Large Language Model Based Intelligent Fault Information Retrieval System for New Energy Vehicles

Haiyu Zhang , Yinghui Zhao , Boyu Sun , Yaqi Wu , Zetian Fu * , Xinqing Xiao *

Posted Date: 18 March 2025

doi: 10.20944/preprints202503.1268.v1

Keywords: new energy vehicle fault; intelligent retrieval system; generative language models; knowledge graphs; retrieval-augmented generation

Article

Large Language Model Based Intelligent Fault Information Retrieval System for New Energy Vehicles

Haiyu Zhang [1], Yinghui Zhao [2], Boyu Sun [2], Yaqi Wu [1] and Zetian Fu [2,*]and Xinqing Xiao [2,*]

[1] Yantai Institute, China Agricultural University, Yantai 264670, China

[2] College of Engineering, China Agricultural University, Beijing 100083, China

* Correspondence: fzt@cau.edu.cn (Z.F.); xxqjd@cau.edu.cn (X.X.)

Abstract: In recent years, the rapid development of the new energy vehicle (NEV) industry has exposed significant deficiencies in intelligent fault diagnosis and information retrieval technologies, especially in intelligent fault information retrieval, which faces persistent challenges including inadequate system adaptability and reasoning bottlenecks. This study proposes a retrieval-augmented generation (RAG) framework that integrates large language models (LLMs) with KGs (KGs) through three key components: fault data collection, knowledge graph construction, and fault knowledge model training. The primary research contributions are threefold: (1) A domain-optimized fine-tuning strategy for LLMs based on NEV fault characteristics, verifying the superior accuracy of the Bidirectional Encoder Representations from Transformers (BERT) model in fault classification tasks. (2) A structured knowledge graph encompassing 122 fault categories, developed through the ChatGLM3-6B model completing named entity and knowledge relation extraction to generate fault knowledge and build a paraphrase vocabulary. (3) An intelligent fault information retrieval system that significantly outperforms traditional models in NEV-specific Q&A scenarios, providing multi-level fault cause analysis and actionable solution recommendations.

Keywords: new energy vehicle fault; intelligent retrieval system; generative language models; knowledge graphs; retrieval-augmented generation

1. Introduction

Currently, China is becoming a global leader in the research, development, production, sales, and maintenance of NEVs, maintaining the world's largest number of NEVs for several consecutive years. Along with the continuous evolution of digitalization and intelligent technologies in NEVs, a large number of new platforms and new vehicle fault evaluation indicators are also emerging. This rapid evolution is bringing about a series of new challenges, such as the complexity of functional layouts and configurations in NEVs, the reliability and stability of charging infrastructure, and consumers' insufficient understanding of potential fault causes, manifestations, and consequences in NEVs.

Due to significant differences in the technology of NEVs compared to traditional gasoline vehicles, and the complexity of the three-electric system, consumers often struggle to cope with the integrated and complex nature of NEVs. These issues lead to difficulties for ordinary consumers in addressing complex usage situations, fault maintenance, and relevant policy regulations. Currently, fault diagnosis technologies in this field still face the following challenges: (1) insufficient data, fragmented information, and difficulties in acquiring knowledge. Current fault diagnosis technologies face challenges such as insufficient data, fragmented information, and difficulty in acquiring knowledge. Traditional fault diagnosis methods rely on large amounts of fault data to establish signal features or model parameters, which is extremely difficult for the diversified range of NEVs, at least at the current stage. Furthermore, fault data from different brands and models of

 doi:10.20944/preprints202503.1268.v1

NEVs is difficult to unify. In this context, using system knowledge or rules to construct expert systems for fault reasoning presents challenges due to incomplete knowledge systems, fragmented and unstructured data and information, incomplete knowledge representation, and difficulties in updating new knowledge. (2) Information fragmentation and low integration in the NEVs field. In recent years, significant breakthroughs have been made in knowledge reasoning and intelligent generation through LLMs and KGs. However, in the specific and rapidly developing field of NEVs, there is still a lack of well-organized and comprehensive fault information and knowledge. Problems such as severe information fragmentation and low integration remain widespread. (3) Challenges in building specialized LLMs and knowledge bases for NEVs. Developing specialized LLMs and knowledge bases for NEVs faces many difficulties. In the context of information fragmentation and incomplete knowledge system structures, intelligently training models and generating knowledge requires breakthroughs in methods and technologies.

To address these issues, this study proposes a method that combines fault tree analysis with an expert system for fault diagnosis. The knowledge base constructed through fault tree analysis not only contains the causal relationships of NEVs faults and solutions but also retains the unique hierarchical relationships between fault tree knowledge conclusions. By converting the fault tree into a knowledge structure combining KGs and rules, this method can effectively support reasoning and analysis in the fault diagnosis process. This method not only addresses current technological bottlenecks but also provides a sustainable solution for NEV fault diagnosis.

The main contributions of this study include: (1) proposing an innovative electric vehicle fault information retrieval system that integrates large-scale language models and KGs, capable of efficiently managing diverse and fragmented fault data; (2) developing a knowledge graph covering 122 different fault categories and using the ChatGLM3-6B model to promote efficient extraction and reasoning of fault knowledge; (3) creating a fault diagnosis system based on RAG technology that provides accurate fault cause analysis and offers feasible solution recommendations.

2. Current Trends and Functional Requirements Analysis of Intelligent Fault Information Retrieval Systems for NEVs

2.1. Domestic and International Research Trends

2.1.1. Fault Diagnosis and Information Retrieval Systems Research Trends

In recent years, the development of fault diagnosis technologies for automobiles and integrated electromechanical equipment has progressed rapidly and can generally be divided into three types: signal-based fault diagnosis [1], knowledge-based fault diagnosis[2], and model-based fault diagnosis [3]. The classification diagram of fault diagnosis technology is shown in Figure 1. With advancements in artificial intelligence technology, increasing attention is being paid to combining model-driven and data-driven approaches. For example, Qiao et al[4] applied stochastic resonance (SR) to mechanical fault detection. Unlike traditional signal processing methods, stochastic resonance utilizes noise embedded within signals to extract weak fault features, making it widely used in mechanical fault detection. Xiao et al[5] introduced a new expert system knowledge base formed by combining the dynamic fault tree model with the expert system knowledge base. Zheng et al[6] introduced a knowledge transfer strategy for cross-domain fault diagnosis, highlighting the application of transfer learning in constructing efficient fault diagnosis models. Jafari et al[7] proposed an online lithium-ion battery state estimation method based on the extreme gradient boosting (XGBoost) algorithm. This method uses voltage-time data from partial constant current stages as input and obtains a nonlinear relationship model through offline training, thereby improving the accuracy of battery state estimation in electric vehicle applications. Xia et al.[8] proposed a real-time fault detection and process control method based on multi-channel sensor data fusion. Using multi-channel sensor data fusion and uncorrelated multilinear discriminant analysis (UMLDA) to model multi-channel data, the method achieved superior monitoring and fault

 doi:10.20944/preprints202503.1268.v1

diagnosis performance compared to other approaches. Simulation and case analysis demonstrated the effectiveness of this method in rapidly detecting equipment faults. In reality, automotive automatic diagnosis and expert system fault diagnosis methods maintain high usage rates and have become fundamental means of fault diagnosis. With the development of network technologies, digital communication, and artificial intelligence, integrated fault diagnosis systems that combine networks and artificial intelligence are becoming the new trend leading the field of automotive fault diagnosis[9–11].

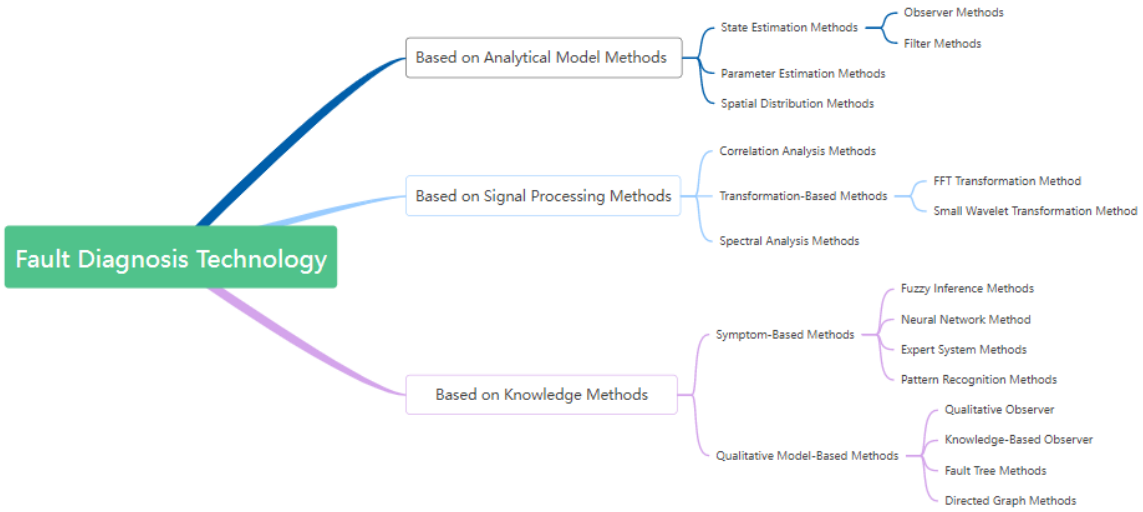

Figure 1. The classification diagram of fault diagnosis technology.

2.1.2. Knowledge Graphs and Generative Language Models Research Trends

(1) Generative Language Models

Since the introduction of the Transformer model in 2017, various models based on Transformer have emerged: The BERT model is primarily used for traditional NLP tasks, while the GPT model is mainly used for text generation tasks. Currently, various image processing methods based on Transformer architecture have emerged, making significant progress[12].

GPT (Generative Pre-trained Transformer), a generative pre-trained model based on the Transformer architecture, was proposed by OpenAI in 2018. It serves as the foundation for all LLMs and is a model capable of generating natural language text based on a given context. Transformer is a deep neural network architecture based on the attention mechanism, which can effectively capture long-distance dependencies in text and has high parallelism and scalability. The GPT models have made groundbreaking advancements in natural language processing and have demonstrated powerful capabilities in text generation tasks, such as generating coherent and diverse follow-up text[13–15].

Generative language models like GPT have shown broad application prospects in various fields, generating fluent, coherent, and logical text, and handling multimodal input of images and text. For example, GPT-4, as a large-scale multimodal model, can accept both image and text as input and generate text as output. Its performance surpasses previous language models and can complete difficult tasks across multiple fields. Furthermore, GPT-4 significantly outperforms existing language models in traditional natural language processing tasks, even though these systems are usually fine-tuned for specific task benchmarks[16,17].

(2) Knowledge Graphs and Applications

A knowledge graph is a technology that represents and organizes structured knowledge through a graphical model, containing rich semantic information between entities, relationships, and attributes. Its application is of great significance to expert systems, providing rich background knowledge and semantic associations to help answer user questions. The construction methods of

 doi:10.20944/preprints202503.1268.v1

KGs have always attracted attention. Traditional methods mainly rely on manual annotation or knowledge extraction from structured data. In recent years, methods based on natural language processing and machine learning have gradually emerged, enabling automatic extraction of knowledge from unstructured text data, improving the efficiency and coverage of knowledge graph construction.

Currently, the applications of KGs can be categorized into four main types: the first is methods based on question-answer templates, the second is research conducted through semantic parsing, the third is methods using deep learning to rank answers, and the last is research conducted using knowledge graph embedding learning. In recent years, the continuous progress of KGs and their deep application in various fields have made knowledge graph-based information systems a growing research focus.

