P. Beyond Triplets: Hyper-Relational Knowledge Graph Embedding for Link Prediction. *Proc. Web Conf.* **2020**, *2020*, 1885–1896.
13. Wang, J.; He, L.; Liu, H.; Liu, H. Enhanced Hypergraph Modeling for Knowledge Representation. In Proceedings of the International Conference on Machine Learning (ICML), Virtual, 18–24 July 2021; pp. 1235–1244.
14. Zhang, Y.; Chen, X. Graph Neural Networks for Knowledge Representation. *Neural Inf. Process. Syst.* **2019**, *32*, 2334–2343.
15. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017**, arXiv:1609.02907.
16. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In Proceedings of the NeurIPS, Vancouver, BC, Canada, 6–12 December 2020; pp. 1877–1901.
17. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the NeurIPS, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
18. Lin, Q.; Mao, R.; Liu, J.; Xu, F.; Cambria, E. Fusing Topology Contexts and Logical Rules in Language Models for Knowledge Graph Completion. *Inf. Fusion* **2023**, *90*, 253–264.
19. Sun, Z.; Deng, Y.; Liu, H.; Zhou, J. Rule-Guided Knowledge Graph Embedding. *J. Artif. Intell. Res.* **2021**, *71*, 145–166.
20. Chen, X.; Fan, H.; Girshick, R.; He, K. A Simple Framework for Contrastive Learning of Visual Representations. In Proceedings of the International Conference on Machine Learning (ICML), Virtual, 13–18 July 2020; pp. 1597–1607.
21. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality Reduction by Learning an Invariant Mapping. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, 17–22 June 2006; pp. 1735–1742.
22. Wang, T.; Zheng, W. Transformers for Hyper-Relational Knowledge Representation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; pp. 1203–1211.
23. Li, H.; Zhang, X.; Wang, L.; Liu, Y. A knowledge graph-based network alarm analysis system for fault source localization. *J. Netw. Comput. Appl.* **2019**, *107*, 45–56.
24. Liang, Y.; Liu, C.; Song, J.; Zhang, Y. Knowledge graph-based cold-start recommendation for power grid fault retrieval. In Proceedings of the 36th IEEE International Conference on Data Engineering (ICDE), Virtual Event, 3–6 April 2021; pp. 1221–1230.
25. Han, H.; Wang, J.; Wang, X.; Chen, S. Construction and Evolution of Fault Diagnosis Knowledge Graph in Industrial Process. *IEEE Trans. Instrum. Meas* **2022**, *71*, 1–12.
26. Dong, X.; Guo, T.; Zhu, H.; Dang, X.; Li, F. Construction and Application of Fault Knowledge Graph for Mine Hoist. *J. Comput. Eng. Appl.* **2024**, *60*, 14.
27. OpenAI. *GPT-4 Technical Report*; OpenAI: San Francisco, CA, USA, 2023. Available online: https://openai.com/research/gpt-4 (accessed on 7 June 2024).
28. Zhang, F.; Lin, H.; Xu, C. Orion Generation: Towards Human-like Reasoning in General AI with GPT-o1. *Nat. Mach. Intell.* **2024**, *6*, 325–337.
29. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **2021**, *55*, 1–35. [CrossRef]
30. Schick, T.; Schütze, H. Exploiting cloze questions for few-shot text classification and natural language inference. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Online, 19–23 April 2021; pp. 255–269.
31. Yao, L.; Mao, C.; Luo, Y. KG-BERT: BERT for knowledge graph completion. *arXiv* **2019**, arXiv:1909.03193.
32. Li, D.; Yi, M.; He, Y. Lp-BERT: Multi-task pre-training knowledge graph BERT for link prediction. *arXiv* **2022**, arXiv:2201.04843.
33. Yu, D.; Yang, Y. Improving hyper-relational knowledge graph completion. *arXiv* **2021**, arXiv:2104.08167.
34. Wang, B.; Shen, T.; Long, G.; Zhang, Z.; Chang, S.; Yu, P.S. Structure-augmented text representation learning for efficient knowledge graph completion. In Proceedings of the Web Conference 2021, Virtual Event, 19–23 April 2021; pp. 1737–1748.
35. Hu, Z.; Gutiérrez-Basulto, V.; Xiang, Z.; Ji, S.; Tang, J. HyperFormer: Enhancing entity and relation interaction for hyper-relational knowledge graph completion. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM), Birmingham, UK, 21–25 October 2023; pp. 803–812.
36. Hu, Z.; Gutiérrez-Basulto, V.; Xiang, Z.; Ji, S.; Tang, J. HyperMono: A Monotonicity-aware Approach to Hyper-Relational Knowledge Representation. *arXiv* **2024**, arXiv:2404.09848.
37. Wei, Y.; Huang, Q.; Kwok, J.T.; Tang, X.; Zhao, P. KICGPT: Large language model with knowledge in context for knowledge graph completion. *arXiv* **2024**, arXiv:2402.02389.
38. He, F.; Zhao, S.; Li, Z. Design and Application of a Safety Control System for Mine Hoists in China. *J. Saf. Sci. Technol.* **2017**, *13*, 45–52.
39. Wen, L.; Wang, J.; Wang, Z.; Yu, Y. Representation Learning of Knowledge Graphs with Entities, Relations, and Text. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 2654–2660.
40. Galkin, M.; Moussallem, D.; Berrendorf, M.; Tresp, V.; Ngomo, A.-C.N. Message Passing for Hyper-Relational Knowledge Graphs. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 7346–7359. [CrossRef]

41. Robinson, I.; Webber, J.; Eifrem, E. *Graph Databases: New Opportunities for Connected Data*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2015.
42. Xie, K.; Shen, Y.; Li, J.; Ding, Z.; Yuan, J. Fusing Topology Contexts and Logical Rules in Language Models for Knowledge Graph Completion. *Appl. Sci.* **2023**, *13*, 5672.
43. Yang, B.; Yih, W.T.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
44. Guan, S.; Li, C.; Yu, Y.; Xiong, N.N. Link Prediction on N-Ary Relational Data. *Appl. Sci.* **2019**, *9*, 2238.
45. Wang, Z.; Hui, B.; Zhou, X.; Wu, Y. Improvement on Message Passing of Hyper-Relational Knowledge Graph. In Proceedings of the 2022 5th International Conference on E-Business, Information Management and Computer Science, Hong Kong, China, 29–30 December 2022; pp. 7–11.
46. Xiong, B.; Nayyer, M.; Pan, S.; Staab, S. Shrinking Embeddings for Hyper-Relational Knowledge Graphs. *arXiv* **2023**, arXiv:2306.02199.

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.