scientific reports

Check for updates

OPEN

LLM-based intelligent Q&A system for railway locomotive maintenance standardization

Ao Chen[1]✉, Ye Tian[1], Jinyi Zhang[2,3], Chen Li[1] & Huiyuan Zhang[1]

The standardization of locomotive maintenance data is a critical step in facilitating *reliability centered maintenance* (RCM) data analysis for locomotive maintenance. The performance of this analysis directly impacts the overall effectiveness of the RCM approach. However, challenges such as small sample sizes, nonstandardized data formats, complex analyses, and high labor costs make it difficult to standardize data via traditional manual methods. To address these challenges, we leverage the outstanding performance and unique learning capabilities demonstrated by *large language models* (LLMs), as extensively documented in academic research and industrial applications, to standardize the data related to locomotive maintenance data. This paper adopts a framework based on the premise of "quality data + universal LLMs + fine-tuning". We utilize custom scripts to generate high-quality locomotive maintenance data, integrate the distinct characteristics of such data, and develop customized LLMs specifically designed to standardize locomotive maintenance data via models such as UIE and ChatGLM. Furthermore, we present an auxiliary tool for locomotive maintenance data standardization, along with an intelligent *question and answer* (Q&A) system, both of which are based on the customized LLM. The proposed Q&A system achieves scores of 86.87% for Bleu-4, 89.60% for Rouge-1, 87.54% for Rouge-2, and 94.26% for Rouge-L on the locomotive maintenance dataset and demonstrates impressive performance, with an auxiliary tool efficiency of only 18 ms per piece. Consequently, the customized LLM can not only enhance the performance of locomotive data standardization but also serve as the basis for developing auxiliary tools and intelligent Q&A systems, simplifying the data standardization process and saving time and costs.

The rapid development of rail transportation has brought profound changes to society, facilitating human mobility and social progress. However, as train speeds and passenger numbers have increased, the failure rates of associated systems and components have also increased considerably over the years[1]. To mitigate the costs associated with downtime and enhance the availability of components and the reliability of locomotive operations, it is increasingly imperative to implement economical and precise maintenance strategies for locomotives[2]. *Reliability centered maintenance* (RCM) is an internationally recognized system engineering methodology employed to ascertain the preventive maintenance requirements of equipment and optimize maintenance systems[3]. Originally developed in the aerospace industry[4], RCM has also been successfully applied in sectors such as the petroleum industry[5], shipbuilding[6], and various other fields[7]. Given the considerable economic advantages of RCM in these industries, its application has begun to gain traction within rail transport for the development of effective maintenance strategies[8].

However, the implementation of RCM technology requires skilled personnel with extensive experience, as well as a substantial amount of standardized data, to effectively summarize failure patterns across similar equipment. Consequently, in the field of rail transport, prior to the application of RCM technology in practical scenarios, a thorough analysis of locomotive maintenance data is needed. Nevertheless, this analysis hinges on the standardization and preprocessing of the maintenance data. Common challenges encountered during this process include nonstandardized data formats, limited fault sample sizes, and difficulties in identifying the causes of faults. These issues often lead to incomplete data and result in a time-consuming standardization process.

Simultaneously, the efficient identification of parts that require maintenance and the creation of related repair plans according to a locomotive's unique fault content represent important advancements in the maintenance industry. Rapidly selecting accurate maintenance components in a short amount of time has become more

[1]Zhuzhou CRRC Times Electric Co., Ltd., Data and Intelligent Technology Center, Zhuzhou 412001, China. [2]School of Information Science and Engineering, Shenyang Ligong University, Shenyang 110000, China. [3]Faculty of Engineering, Gifu University, Gifu 501-1193, Japan. ✉email: Ao_Chen_0628@163.com

difficult due to the variety and complexity of locomotive system components. Moreover, the intricacy of standardized locomotive maintenance manuals and disparities in frontline workers' skill levels can result in inefficiencies and mistakes at the job site. A locomotive maintenance intelligent Q&A system that can deliver succinct, relevant, and understandable responses is desperately needed to handle this circumstance.