(3) The Combination of Generative Language Models and KGs

Large models have demonstrated excellent performance in various natural language processing tasks, but they still have significant limitations when facing complex knowledge reasoning tasks, such as data biases, fairness issues, poor interpretability, transparency, and factual inaccuracies leading to model hallucinations. To address these problems, researchers are improving training algorithms, proposing new model architectures, using more fair and balanced datasets, increasing model transparency and interpretability, and developing more effective knowledge integration techniques[18,19].

A knowledge base can be a structured database, unstructured textual literature, or a knowledge graph's graph database. In the application of combining generative language models with KGs, besides embedding graph data into model training and enhancing models with graphs, the approaches shown in Table 1 are widely used.

Table 1. The integration methods of KGs and LLMs.

Method	Core Focus	Limitation	Reference Examples
Pure LLM Reasoning	Using the idea of linking, LLMs can solve some reasoning problems	Difficult to handle professional knowledge-intensive reasoning and complex reasoning tasks	ChatGPT4.0, ChatGPT3.5, ChatGLM, Wenxin-Yiyan, etc
LLM⊕KG	Large models play various roles, through querying knowledge in KG to enhance reasoning capabilities, enabling the addition of external knowledge into the model	Embedding LLM into KG limits its interpretability and introduces complexity when updating the knowledge base	Li et al. used LLM to generate SPARQL queries for KGs, and the main subject of the study is to complete KGs
LLM⊗KG	Cooperation between KG and LLM, enabling knowledge graph completion and reasoning through both LLM and KG integration	Need to consider the accurate path to knowledge graph completion while integrating external knowledge	Sun et al. used KG/LLM in the beam search algorithm for knowledge reasoning, enabling enhanced path expansion for KG generation[20]

2.2. Functional Requirements of Intelligent Fault Information Retrieval Systems for NEVs

NEVs are essentially complex electromechanical systems in motion. Their fault information retrieval systems must fulfill multiple functions, including fault information acquisition, knowledge-based Q&A on faults, and maintenance guidance.

 doi:10.20944/preprints202503.1268.v1

To fully understand the characteristics and patterns of fault information retrieval for NEVs, this study designed a questionnaire survey targeting different users and market sales personnel, focusing on clarifying the following key issues.

(1) Basic user needs: This includes understanding the aspects of NEVs-related fault knowledge that different users prioritize; whether they have reliable methods and resources for obtaining information on NEV usage, maintenance, and upkeep; their preferred channels for accessing information when faults occur and the reliability of these channels, such as whether they can help users promptly identify and locate issues; users' needs for rapid fault diagnosis tools and their trust in maintenance suggestions. Additionally, it investigates the performance of existing NEV mobile applications and automotive apps in meeting users' daily fault information retrieval needs, and whether these platforms and applications can provide users with fast and convenient services.

(2) User preferences for fault information retrieval: For example, whether users prefer using search engines to find possible fault causes and solutions; their awareness of and specific requirements for professional intelligent search systems; whether traditional automotive fault diagnosis systems are difficult to use or require high technical expertise; the common challenges users face in accessing NEVs fault data, and their systematic needs for fault information retrieval.

(3) Requirements for system functionality: Whether existing fault diagnosis expert systems can provide rapid analysis of fault information for different NEVs; whether they meet the needs of users who lack professional knowledge; whether the knowledge base of NEVs faults within these systems is comprehensive; whether existing automotive mobile applications support fault consultation and provide problem-solving suggestions; whether these applications present NEVs faults and solutions in an intuitive and user-friendly way; and whether the system platforms suffer from issues such as overly complicated interfaces, inefficiency in responding to user inquiries, or low responsiveness that negatively impact the user experience.

Through user surveys, literature analysis, and an evaluation of existing automotive client applications, it has been found that current fault information retrieval systems for NEVs face significant challenges in meeting the diverse needs of different users. The primary issues include the following:

(1) The user interfaces of existing automotive fault diagnosis expert systems are overly complex and crude, and there is limited information specifically related to NEVs.

(2) Most systems are unable to conduct precise reasoning and analysis of automotive faults. The fault diagnosis technology they rely on requires a large amount of fault data to establish signal features or model parameters. For the vast variety of NEVS currently available, this is difficult to achieve under current conditions. Moreover, it is challenging to unify fault data for different models of NEVs. While knowledge-based reasoning methods do not require extensive fault data, they also face certain difficulties. For example, they can only perform fault detection within the scope of existing knowledge, and acquiring new knowledge is challenging, with incomplete representation of knowledge being another issue.

(3) Existing automotive fault diagnosis systems are cumbersome to operate. Each vehicle model requires individual analysis, and the interfaces of both client applications and web pages are disorganized and cluttered.

Table 2 summarizes the problems and requirements of the system. By addressing the existing issues at various levels of these requirements, the direction for the construction of the NEVs fault data retrieval system has been determined.

Table 2. Survey of existing system problems.

Serial Number	Existing Problems in the System	Brief Description	System Requirements	New System Requirements
1	Information search is inconvenient	The client-side web page refreshes, the	Customer-side retrieval is complicated, and	The interface should be concise, user-friendly, and

 doi:10.20944/preprints202503.1268.v1

		interface is not user-friendly, and there is no focus on NEV fault optimization	the interface is inconvenient	able to quickly locate the required functions
2	No specific model for vehicle faults	There is no specific fault diagnosis system for vehicles	Technical method	Needs a fault detection model for vehicle fault diagnosis, providing precise and natural interaction for troubleshooting
3	Lack of relevant knowledge	NEV fault knowledge is scattered and unorganized	Data acquisition	Needs to organize knowledge on NEVs faults and establish a knowledge base with feedback.
4	Data resource scarcity	Needs to build a knowledge base for NEV faults	Knowledge base	Needs to build a specialized knowledge base for NEVs with capabilities for easy integration and expert use

To address the above issues, an intelligent retrieval system for NEV fault information must achieve breakthroughs in the following areas:

(1) Data acquisition: intelligent methods should be used to construct a dataset specifically focused on the domain of NEV faults; additionally, a large amount of unstructured and semi-structured data scattered across different sources and literature should be systematically integrated.

(2) Knowledge database layer: current fault diagnosis expert systems based on fault tree analysis methods lack quick and convenient maintenance and knowledge acquisition capabilities; automotive-related websites and applications, such as Autohome and DCar, provide users with basic data like vehicle configuration parameters but fail to deliver sufficient data to comprehensively integrate knowledge in the field of NEVs. Therefore, it is necessary to construct databases and presentation methods tailored to the characteristics of NEV knowledge data. This process should involve maintaining data by model, based on common fault types and phenomena, to ultimately form a comprehensive knowledge set for NEVs faults.

(3) Technical methods: current automotive fault diagnosis technologies are unable to perform precise reasoning and analysis for NEV faults, as they rely heavily on large amounts of fault data to establish signal features or model parameters, which is difficult to achieve given the relatively short time NEVs have existed and their diverse range of models. Although knowledge-based reasoning methods do not require extensive fault data, they have limitations, such as their inability to detect faults beyond the existing knowledge base, difficulties in acquiring new knowledge, and incomplete knowledge representation. To address these challenges, it is necessary to employ artificial intelligence, expert system methods, and knowledge graph technologies, such as RAG for LLMs, Prompt engineering[21,22], fine-tuning of LLMs[23], BERT classification algorithms, Cypher query language, front-end visualization, and API streaming for LLMs. Through the systematic integration of these technologies, an intelligent fault diagnosis and analysis platform for NEVs can be established.

(4) System presentation: traditional fault diagnosis expert systems have relatively simple user interface designs, while existing automotive website platforms suffer from overly complex and cluttered page information, with their main content dominated by conventional automobiles and related data. Additionally, pages designed for user feedback on fault issues are poorly structured, making information retrieval difficult. To resolve these issues, new technologies should be implemented, such as the Flask framework combined with visualization techniques, to develop a fault data retrieval system specifically for NEVs. This system should enable intuitive visualization of

 doi:10.20944/preprints202503.1268.v1

the NEVs fault knowledge graph, providing users with a more user-friendly and convenient service for fault diagnosis and information retrieval.

3. System Design of Intelligent Retrieval of Fault Information

3.1. The Logical Structure Design of the Fault Data Retrieval System

3.1.1. The Integration of Fault Diagnosis Technology

The Fault Tree Analysis (FTA) based on the Fault Tree Model and the Fault Diagnosis Expert System are two important branches in the field of fault diagnosis. Exploring their integration to meet system requirements.

(1) Fault Tree Analysis

Fault Tree Analysis (FTA), also known as Causal Tree Analysis, combines the advantages of qualitative analysis, rule-based reasoning, and quantitative analysis, making it a simple yet effective analysis method. It models the fault object as a fault tree, identifying and determining all possible patterns that could lead to the top-level fault, and is a powerful tool for reliability analysis.

The qualitative analysis in FTA focuses on evaluating the importance of events, i.e., analyzing the extent to which each event influences the occurrence of the top-level fault, and based on this, preventive measures are formulated. In a fault tree model, the occurrence of the top-level event may be determined by all the underlying events. However, in practice, the occurrence of the top-level event is often caused by a subset of the underlying events (cut sets). When the removal of any event in the cut set results in the non-occurrence of the top-level event, the cut set is referred to as a minimal cut set. The identification of minimal cut sets in the fault tree reveals various patterns of top-level event occurrences, highlighting the core role of minimal cut sets in fault tree analysis.

(2) Fault Diagnosis Expert System

A fault diagnosis expert system is an integrated system that combines artificial intelligence technology, aiming to simulate human experts in system fault data retrieval and diagnosis. These systems mainly achieve accurate and rapid diagnosis of complex system faults by integrating components such as specialized knowledge bases, reasoning mechanisms, and user interfaces. By combining expert knowledge with computer technology, a fault diagnosis expert system is constructed. This system follows specific reasoning algorithms and facilitates communication between the user and the computer through a human-computer interaction interface. The user needs to answer the questions posed by the system, and the system performs logical reasoning based on these questions and answers, ultimately deriving and presenting the diagnostic conclusion.

(3) Integration of Fault Tree Analysis and Fault Diagnosis Expert System

Through comparative analysis of the diagnostic expert system and fault tree analysis, commonalities between them can be identified. First, the fault tree can be seen as a fault model within the fault diagnosis expert system. The reasoning process of the expert system and the logical structure of the fault tree have similar designs. Additionally, from the perspective of knowledge acquisition, the fault tree represents a standardized knowledge framework. Using this framework to construct the knowledge base of the diagnostic expert system not only clearly outlines the solution path to the diagnostic problem but also significantly reduces the difficulty of knowledge acquisition for the system.

The knowledge base constructed based on fault tree analysis contains not only the rich experience and causal relationships between faults in NEVs but also preserves the unique hierarchical relationships between the conclusions in the fault tree. Transforming the fault tree into a knowledge structure that combines a knowledge graph framework with rules, effectively supports reasoning and analysis in the fault diagnosis process. This method not only facilitates the acquisition and simplification of fault diagnosis knowledge but also improves the efficiency and accuracy of the diagnostic process, providing technical support for the fault data retrieval system in this research on NEVs.

 doi:10.20944/preprints202503.1268.v1

3.1.2. The Logical Integration of Large Language Models and Knowledge Graphs

Although LLMs have demonstrated excellent capabilities in natural language processing tasks, they exhibit limitations when faced with complex knowledge reasoning tasks that require deep and responsible reasoning. First, LLMs often cannot accurately answer specialized knowledge questions that go beyond their pre-trained content, especially those that rely on outdated knowledge or require long logical chains or multiple knowledge jumps. Second, the shortcomings of LLMs in responsibility, interpretability, and transparency increase the risk of generating erroneous information or harmful text. Finally, the training process of LLMs is not only costly and time-consuming but also presents a continuous challenge in keeping their knowledge up to date[24].

When using pure LLMs with chain-of-thought prompts to answer questions about the common faults of specific models of NEVs, the model typically fails to provide accurate answers beyond its pre-training phase. Additionally, it shows a lack of common sense reasoning and insufficient handling of semantic ambiguities. Given the limitations of relying solely on the model for reasoning, introducing fine-tuning training and KGs as external knowledge sources becomes a natural and promising solution, aimed at enhancing the model's reasoning ability.

In Figure 2, the response using only the large language model is shown, while in the example in Figure 3, the model plays the role of "translation" + "reasoning." It transforms the input question into machine-readable commands for the graph to perform search and reasoning. The complete knowledge graph successfully retrieved the correct knowledge, correcting the fault reasoning model that was fine-tuned with downstream learning on electric vehicle failure knowledge. Due to the limited and non-updatable knowledge base, it had led to a lack of fault knowledge, but the model provided reasonable fault data retrieval results.

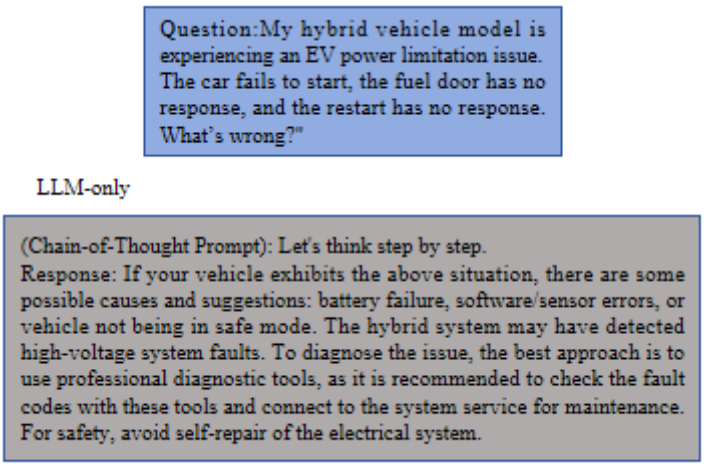

Figure 2. Only using LLMs.

 doi:10.20944/preprints202503.1268.v1

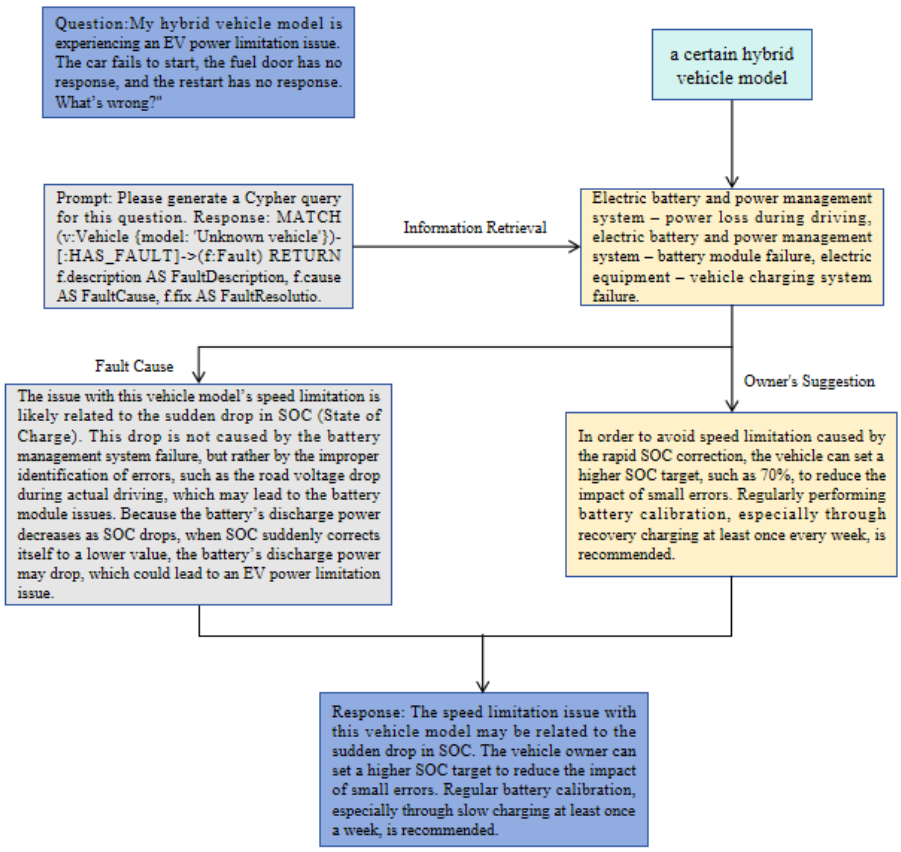

Figure 3. LLMs combined with knowledge graph retrieval.

3.2. The Framework Design of the Fault Data Retrieval System

The primary goal of developing an intelligent retrieval system for NEV fault data is to provide support for various users in aspects such as fault-related information, intelligent diagnostic analysis, fault repair and maintenance, equipment operation, and comprehensive knowledge services. Therefore, the overall design of the system requires a high level of fault information retrieval capability and sufficient knowledge capacity.

This study's intelligent retrieval system for NEV fault information uses a large language model as the reasoning engine and a knowledge graph as the knowledge base to analyze fault information and provide diagnostic and retrieval results. Its key features include leveraging the powerful reasoning and generation capabilities of LLMs, utilizing fine-tuning transfer learning techniques to handle and complete specific downstream tasks, and integrating the Neo4j database to store knowledge triples related to NEVs faults. Based on the fault knowledge model, the system provides fault information retrieval and analysis for users.

During the system's operation, when a vehicle experiences an abnormality, the user can input fault-related information through the front-end interface. After acquiring this information, the system processes it using a fault classification model to preliminarily determine the fault type. Depending on the user's specific situation, the system selects an appropriate query template and generates a corresponding Cypher query statement. It then performs query operations in the knowledge graph database, retrieves the results, and feeds them back to the fault reasoning and analysis model. The large language model performs fault reasoning on the fault information, determines the fault condition, and provides relevant suggestions, completing the entire fault data retrieval and diagnostic process.

The framework structure of the intelligent retrieval system for NEV faults data includes a large language model module, a knowledge graph module, an integrated reasoning module, and a human-computer interaction module, as shown in Figure 4.

 doi:10.20944/preprints202503.1268.v1

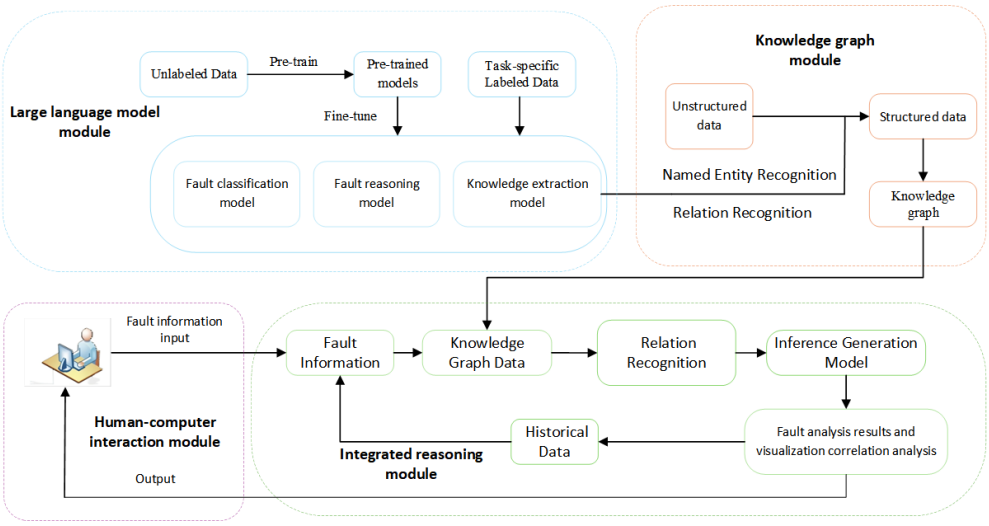

Figure 4. System design framework diagram.

3.2.1. Large Language Models Module

The LLM module consists primarily of two parts: data processing and the training and fine-tuning of the LLM. This module serves as the foundation of the entire system. Fine-tuning is employed in the design to train the model with knowledge related to various fault issues in NEVs, which involves three main stages: data acquisition and preprocessing, model training, and training evaluation. The LLMs module is shown in Figure 5.

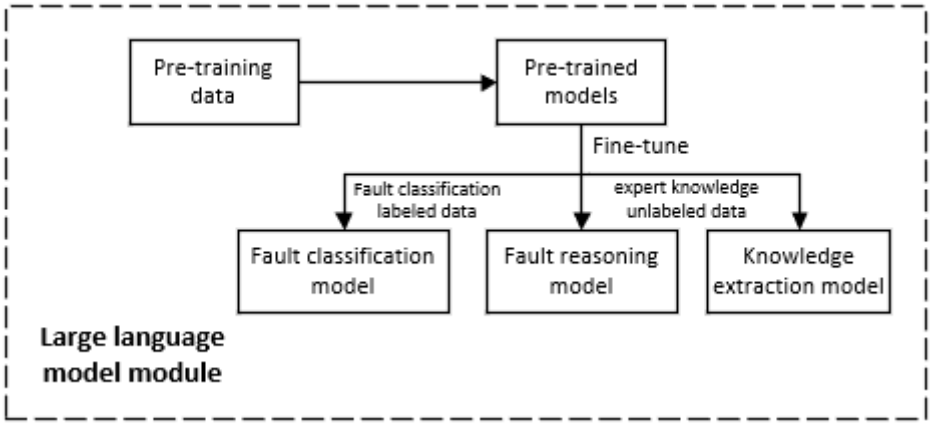

Figure 5. The LLM module.

(1) Data acquisition and preprocessing

The primary data sources for this study include websites related to NEV faults, enterprise design and production materials, as well as local and national standards. A differentiated strategy is adopted for data acquisition: semi-structured data is collected using Python and its associated toolkits, while unstructured data is obtained via text parsing and OCR recognition[25]. Due to differences in website structures and scanning limitations, the collected data may contain invalid or incorrectly formatted entries. Therefore, preprocessing is necessary, which includes data format conversion and filtering, primarily achieved using regular expressions and custom scripts. This ensures that the data is adequately prepared and valid, providing strong support for subsequent fault retrieval and analysis.

(2) Model training

The purpose of model training is to enable the model to deeply learn domain-specific data, allowing it to accurately identify and analyze fault information for NEVs. By comparing different training strategies, the best training process for handling NEV faults can be identified. In this study,

 doi:10.20944/preprints202503.1268.v1

DeepSpeed acceleration technology and CUDA GPU computation are used to improve training efficiency. During experiments, parameters such as learning rate, batch size, and training epochs are continually adjusted to optimize model performance. The training process leverages the Seq2SeqTrainer from the Transformers library for efficient model training. Once training is complete, the model is saved, and key performance metrics, such as loss variations and accuracy, are recorded for future evaluation and application.

(3) Model evaluation

The Model evaluation primarily focuses on assessing the system's performance in intelligent retrieval of NEV fault information, particularly in expert Q&A tasks. The model is tested using existing Q&A datasets, and its performance is measured by calculating BLEU and ROUGE scores, which evaluate text similarity and semantic similarity between the model's responses and standard answers[26]. After considering these metrics, the model that performs best in terms of text similarity, semantic similarity, and logical coherence is selected as the final system fault knowledge model.

3.2.2. Knowledge Graphs Module

To establish an intelligent system that can truly solve the problem of NEV fault information retrieval, it is necessary to build a database of NEV faults and a relatively complete knowledge system. The realization of this function requires the knowledge body graph module. In the study, a combination of top-down and bottom-up methods was used to construct a knowledge graph of NEV fault information; on the basis of data preprocessing, the function was realized through knowledge modeling, knowledge extraction and knowledge storage[27,28]. The framework of the knowledge graph module is shown in Figure 6.