Recently, *large language models* (LLMs) have undergone rapid development and achieved substantial advancements in industrial and academic settings. This progress is attributed to their exceptional performance and unique contextual learning capabilities, leading to widespread adoption in the field of natural language processing (NLP)[9]. Notably, ChatGPT, which leverages the GPT series of LLMs[10–13], has developed chatbots that exhibit remarkable conversational abilities with humans. These chatbots possess strong mathematical reasoning skills and the capacity to accurately maintain context across multiturn dialogs. These innovations have garnered considerable attention from the NLP community and have substantially impacted the broader AI landscape. *Large Language Model Meta AI* (LLaMA)[14] has become a popular LLM due to its openness and effectiveness, as well as its superior performance in instruction-following tasks, which has attracted considerable attention from the research community. Many researchers[15–17] are committed to fine-tuning or continuously pretraining different versions of LLaMA models to implement new models or tools.

The standardization preprocessing of locomotive maintenance data involves transforming colloquial maintenance records into standardized formats that encompass component names and numbers along with fault labels. This process ensured that the data were uniformly structured for subsequent analyses and applications. The intelligent maintenance *question and answer* (Q&A) system leverages actual fault scenarios encountered during the locomotive operation or maintenance phases to generate real-time maintenance positioning and treatment measures in textual form. This capability facilitates rapid and precise completion of maintenance tasks. In NLP, the task of *information extraction* (IE) involves extracting specific factual information, such as entities, relationships, and events, from natural language text and structuring these outputs[18,19] highlight the importance of IE tasks in this context. Notably, IE tasks are exceptionally well suited for the standardized preprocessing of locomotive maintenance data because of their capacity to organize and categorize information comprehensively. In contrast, the Q&A task involves automatically answering user-posed questions to fulfill their knowledge requirements. In this context, LLMs play a pivotal role. These models were trained via extensive text datasets, resulting in exceptional generality and generalization capabilities. Consequently, LLMs exhibit robust performance in IE and automated Q&A tasks, as demonstrated in[9].

Currently, most locomotive maintenance is based on a mathematical modeling approach[20], which requires much manual work and relies heavily on expert knowledge, leading to subjectivity in the results. Moreover, the reliance on traditional kilometers or time intervals to perform a personalized overhaul program for older and heavily worn locomotives is lacking[21]. This deficiency may lead to some potential faults not being detected and repaired in time, creating a safety hazard for the normal operation of locomotives. By leveraging these advanced LLMs, an intelligent maintenance Q&A system can provide timely and accurate responses to maintenance-related queries, thereby increasing the effectiveness of locomotive maintenance operations.

Therefore, this study focuses on the analysis of locomotive maintenance data by leveraging LLMs. Initially, a meticulous data preprocessing phase is conducted on the existing locomotive maintenance data to assemble a comprehensive and structured locomotive maintenance dataset. This refined dataset is subsequently utilized to generate LLMs customized specifically for locomotives. A data standardization tool is subsequently developed and ground in a customized locomotive LLM. Finally, a multiround dialog dataset is formulated, which serves as the foundation for designing an intelligent maintenance Q&A system anchored in a customized locomotive LLM. In summary, the contributions of this study are as follows:

1. Customized the first locomotive-specific LLM customized for rail data standardization and developed a locomotive maintenance data standardization tool based on this customized LLM;
2. Released the first multiround conversation dataset dedicated to the field of locomotive maintenance and designed an intelligent maintenance Q&A system based on a locomotive-specific LLM;
3. Developed a standardized dataset of locomotive maintenance data and pioneered the automation of the data annotation process.

The remainder of this survey is organized as follows: Section 2 presents an overview of the relevant literature and relevant work. Section 3 describes the implementation specifications of the standardized LLM construction customized for locomotive maintenance data. Section 4 introduces an auxiliary instrument designed to facilitate the standardization of locomotive maintenance data. Section 5 describes an intelligent Q&A system developed for locomotive maintenance applications. Finally, Section 6 summarizes the survey.