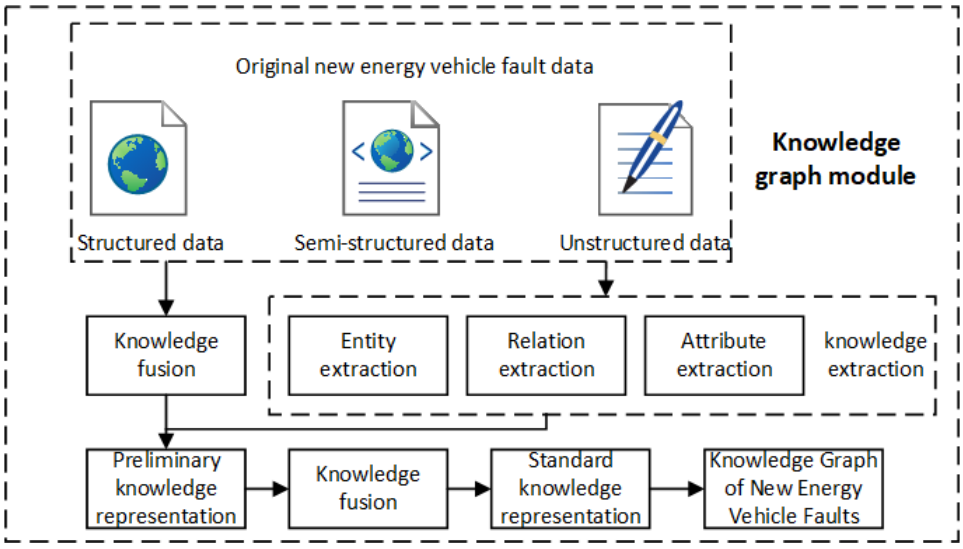

Figure 6. The knowledge graph module.

(1) Knowledge modeling

This stage focuses on building the ontological framework structure for NEV fault issues. Through extensive research into real-world scenarios and data analysis, and based on the diagnostic characteristics of the fault tree model, the knowledge system for NEV faults is structured and layered. Relevant categories, attributes, and relationships are defined. Ontology modeling tools like Protégé are used[29], and an improved seven-step top-down method is applied to establish the ontology database for NEV faults.

(2) Knowledge extraction

This step involves extracting entities, relationships, and attributes from preprocessed data to construct and enrich the systematic knowledge graph for NEV faults. It primarily focuses on two types of triples: (entity, relationship, entity) and (entity, attribute, attribute value). To enhance the efficiency of knowledge extraction for fault information, extraction techniques are chosen based on

 doi:10.20944/preprints202503.1268.v1

the data structure. For semi-structured data, entities and attributes are extracted using rule-based scripts; for unstructured data, deep learning models are employed to extract entity relationships[30,31].

(3) Knowledge storage

Currently, knowledge storage methods can be categorized into table-based and graph-based storage methods. Due to its strong expressiveness, ease of understanding, scalability, and ability to facilitate cross-domain knowledge integration, this study adopts a graph-based storage method using Neo4j[23]. This enables the storage, subsequent updates, and reasoning of knowledge related to NEVs.

3.2.3. Integrated Reasoning Module

The core function of the integrated reasoning module is to correctly interpret user intent, classify faults using a fault classification model, retrieve related fault data from the knowledge graph database, and use the LLM for centralized reasoning. This enables intelligent fault data retrieval and provides actionable recommendations. The system adopts the Model-View-Controller (MVC) architecture and is developed using Python's Flask framework. It includes the data layer, algorithm layer, business layer, and presentation layer. Various tools, including Flask, Python, Neo4j, MySQL, ECharts, PyTorch, and Transformers, are employed in its design and implementation. This module consists of three main parts: fault classification, information retrieval, and model reasoning. The integrated reasoning module is shown in Figure 7.

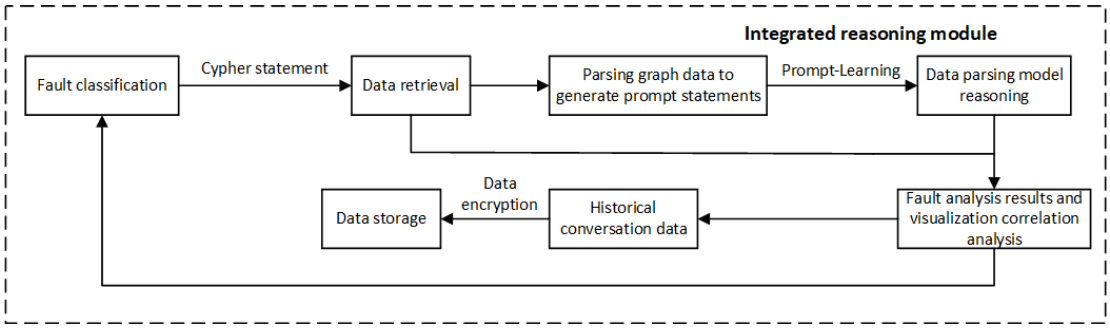

Figure 7. The integrated reasoning module.

(1)Fault classification

The fault classification module is designed to interpret user intent and preliminarily determine fault categories based on the information provided, laying the foundation for subsequent information filtering. This relies on fault classification models and focuses on mainstream classification techniques such as deep learning, machine learning, and pre-trained Transformer-based models[32]. Fault data for NEVs is annotated and applied to various classification models; the performance of these models is compared to select the one that achieves the best classification performance.

(2)Information retrieval

This module converts the fault information output by the fault classification model into specific Cypher query statements, which are then executed on the Neo4j knowledge graph database. The retrieved data is transformed into a predefined format for Prompt statements, which are then submitted to the knowledge model for NEV faults for comprehensive judgment. The key to system design lies in mastering Neo4j's Cypher query language and understanding the Prompt format requirements of LLMs to ensure the correct generation of query prompts. Finally, the results are outputted through the LLM.

(3)Fault reasoning

This module uses the interactive API of the LLM to process the retrieved Prompt information and submit it to the fault knowledge model trained with expert knowledge. It identifies the causes of faults in NEVs and provides corresponding recommendations. Simultaneously, the retrieved data is

 doi:10.20944/preprints202503.1268.v1

presented in the form of graph animations, visualizing the knowledge graph and the entire reasoning chain.

Method	Category	Accuracy P%	Recall Rate R%	F1 Score %
BERT	Braking System Fault	87.22	88.86	88.03
	Electric Drive System Fault	85.68	86.24	85.96
LSTM	Braking System Fault	81.14	82.59	81.86
	Electric Drive System Fault	84.27	85.61	84.93
TF-IDF + Naive Bayes	Braking System Fault	71.42	72.79	72.10
	Electric Drive System Fault	66.34	71.66	68.90
TF-IDF + Ridge Regression	Braking System Fault	73.19	72.59	72.89
	Electric Drive System Fault	77.15	79.60	78.36

4. System Implementation and Testing

The fundamental requirement for constructing an intelligent retrieval system tailored for NEV fault data is to accumulate sufficient vehicle fault data and knowledge including related concepts, methodologies, instrument tools, logic and information resources related to dealing with fault problems. The principal challenge encountered in practical application is how to obtain a sufficient quantity of pertinent data and knowledge resources, followed by their systematic organization and amalgamation to form a constantly enriched, holistic, and comprehensive knowledge architecture.

4.1. Data Acquisition and Preprocessing

The data sources mainly include text, audio, video, and other digital resources on the internet. In the study, we accessed various available resources, websites, and online databases related to automobiles through manual statistics and web crawler technology. After analyzing the structural features and data feedback methods of various websites, we systematically merged and integrated NEV data, which served as the basis for constructing the NEV knowledge graph and intelligent fault data retrieval system.

The dataset pertaining to NEVs encompasses three primary categories: battery electric vehicles, plug-in hybrid electric vehicles, and range-extended electric vehicles. This comprehensive dataset comprises essential details regarding the specific specifications and technological types of the existing NEVs. A cumulative total of 822 distinct vehicle models has been collated, which includes 631 battery electric vehicle models, 35 range-extended electric vehicle models, and 171 plug-in hybrid electric vehicle models. Moreover, an extensive compilation of 25,000 fault types has been compiled into 12 principal classifications, with further subdivision into 122 subcategories based on diverse fault characteristics. Additionally, an impressive repository of 22,709 instances of expert knowledge has been meticulously sorted out under various fault scenarios, providing a robust foundation for the preprocessing of data and the generation of knowledge for LLMs.

In the application of LLMs, it is imperative to undergo both training and fine-tuning phases, particularly in the context of QA-type applications, wherein the objective is to facilitate effective

 doi:10.20944/preprints202503.1268.v1

question-answering. The process of fine-tuning is systematically executed in accordance with the following steps:

Step 1: Data sanitization, which entails the meticulous purification of the original dataset by eliminating special characters, punctuation, HTML tags, etc., thereby guaranteeing the integrity and uniformity of the information.

Step 2: Data format conversion is the transformation of the sanitized data into a JSON format that is conducive to model ingestion, typically structured in an interactive question-answer paradigm. Each data sample comprises a query and a corresponding response, delineated by specific delimiters. An example of the data format conversion is as follows: question,{"conversations": [{"role": "user", "content": "Throttle Position Sensor, Structure and Working Principle."}, "answer", {"role": "assistant", "content": "Greetings! The throttle position is situated posterior to the hose and antecedent to the manifold. The throttle constitutes a regulatable valve that governs the inflow of air into the engine. Subsequent to the air's entry into the intake manifold, it is combined with gasoline to create a flammable mixture, which is thereafter ignited to generate power."}]}

Step 3: Insertion of demarcation tokens, to apprise the model when it should start processing the problem and the corresponding answer. Special tokens, including [gMASK]、 sop、<|system|>、<|assistant|>and others, are interspersed at the commencement, conclusion, and specific junctures within the input text, thereby facilitating the model's accurate comprehension of the input data.

Step 4: Encoding of data, entailing the conversion of the annotated text into a recognizable code format for the model. This process employs word embedding techniques to transform textual data into vector representations and supplements them with positional encoding to constitute the model's input vector.

Following these preprocessing procedures, the primal dataset is meticulously converted into an appropriate format conducive to the fine-tuning of LLMs, subsequently undergoing meticulous annotation and encoding processes[33,34]. This measure can significantly enhance the efficacy of LLMs when applied to the downstream task of expert knowledge acquisition for NEVs.

4.2. New Energy Vehicles Fault Model Training

4.2.1. New Energy Vehicle Fault Knowledge Model Training

The core objective of fine-tuning LLMs is to achieve efficient adaptation of domain knowledge with limited resources. Traditional full-parameter fine-tuning requires updating all model weights, which is computationally expensive. To address this, parameter-efficient fine-tuning methods have emerged, which update only a subset of parameters or optimize prompts, significantly reducing resource consumption while maintaining model performance. This study focuses on two cutting-edge parameter-efficient fine-tuning methods—LoRA (Low-Rank Adaptation) and P-Tuning V2—to investigate which method is more suitable for training on electric vehicle fault data.

(1) P-Tuning V2 fine-tuning

P-Tuning V2 is an emerging fine-tuning strategy designed based on traditional text-based prompting, which, although effective in some cases, has limited expressive power due to its reliance on a fixed vocabulary and is unable to fully exploit the model's potential expressive space. Therefore, it enhances the model's performance in fault data retrieval by introducing learnable continuous vectors (referred to as "prompts" or "soft prompts") without directly modifying the pre-trained model's parameters. These continuous vectors are designed to be embedded into the model's input as task instructions, guiding the model to generate outputs for specific tasks. P-Tuning V2 optimizes these soft prompts rather than the model's weights, enabling adaptation to expert knowledge of NEV faults while keeping the pre-trained parameters unchanged. This approach is particularly suitable for scenarios with large-scale models in the current system, where directly fine-tuning the model would be computationally expensive[35].

(2) LoRA fine-tuning

 doi:10.20944/preprints202503.1268.v1

LoRA is a fine-tuning method designed for the Transformer architecture. It achieves fine-tuning by introducing additional, low-rank adaptation matrices into the model's self-attention and feed-forward networks[36]. The dimensions of these matrices are relatively small, significantly reducing the number of parameters that need to be updated during fine-tuning. The key idea of LoRA is to capture task-specific variations through low-rank matrices, rather than directly modifying the original model's weights. This method reduces the number of parameter updates during fine-tuning while maintaining high performance across various tasks.

(3) Experiment and results analysis

This study uses Loss and Accuracy in model training to predict and evaluate the performance of the model. Loss is calculated during the model training phase and is used to assess the difference between the model's predictions and the actual labels. The cross-entropy loss function is used to compute the difference between the predicted probability distribution at each time step and the actual labels. At the same time, BLEU and ROUGE scores are used to evaluate the quality of the generated text, with these scores calculated by comparing the generated text to the reference text, providing a comprehensive measure of the accuracy and fluency of the model's generated text.

In this experiment, data from 22,709 expert Q&A pairs were processed, filtering out question-answer pairs with a maximum output length of 256 tokens to ensure proper training for the LLM. The data was then split into training and testing sets at an 80/20 ratio, resulting in 18,107 training samples and 4,408 testing samples.

The model training experimental results are shown in Figure 8.

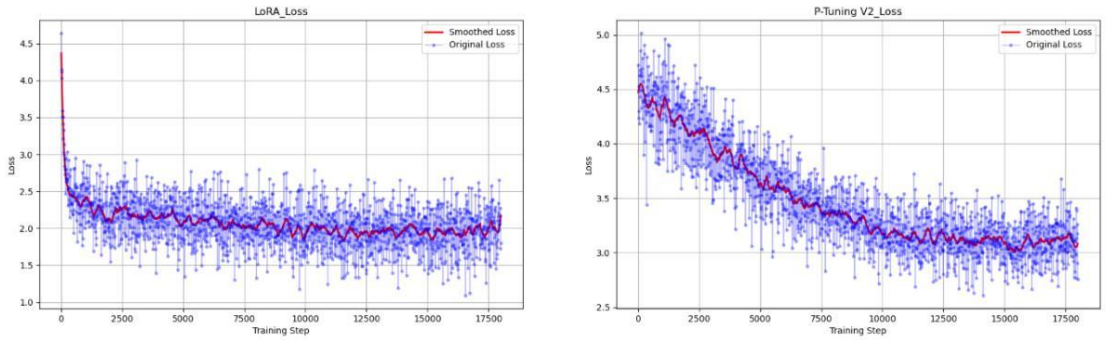

Figure 8. Model training loss chart.

From the experimental data, it can be observed that the model using LoRA technology shows a faster decrease in loss value during the training process compared to the model using P-Tuning V2. This indicates that the LoRA training method adapts more quickly to the NEVs fault expert knowledge training data, learning the complex relationships between various fault information. This rapid decline in loss value is evidence of good model training performance, suggesting that the model demonstrates high efficiency and accuracy in recognizing and understanding the training data.

In the model evaluation phase, the BLEU-4 and ROUGE-L scores are shown in Figure 9, with the BLEU-4 score on the left and the ROUGE-L score on the right.

The comprehensive experimental results show that both P-Tuning V2 and LoRA fine-tuning methods significantly improve the performance of the base model in the NEVs fault expert Q&A generation task. Specifically, P-Tuning V2 has a slight advantage in model input tokens, allowing it to train longer token sequences. On the other hand, LoRA performs better on large-scale datasets, demonstrating faster loss reduction, and its question-answering accuracy in BLEU-4, ROUGE-1, ROUGE-2, and ROUGE-L scores generally outperforms P-Tuning V2. The low-rank matrix update strategy of LoRA offers clear advantages in parameter efficiency and computational cost.

 doi:10.20944/preprints202503.1268.v1

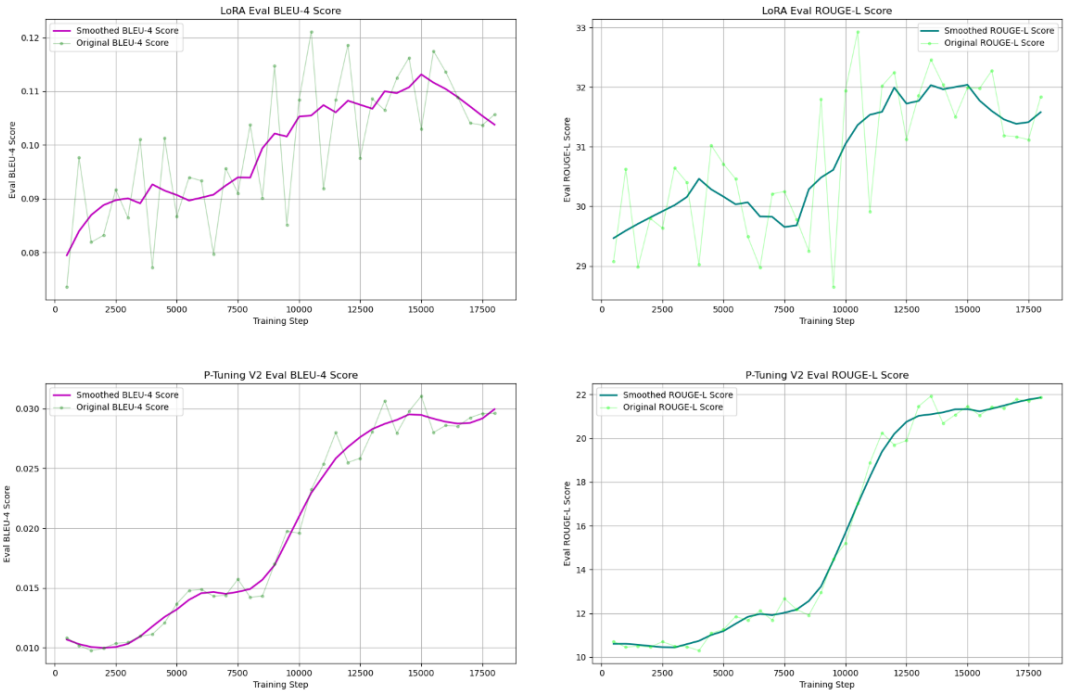

Figure 9. Model Training Evaluation Scores.

4.2.2. New Energy Vehicle Fault Classification Model Training

Since annotated data is sufficient in the field of fault classification, and BERT models often achieve better results in NLP tasks, considering the resource consumption of model deployment, we conduct a fault text classification experiment using Long Short-Term Memory(LSTM)and BERT. Additionally, TF-IDF + Naïve Bayes and TF-IDF + Ridge Regression classifiers are used as baseline comparison groups.

The study employs a cross-validation method with training and test sets, dividing the NEVs fault corpus into 10 parts, with 8 parts used for training and 2 parts for testing. The dataset is not shuffled. The deep-learning Transformers library provided by Hugging Face is used, and the pre-trained Chinese hfl/chinese-roberta-wwm-ext model from iFlytek is loaded for training. In the BERT-based text classification experiment, the tokenization and preprocessing step sets the maximum sequence length to 512 tokens. The experimental model includes a DistilBertModel as the pre-trained model, followed by a fully connected layer (768 to 768), a ReLU activation function, a Dropout of 0.2, and a final fully connected layer (768 to 12) for classification. The classifier layer parameters of the pre-trained model are used to initialize the final fully connected layer. The optimizer is set to AdamW with a learning rate of 2e-5. Dynamic padding is used to ensure uniform data length within each batch. During training and testing, data shuffling is enabled for training, and the last incomplete batch is discarded. The results of some fault classification algorithm experiments are shown in Table 3

Table 3. The results of some fault classification algorithm experiments.

Method	Category	Accuracy P%	Recall Rate R%	F1 Score %
BERT	Braking System Fault	87.22	88.36	88.03
	Electric Drive System Fault	85.68	86.24	85.96

 doi:10.20944/preprints202503.1268.v1

LSTM	Braking System Fault	81.14	82.59	81.86
	Electric Drive System Fault	84.27	85.61	84.93
TF-IDF+Naïve Bayes	Braking System Fault	71.42	72.79	72.10
	Electric Drive System Fault	66.34	71.66	68.90
TF-IDF+Ridge Regression	Braking System Fault	73.19	72.59	72.89
	Electric Drive System Fault	77.15	79.60	78.36

From the analysis of the above experimental results, it can be concluded that Na-ïve Bayes, due to its simplicity and efficiency, is usually suitable as a baseline model, while Ridge Regression performs better when dealing with fault data classification, especially in datasets with high feature correlation. Ultimately, the BERT model significantly outperforms both the deep learning LSTM model and the machine learning approaches, including TF-IDF + Naïve Bayes and TF-IDF + Ridge Regression, in classifying NEVs faults.

BERT optimizes the evaluation of the importance of different parts of an input sentence through its self-attention mechanism, enabling superior performance in handling long-distance dependencies and complex sentence structures. In the task of NEVs fault classification, BERT's fully bidirectional contextual understanding, strong generalization ability, and the advantage of reducing task-specific engineering requirements make it the preferred model for processing subtle textual differences. Based on the characteristics of the NEVs fault corpus and experimental results, this study selects the BERT model as the fault classification model for the NEVs fault data retrieval system.

4.3. Knowledge Graph and Fine-Tuning of Large Language Models

Through an in-depth analysis of the fault data procured from NEVs, a comprehensive knowledge graph pertaining to the failures encountered in NEVs has been meticulously designed and instantiated, leveraging the distinctive attributes of the automotive fault tree analysis methodology. The knowledge graph was systematically populated with specific fault categories pertaining to discrete vehicle models, encompassing 122 fault types along with their associated fault phenomena and etiologies. For LLMs, the ChatGLM3-6 model, engineered by Tsinghua University[37], was prudently chosen for the purposes of fault information retrieval and knowledge generation, in consonance with the unique characteristics of the problem at hand. An exhaustive extraction of fault information and knowledge relationships was extracted, culminating in the formulation of a comprehensive summary lexicon for the principal fault domains. Figure 10.delineates the procedural framework for constructing the fault knowledge graph tailored for NEVs.

 doi:10.20944/preprints202503.1268.v1

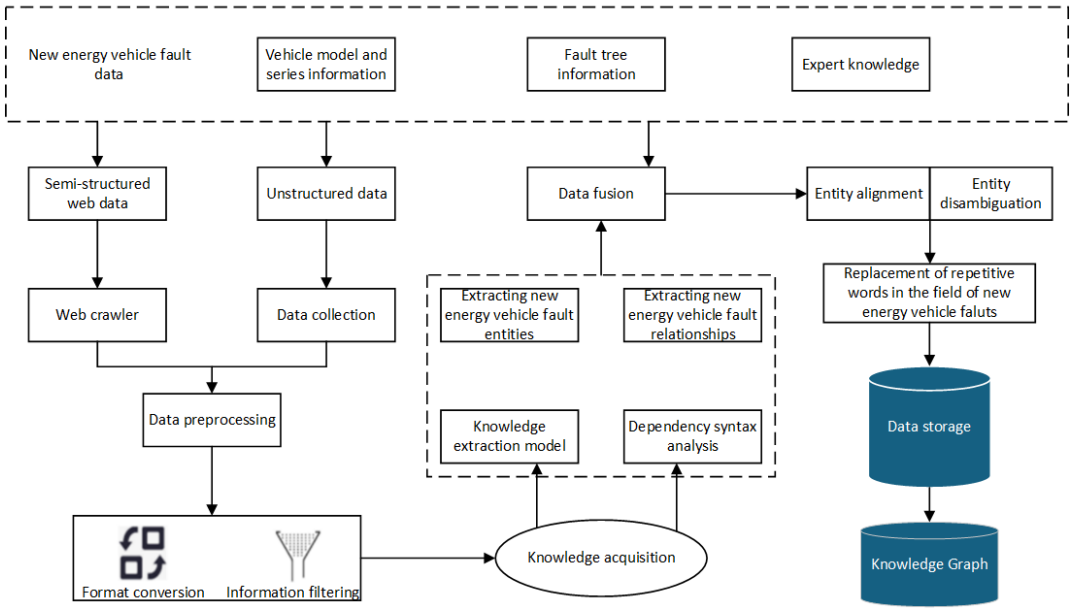

Figure 10. Fault Knowledge Graph Construction Process.