Related work

Information extraction

Information extraction (IE) is a prevalent task within the domain of NLP, with the objective of identifying and extracting particular types of information from unstructured or semistructured data, typically in textual form[22]. Based on advancements in IE research, the process can be categorized into three stages:

- Knowledge engineering approaches. Methods belonging to this category leverage the expertise and experience of professionals to create a structured understanding of natural language through the formulation of rules or patterns. A text-matching methodology is subsequently employed to detect and extract targeted information from textual data. This process is typically iterative, commencing with a limited set of extraction rules rigorously tested on an existing corpus. These rules are progressively refined and expanded through iterative testing until an optimal balance between accuracy and recall is achieved. These methods rely on

manually formulated rules and templates, which makes the extraction rules relatively stable. Moreover, once the rules and templates are formulated, the system can efficiently apply these rules for information extraction with high extraction efficiency. However, this methodology has nonmodular black-box characteristics, indicating that internal workings and decision-making processes are not transparent. Consequently, these approaches are heavily reliant on the specialized knowledge of professionals, resulting in considerable labor and time expenditures. Furthermore, owing to a lack of adaptability, such methods cannot be readily migrated or extended to new domains, entity types, or datasets.

- Machine learning approaches. In these approaches, the task is reformulated as a sequence-labeling problem, where the current prediction is influenced not only by the immediate input features but also by the preceding prediction labels. This interdependence arises from strong correlations between the sequences. Common methodologies employed in this context include the *hidden Markov model* (HMM)[23], *support vector machine* (SVM)[24] and *conditional random fields* (CRFs)[25]. These methods can learn and adapt themselves, continuously updating and optimizing model parameters with new data, thus improving model adaptability and accuracy. Simultaneously, the model is trained to automatically identify and address noise and outliers in the data, thereby improving the accuracy of the enhanced information extraction. However, machine learning approaches rely on a large amount of training data, especially for supervised learning algorithms, which require a large amount of manually labeled data to train the model. This reliance increases the cost and time of data labeling and limits the wide application of information extraction methods. Moreover, different machine learning algorithms have different advantages, disadvantages, and scopes of application, so the selection of appropriate algorithms and parameters is crucial for the effectiveness of information extraction. Unfortunately, algorithm selection and parameter tuning are complex processes that require rich experience and specialized knowledge; thus, this type of method increases the difficulty of application.
- Deep learning approaches. Given the comprehensive and holistic nature of deep learning training approaches, gradient propagation techniques are adopted to build increasingly intricate and sophisticated network architectures. This development, in turn, enables the extraction of more discriminative and effective features from natural language data, thereby enhancing the capacity to mine and analyze factual information with greater precision, particularly in identifying specific types of entities and relationships. Consequently, research methodologies such as attention mechanisms, transfer learning, and distant (or far) supervised learning have gained prominence and become the focal points of mainstream research endeavors in the domain of information extraction (IE) tasks. Common methods include LatticeLSTM[26], BERT[27] and ERNIE[28]. Deep learning approaches can automatically extract features from raw data without the need for manual feature selection and extraction, which greatly reduces labor costs, reduces subjective errors, and improves efficiency. Models trained via deep learning approaches have powerful representation capabilities that can capture complex structures and underlying relationships in the data to extract information more accurately. Moreover, these models can achieve end-to-end learning, learning the desired output directly from the input data without the need for explicit modeling of intermediate processes, improving flexibility and accuracy.

Knowledge question and answer

Knowledge Q&A constitutes a fundamental task within the realm of NLP, and it is primarily tasked with responding to inquiries framed in natural language by amalgamating outcomes derived from information retrieval, information extraction (IE), and NLP techniques[29]. Based on advancements and research progress in the domain of knowledge Q&A, the process can be categorically delineated into the following four stages:

- Traditional rule-based approaches. This type of methodology generally encompasses problem classification, answer retrieval, and answer generation processes. Relying mainly on rule-based manual processing, these methods can therefore handle some structured data and provide consistent results within the constraints of these data. Moreover, based on predefined rules and logic, the decision process of such methods is usually traceable and interpretable, which makes them suitable for domains that usually have a strict structure of rules and logic. However, these approaches rely on rules and logic provided by the developer, which results in high labor costs and time expenses. Moreover, as the complexity and scale of knowledge Q&A increase, it may become very difficult to maintain and update the rules, which may lead to a "rule explosion". Furthermore, such approaches cannot automatically learn new knowledge from data, which limits their adaptability when new data or new tasks are addressed[30].
- Knowledge graph-based approaches. In this methodological framework, the primary objective is first to leverage structured data, text corpora, and semistructured data to construct a comprehensive domain knowledge graph. This graph subsequently serves as the foundation for extracting precise and detailed answers. However, the limitations associated with knowledge graphs must be acknowledged. These limitations include excessive dependence on expert knowledge, the potential for incomplete knowledge representation, a lack of robust linguistic understanding, and other inherent disadvantages. Knowledge graph-based approaches use structured storage and querying, which help to obtain the most accurate information and reduce misunderstandings and ambiguities, thus enhancing the accuracy and reliability of the Q&A system. The knowledge graph supports fast retrieval and reasoning, which can quickly find the information related to the query and generate the best answer, thus improving the response speed of the Q&A system. However, a knowledge graph is constructed using considerable manual labor and time, and its knowledge coverage is somewhat limited, making it difficult to cover all domains and details. Moreover, some knowledge graph-based Q&A systems may rely on specific templates or rules to generate answers. Although this approach improves the accuracy of the answers to a certain extent, it may also result in the system being unable to respond flexibly when facing new types of questions[31].

- Traditional deep learning-based approaches. In this methodological approach, a small deep learning model is utilized to convert natural language into a semantic representation. Q&As are subsequently represented as vectors within this semantic space. The determination of the optimal answer is facilitated by calculating the similarity matching score between the respective vectors. Based on deep learning approaches, these methods automatically extract useful features from raw data, greatly reducing the burden of manual feature engineering. Simultaneously, these methods can perform end-to-end learning from the input data to the output results, which can directly optimize the entire Q&A process and improve the overall performance. Moreover, based on deep learning, the approaches can be adapted to different tasks and scenarios by adding more layers or adjusting the model structure. Consequently, they can be flexibly adapted to the needs and question types to provide more personalized answers. Furthermore, with a large amount of data training, the deep learning approach can learn the intrinsic laws of the data and has a better prediction ability for new data that have not been previously described. These attributes ensure high accuracy and reliability when various problems are addressed. However, such methods require a large amount of data for training; otherwise, they are prone to overfitting. In knowledge Q&A, this requirement means that many Q&A pairs and related knowledge need to be collected and processed to ensure accuracy and generalizability. In practical applications, obtaining high-quality Q&A data and related knowledge is often a considerable challenge. Moreover, the performance of deep learning-based methods is often affected by hyperparameter settings, which indicates that the hyperparameters need to be carefully tuned and optimized for the best performance; unfortunately, the selection and tuning of hyperparameters is a complex and time-consuming process that relies heavily on professional knowledge and experience[32].
- LLM-based approaches. In deep learning methodologies, large neural networks, such as GPT[10–13] and LLaMA[14], are employed as pretrained models and are subsequently fine-tuned for specific domains based on downstream tasks. These LLMs typically encompass hundreds of billions of parameters, enabling them to address knowledge-based Q&A tasks. By leveraging vast textual data, LLMs capture richer features by learning contextually relevant meanings and structures of natural language, thereby enhancing their performance in handling such tasks. However, LLMs have shortcomings in terms of "factuality" and "real-time", which are insufficient in domain knowledge Q&A scenarios that require accurate answers, so external knowledge bases have to be used to generate high-quality and accurate responses.