The training and fine-tuning of LLMs are indispensable in constructing sophisticated domain-specific intelligent information retrieval and question-answering systems. In the context of this investigation, the P-Tuning and LoRA fine-tuning techniques were selected for the fine-tuning and training of the ChatGLM3-6B model. Extensive experimental evaluations have demonstrated that both the P-Tuning V2 and LoRA fine-tuning methodologies considerably enhance the performance of the ChatGLM3-6B model when answering questions related to NEV malfunctions. In particular, P-Tuning V2 has an advantage in processing input tokens, enabling the concurrent training of longer token sequences. On the other hand, LoRA demonstrates superior efficacy on extensive datasets, showcasing enhanced accuracy metrics (Accuracy) in BLEU-4, ROUGE-1, ROUGE-2, and ROUGE-L for question-answering tasks. Its low-rank matrix updating approach is particularly advantageous in terms of parameter efficiency and computational expense. Owing to the intricate nature of the model calculations involved, a detailed analysis is not provided here. Figure 11 illustrates a comparative analysis of the output outcomes before and after the training of the LLM.

It is evident that the trained model effectively circumvents the fallacies illusions in the original model, manifesting a superior comprehension of NEVs fault knowledge, coupled with enhanced logical reasoning and analysis capabilities.

 doi:10.20944/preprints202503.1268.v1

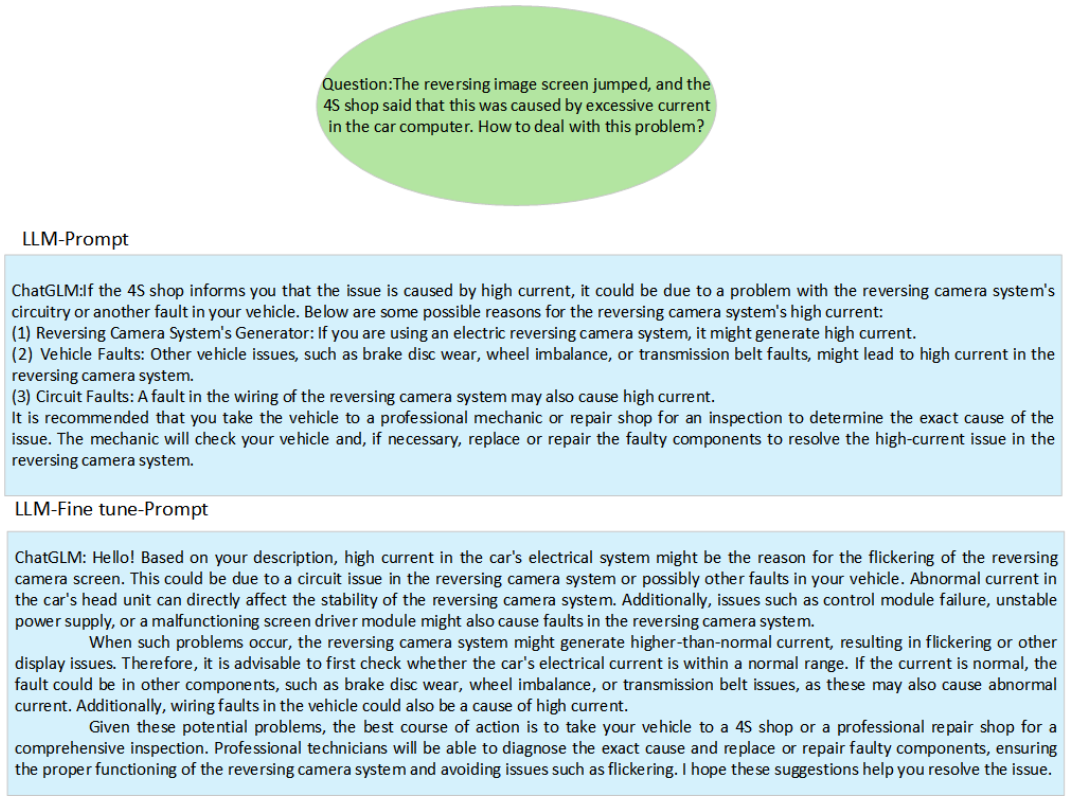

Figure 11. Comparison before and after model training.

4.4. The Process of Intelligent Retrieval of Fault Information of NEVs

In the typical pursuit of fault data retrieval, users are generally required to initiate the process by inputting a preliminary description of the fault-related information. Subsequently, the system engages in the identification and categorization of the input, ascertaining whether the information is fault-related or not. In the event that the input is identified as non-fault-related information, it is managed through the intervention of conventional, untrained fault knowledge-based LLMs, which provide pertinent services. Should the system deduce that the user's submission pertains to a fault issue associated with NEVs, the information is then routed to the intelligent fault data search service system. The fault classification model proceeds to classify the information, ascertaining the appropriate category from among the 12 predefined fault types. Upon establishing the general category of the fault information, the fault knowledge model extracts the specific type of vehicle fault from the fault knowledge graph via an information search. Following the correlation of the specific fault phenomenon, the fault knowledge model is tasked with delivering to the user a comprehensive search outcome detailing the fault cause and corresponding resolution, thus facilitating a logical, reliable, and coherent fault reasoning analysis. If deemed necessary, the system is also capable of transmitting the entire entity, attribute, and relationship information contained within the fault knowledge graph to the user interface, thereby elucidating the complete intelligent fault data search process. Figure 12 illustrates the intelligent search query process for fault information when the state of charge of an electric vehicle's battery is low, which is to say, the ratio of the remaining battery capacity to the total battery capacity falls below an acceptable threshold.

 doi:10.20944/preprints202503.1268.v1

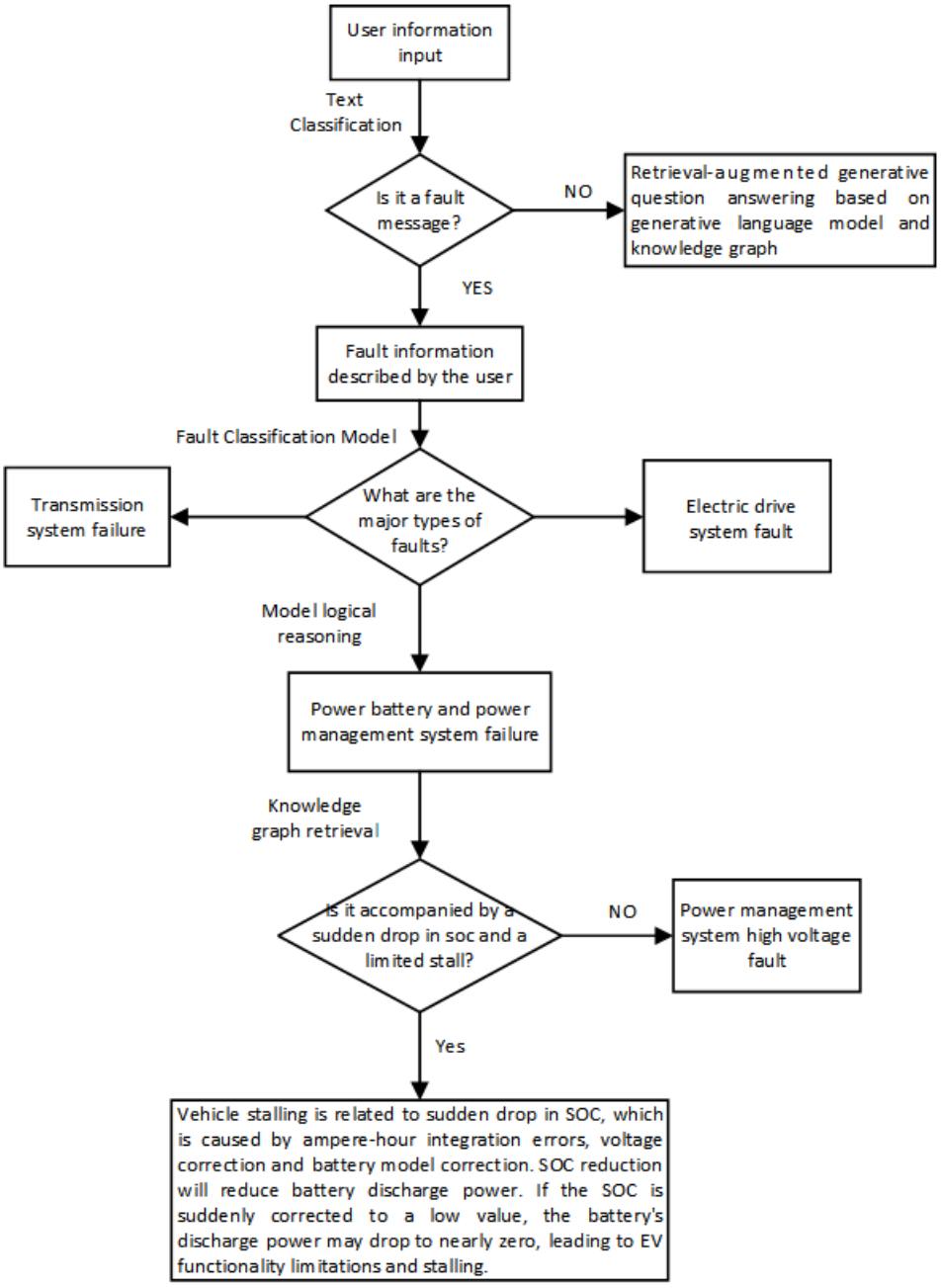

Figure 12. User fault data retrieval process.

4.5. Elaboration of the System Information Interaction Flow's Framework

In the functioning of the NEVs Fault Data Retrieval System, the primary dependence is placed upon a meticulously constructed knowledge base, derived from the automobile fault knowledge graph. This knowledge base facilitates the retrieval of pertinent fault data and their associated attribute relationships. Following sophisticated processing and refinement, the information is presented in the form of a Prompt to an LLM. Subsequently, leveraging the enhanced knowledge generation and reasoning capabilities of the LLM, which have been developed through rigorous training and fine-tuning, the system obtains the desired retrieval outcomes and formulates a proposed solution for the rectification of faults, as depicted in Figure 13.

 doi:10.20944/preprints202503.1268.v1

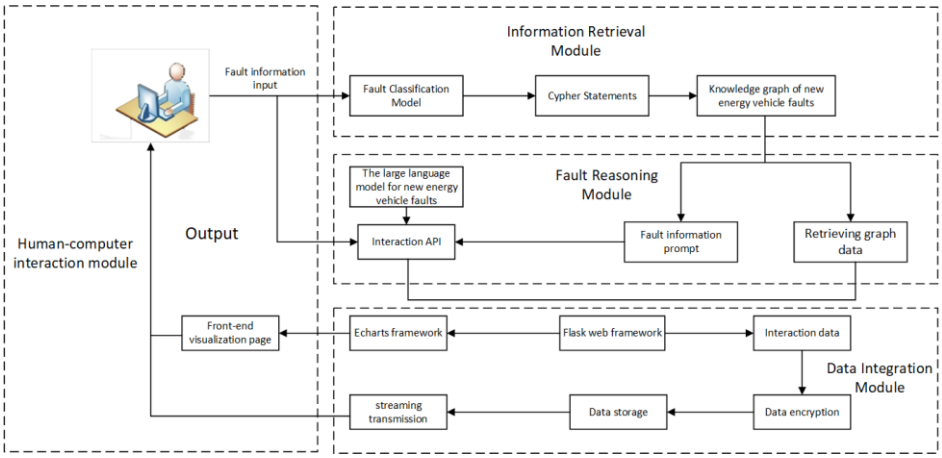

Figure 13. Fault information system interaction process.

In the initial phase, the system engages in the retrieval of fault-related information. This involves encoding the fault data of the user's NEVs, incorporating techniques such as truncation and padding to ensure compatibility with the input requirements of the model. Subsequently, the model's weights are loaded, and the encoded data is fed into the LLM for inferential processing, thereby yielding the desired output results. These output statements are then utilized to construct Cypher queries that align with the structure of the fault knowledge graph, enabling the retrieval of pertinent fault information from the Neo4j database.

During the second phase, fault data inference is conducted. To facilitate interactive services for users and to allow for the seamless integration of knowledge graph data into the interface, the system employs an API styled after OpenAI's format. This API extracts the relevant prompt from the knowledge graph and presents it to the model in a "role": "system" format. LLM then performs reasoning analysis to ascertain the cause of the fault in NEVs and to determine the appropriate countermeasures.

The third phase entails data integration. The system has the capability to encrypt the user's fault-related question and answer data for secure storage, and to feed back data that has received high user satisfaction to NEVs fault LLM for subsequent learning and enhancement of the model.

Finally, in the fourth phase, the system facilitates the display of fault information to the user by transmitting the data acquired from the API. Concurrently, the data sourced from the knowledge graph is visualized using the echarts framework. This approach enables the system not only to visualize the knowledge graph but also to achieve a graphical representation of the complete reasoning chain.

4.6. System Development Technology

The evolution of an intelligent search system tailored for fault diagnosis in NEVs has adopted a diverse array of development methodologies across multiple modules and hierarchical levels, as illustrated in Figure 14. On the database level, the Cypher query language is employed to execute read and write operations on the NoSQL database, Neo4j. Regarding the programming languages utilized in system development, Python, HTML, and CSS have been chosen for their efficacy. On the algorithmic design level, Python's robust data analysis libraries—Numpy, Pillow, and Matplotlib—are harnessed to facilitate model training, evaluation, and data processing. For the selection of deep learning models, the PyTorch framework is adopted, with Transformer serving as the model repository. In the domain of Web development, Echarts is utilized as the front-end data visualization framework, while Flask provides the structural foundation for the development environment.

 doi:10.20944/preprints202503.1268.v1

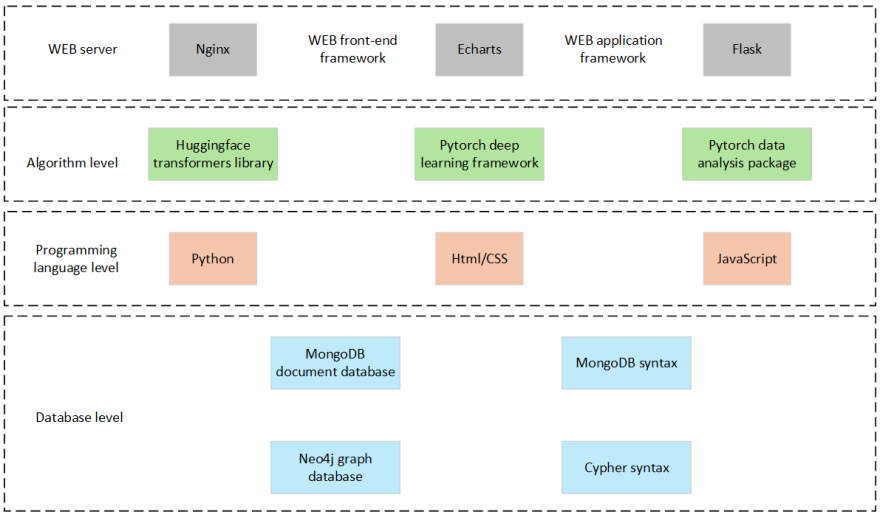

Figure 14. System development technical implementation.

4.7. System Demonstration and Testing

4.7.1. System Presentation

The retrieval mechanism embedded within the intelligent retrieval system tailored for fault data analysis of NEVs fundamentally encompasses two pivotal stages at its core: initially, a sophisticated classification algorithm is employed to conduct a detailed categorization of NEV malfunctions. Subsequently, advanced text analytics are utilized to distill critical information, such as the vehicle model and its series. Leveraging this information, the system diligently retrieves corresponding fault data and provides it as a Prompt to the reasoning model that incorporates expert knowledge and scavenges the knowledge graph for pertinent fault information, which is then fed into a reasoning model imbued with expert knowledge to generate a comprehensive fault data retrieval prompt. Ultimately, the system will progressively generate and display the entire fault data retrieval results and the knowledge graph visualization data (Figure 15).

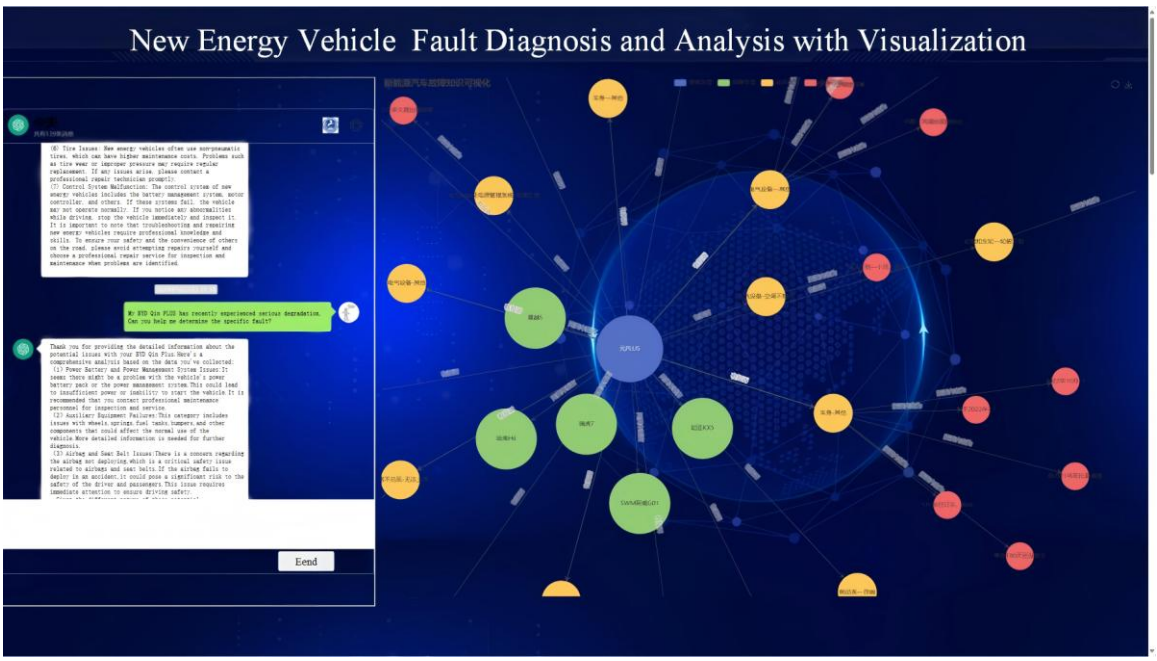

Figure 15. Fault diagnosis and analysis with visualization.

 doi:10.20944/preprints202503.1268.v1

4.7.2. System Testing

To ascertain the efficacy and operational efficiency of the system, a series of functional assessments and practical applicability evaluations were undertaken. Refer to Table 4 for a detailed breakdown of the testing conditions.

Table 4. System testing environment.

Environment	Parameter	Environment	Parameter
Operating System	Windows11 64-bit	Memory	32GB
CPU	Intel(R) Core (TM) i9-11900H CPU @ 2.50GHz (2502 MHz)	Graphics card	NVIDIA GeForce RTX3070 Laptop GPU 8G GDDR6

Throughout the system functionality testing phase, the primary focus is placed on whether the system developed can respond to the user's operations and fault information normally ascertaining the system's capability to adeptly respond to user inputs and to effectively relay fault notifications. Furthermore, it is crucial to evaluate the user's ability to interact with the system seamlessly, leveraging the intuitive nature of the interface display. The detailed procedural steps and corresponding outcomes are delineated in Table 5.

Table 5. System function testing.

Test items	Operating procedures	Expected results	Test results
User Interface Responsiveness Testing	Accessing the system interface via WEB pages	The system icons, buttons, and other user interface elements are displayed normally, and the user interaction functions are free of obstacles	Pass
Type of fault identification	Inputting fault information into the fault classification model in the background	The fault information inputted can be correctly received and parsed by the fault classification model.	Pass
Knowledge base search	The data parsed is retrieved from the knowledge base	The knowledge base can successfully retrieve data related to the parsed information	Pass
Fault data retrieval and graph visualization function	Submit the fault information after inputting it	The page can stream and analyze the data, and display dynamic visual effects of the fault information-related graphs	Pass

The assessment entails the utilization of a comparative analytical framework, comparing the performance of mainstream search engines and similar popular Question Answering Systems at home and abroad. Owing to the void in the intelligent search domain for specialized NEVs faults, there exists an absence of a mature application system; this research represents an exploratory endeavor, with the comparative entities primarily comprising widely-used intelligent question-answering systems such as Microsoft's Copilot, GPT-3.5 Turbo, Claude 3 Haiku, and the ERNIE Bot 4.0.

Throughout the assessment process, an emphasis is placed on scrutinizing the performance metrics of each system, including question-answer relevance, accuracy, textual logic, coherence, and response latency, as per queries crafted by automotive fault specialists. The assessment of question-answer relevance primarily gauges the pertinence and expertise of the responses proffered by the system in relation to the posited inquiries; precision evaluation pertains to the degree of alignment between the responses from the system and those from the automotive fault experts. In order to

 doi:10.20944/preprints202503.1268.v1

mitigate the bias inherent in comparing the efficacy of a particular knowledge model subsequent to targeted learning against its performance in an untrained state, the system assessment incorporates additional evaluation metrics, including textual logicality, coherence, and response latency, among others. These metrics serve to evaluate the logical consistency and coherence of the generated outputs, with the final response latency being quantified as the average generation time per token, where a shorter latency equates to a higher score. The scoring mechanism is predicated on the automated scoring capabilities of GPT-4, and the percentile system of artificial assessment.

The findings indicate that the intelligent detection system for NEV faults, developed as part of this study, is similar to LLMss in terms of targeted responses, has a slightly higher accuracy than all other systems, exhibits a deficiency in both coherence and logical progress, and its response latency marginally surpasses that of its counterparts (refer to Figure 16).

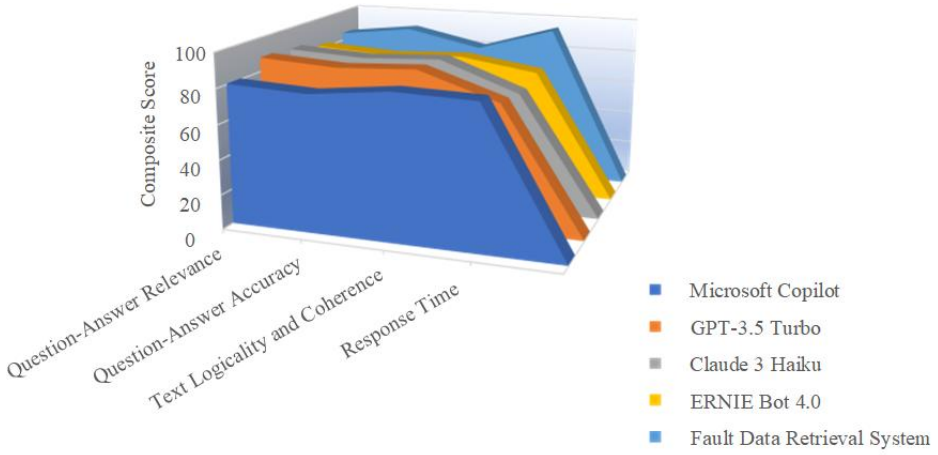

Figure 16. System usability assessment.

5. Conclusion and Prospects

In conclusion, the development of an intelligent retrieval system for fault information pertaining to NEVs has successfully met the predetermined objectives. On a comprehensive basis, the technical methodology employed, coupled with the execution of system functionalities, yielded successful outcomes. The substantial language model ChatGLM3-6B has been introduced, along with a dedicated downstream task dataset. The model has been meticulously fine-tuned using the LoRA methodology to cater specifically to the task of detecting faults in NEVs. Comprehensive experimental assessments have validated that the optimized model outperforms the baseline model in fault diagnosis tasks. During the research phase, fault data was meticulously gathered through web crawler and manual collection, followed by refinement and preprocessing, encompassing 122 common fault categories and associated diagnostic recommendations. The fault tree analysis method was amalgamated with the expert system for fault diagnosis, and the generative language model ChatGLM3-6B was integrated to facilitate the efficient processing of fault data and the construction of a knowledge graph. Based on this knowledge graph and the generative language model, an enhanced fault retrieval generation analysis algorithm has been meticulously crafted and executed, and an innovative intelligent search system tailored for fault data analysis in NEVs has been successfully developed, primed for initial practical deployment. Comparative analyses with analogous systems and models have validated the efficacy and sophistication of the NEVs fault data retrieval system.

Currently, there is still room for improvement in the system, and it will be further deepened and expanded from the following aspects:

In the domain of fault data retrieval for NEVs, significant enhancements are feasible in the areas of entity extraction, fault categorization, and relational identification. Particularly, during the inference and retrieval processes from graphical representations, the amalgamation of text-to-Cypher

 doi:10.20944/preprints202503.1268.v1

transformations with graph traversal algorithms can facilitate a multi-dimensional positioning and interactive approach, thereby augmenting the system's overall performance and precision.

Furthermore, the current rate of knowledge graph construction and updating falls short of optimal efficiency. Future studies should investigate the modalities for automating the construction and real-time refreshment of KGs, leveraging generative language models to dynamically recognize and assimilate novel knowledge, thereby preserving the currency and accuracy of the knowledge graph.

Lastly, it is imperative to amalgamate NEV fault data retrieval systems with other auxiliary systems, such as vehicular networking and maintenance decision support systems, to create an integrated and streamlined NEV service ecosystem. Prospective endeavors should focus on the exploration of inter-system data interchange and functional integration to furnish a more comprehensive and user-friendly fault diagnosis and repair service.

References

1. Cong, X.; Zhang, C.; Jiang, J.; Zhang, W.; Jiang, Y.; Zhang, L. A Comprehensive Signal-Based Fault Diagnosis Method for Lithium-Ion Batteries in Electric Vehicles. Energies **2021**, 14, 1221, doi:10.3390/en14051221.
2. Ahmad, I.S.; Abubakar, S.; Gambo, F.L.; Gadanya, M.S. A Rule-Based Expert System for Automobile Fault Diagnosis. Int. J. Perceptive Cogn. Comput. **2021**, 7, 20–25.
3. Zhang, Z.; Deng, Y.; Liu, X.; Liao, J. Research on Fault Diagnosis of Rotating Parts Based on Transformer Deep Learning Model. Appl. Sci. **2024**, 14, 10095, doi:10.3390/app142210095.
4. Qiao, Z.; Lei, Y.; Li, N. Applications of Stochastic Resonance to Machinery Fault Detection: A Review and Tutorial. Mech. Syst. Signal Process. **2019**, 122, 502–536, doi:10.1016/j.ymssp.2018.12.032.
5. Xiao, Y.; Han, F.; Ding, Y.; Liu, W. Research on Fault Diagnosis Method of Rapier Loom Based on the Fusion of Expert System and Fault Tree. J. Intell. Fuzzy Syst. **2021**, 41, 3429–3441.
6. Zheng, H.; Wang, R.; Yang, Y.; Yin, J.; Li, Y.; Li, Y.; Xu, M. Cross-Domain Fault Diagnosis Using Knowledge Transfer Strategy: A Review. IEEE Access **2019**, 7, 129260–129290, doi:10.1109/ACCESS.2019.2939876.
7. Jafari, S.; Shahbazi, Z.; Byun, Y.-C.; Lee, S.-J. Lithium-Ion Battery Estimation in Online Framework Using Extreme Gradient Boosting Machine Learning Approach. Mathematics **2022**, 10, 888, doi:10.3390/math10060888.
8. Xia, Z.; Ye, F.; Dai, M.; Zhang, Z. Real-Time Fault Detection and Process Control Based on Multi-Channel Sensor Data Fusion. Int. J. Adv. Manuf. Technol. **2021**, 115, 795–806, doi:10.1007/s00170-020-06168-y.
9. Du, C.; Li, W.; Rong, Y.; Li, F.; Yu, F.; Zeng, X. Research on the Application of Artificial Intelligence Method in Automobile Engine Fault Diagnosis. Eng. Res. Express **2021**, 3, 026002, doi:10.1088/2631-8695/ac01ad.
10. Liu, H.; Song, X.; Zhang, F. Fault Diagnosis of New Energy Vehicles Based on Improved Machine Learning. Soft Comput. **2021**, 25, 12091–12106, doi:10.1007/s00500-021-05860-9.
11. Gong, C.-S.A.; Su, C.-H.S.; Chen, Y.-H.; Guu, D.-Y. How to Implement Automotive Fault Diagnosis Using Artificial Intelligence Scheme. Micromachines **2022**, 13, 1380, doi:10.3390/mi13091380.
12. Dalal, A.-A.; Cai, Z.; Al-qaness, M.A.; Alawamy, E.A.; Alalimi, A. ETR: Enhancing Transformation Reduction for Reducing Dimensionality and Classification Complexity in Hyperspectral Images. Expert Syst. Appl. **2023**, 213, 118971.
13. McCann, B.; Keskar, N.S.; Xiong, C.; Socher, R. The Natural Language Decathlon: Multitask Learning as Question Answering. ArXiv Prepr. ArXiv180608730 **2018**.
14. Wang, J.; Wang, C.; Luo, F.; Tan, C.; Qiu, M.; Yang, F.; Shi, Q.; Huang, S.; Gao, M. Towards Unified Prompt Tuning for Few-Shot Text Classification. ArXiv Prepr. ArXiv220505313 **2022**.
15. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. **2018**.
16. Nori, H.; King, N.; McKinney, S.M.; Carignan, D.; Horvitz, E. Capabilities of Gpt-4 on Medical Challenge Problems. ArXiv Prepr. ArXiv230313375 **2023**.
17. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S. Gpt-4 Technical Report. ArXiv Prepr. ArXiv230308774 **2023**.

18. Bi, Z.; Chen, J.; Jiang, Y.; Xiong, F.; Guo, W.; Chen, H.; Zhang, N. Codekgc: Code Language Model for Generative Knowledge Graph Construction. ACM Trans. Asian Low-Resour. Lang. Inf. Process. **2024**, 23, 1–16.
19. Liang, W.; Yuksekgonul, M.; Mao, Y.; Wu, E.; Zou, J. GPT Detectors Are Biased against Non-Native English Writers. Patterns **2023**, 4.
20. Sun, J.; Xu, C.; Tang, L.; Wang, S.; Lin, C.; Gong, Y.; Ni, L.M.; Shum, H.-Y.; Guo, J. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. ArXiv Prepr. ArXiv230707697 **2023**.
21. Gao, A. Prompt Engineering for Large Language Models. Available SSRN 4504303 **2023**.
22. 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. **2023**, 55, 1–35, doi:10.1145/3560815.
23. Wang, J.; Yin, W.; Gao, J. Cases Integration System for Fault Diagnosis of CNC Machine Tools Based on Knowledge Graph. Acad. J. Sci. Technol. **2023**, 5, 273–281, doi:10.54097/ajst.v5i1.5664.
24. Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D. Emergent Abilities of Large Language Models. ArXiv Prepr. ArXiv220607682 **2022**.
25. Hamad, K.; Kaya, M. A Detailed Analysis of Optical Character Recognition Technology. Int. J. Appl. Math. Electron. Comput. **2016**, 244–249.
26. Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. A Survey on Evaluation of Large Language Models. ACM Trans. Intell. Syst. Technol. **2024**, 15, 1–45, doi:10.1145/3641289.
27. Hao, X.; Ji, Z.; Li, X.; Yin, L.; Liu, L.; Sun, M.; Liu, Q.; Yang, R. Construction and Application of a Knowledge Graph. Remote Sens. **2021**, 13, 2511, doi:10.3390/rs13132511.
28. 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. **2022**, 54, 1–37, doi:10.1145/3447772.
29. Jiang, X.-J.; Zhou, W.; Hou, J. Construction of Fault Diagnosis System for Control Rod Drive Mechanism Based on Knowledge Graph and Bayesian Inference. Nucl. Sci. Tech. **2023**, 34, 21, doi:10.1007/s41365-023-01173-8.
30. Deng, J.; Wang, T.; Wang, Z.; Zhou, J.; Cheng, L. Research on Event Logic Knowledge Graph Construction Method of Robot Transmission System Fault Diagnosis. IEEE Access **2022**, 10, 17656–17673, doi:10.1109/ACCESS.2022.3150409.
31. Wang, H.; Qin, K.; Zakari, R.Y.; Lu, G.; Yin, J. Deep Neural Network-Based Relation Extraction: An Overview. Neural Comput. Appl. **2022**, 1–21.
32. Ding, N.; Qin, Y.; Yang, G.; Wei, F.; Yang, Z.; Su, Y.; Hu, S.; Chen, Y.; Chan, C.-M.; Chen, W.; et al. Parameter-Efficient Fine-Tuning of Large-Scale Pre-Trained Language Models. Nat. Mach. Intell. **2023**, 5, 220–235, doi:10.1038/s42256-023-00626-4.
33. Min, B.; Ross, H.; Sulem, E.; Veyseh, A.P.B.; Nguyen, T.H.; Sainz, O.; Agirre, E.; Heintz, I.; Roth, D. Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey. ACM Comput Surv **2023**, 56, doi:10.1145/3605943.
34. Chai, C.P. Comparison of Text Preprocessing Methods. Nat. Lang. Eng. **2023**, 29, 509–553, doi:10.1017/S1351324922000213.
35. Liu, X.; Ji, K.; Fu, Y.; Tam, W.L.; Du, Z.; Yang, Z.; Tang, J. P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-Tuning Universally across Scales and Tasks. ArXiv Prepr. ArXiv211007602 **2021**.
36. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-Rank Adaptation of Large Language Models. ICLR **2022**, 1, 3.
37. Chen, H.; Shi, N.; Chen, L.; Lee, R. Enhancing Educational Q&A Systems Using a Chaotic Fuzzy Logic-Augmented Large Language Model. Front. Artif. Intell. **2024**, 7, 1404940, doi:10.3389/frai.2024.1404940.