LLMs

LLMs primarily denote transformer-based language models that incorporate hundreds of billions (or more) of parameters derived from extensive text-based training[33]. These models learn the structure and usage of language by analyzing vast quantities of textual data, which empowers them to execute a diverse array of language-related tasks. LLMs have emerged as the driving force behind language understanding, generation, and application owing to their exceptional proficiency in the realm of NLP. Specifically, within NLP, they exhibit outstanding performance in tasks such as text generation, Q&A systems, and dialog generation. Furthermore, LLMs play a pivotal role in the construction of knowledge graphs, the development of intelligent assistants, and other advanced applications. Their versatility extends to tasks such as code generation, text summarization, and translation, highlighting their broad applicability and transformative potential[9].

LLMs have been able to achieve rapid and successful development due to the following key techniques:

- Scaling techniques: Enhancing the capacity of a model involves leveraging larger models, increasing the dataset size, and increasing the computational resources required for the training process.
- Training techniques: The successful training of a capable LLM presents substantial challenges owing to its immense size. To learn the network parameters of an LLM, distributed training algorithms are indispensable and often necessitate the concurrent use of multiple parallel strategies.
- Ability elicitation techniques: Pretrained on vast and varied large-scale corpora, an LLM provides latent possibilities as a flexible instrument for general-purpose activities. These innate skills, however, might not be readily apparent while the LLM is performing specific activities. Context-specific learning paradigms or customized task instructions are carefully planned and executed to extract and utilize these potential skills.
- Alignment tuning techniques: LLMs are trained on pretrained corpora to identify and integrate various data features, encompassing high-quality and low-quality information. Consequently, these models may generate content that is harmful, biased, or detrimental to individuals' well-being. To mitigate this risk, alignment strategies are employed to ensure that the actions and outputs of LLMs align with societal norms and human ethical principles.
- Tool manipulation techniques: LLMs are essentially trained as text generators on extensive corpora of unstructured plain text. Consequently, they often demonstrate suboptimal performance on tasks that are ineffectively represented in textual formats, such as numerical computations. Furthermore, their functionality is inherently limited by the data on which they were pretrained, hindering their ability to access real-time or up-to-date information. To address these limitations, researchers and practitioners integrate external tools and resources to mitigate the inherent shortcomings of LLMs.

Different industries and business scenarios have different needs for models, and customized LLMs are necessary[34]. For example, customized LLMs can be optimized for specific business logic, data characteristics and objectives to ensure the accuracy and efficiency of the model in practical applications[34]. In user interaction scenarios, the customized LLM can subsequently make predictions and recommendations based on the user's personalized needs and preferences, thus enhancing the user experience and satisfaction[35]. Furthermore, for industries involving sensitive data, customized LLMs can be deployed locally to avoid the risk of data leakage and to meet the industry's stringent requirements for data privacy and security[36].

Construction of customized LLMs for the standardization of locomotive data

This section explores the field of locomotive maintenance data, focusing on the standardization of such data and their application within locomotive maintenance Q&A) systems in the rail transit industry. The goal is to develop LLMs specifically designed for two downstream tasks: IE and knowledge-based Q&A. Following the framework of "high-quality data + general LLMs + fine-tuning," this paper begins by utilizing relevant locomotive maintenance data. The data then undergo a rigorous preprocessing phase to create a small, high-quality annotated corpus. Building on this curated corpus, the paper applies fine-tuning techniques to train a general LLM (the fine-tuning technique mainly utilizes P-tuning v2, in which the prompt needs to be designed; moreover, owing to the hardware limitations of the local server, the batch and epoch are modified, and the remaining parameters are referred to the model's official defaults), ultimately resulting in a locomotive specific, customized LLM. A locomotive maintenance intelligent Q&A system is subsequently constructed based on this customized LLM. The overall research process is shown in Fig. 1.

In this paper, the selection of general LLMs is guided by several key considerations: their performance on Chinese text corpora, the number of stars received by related applications on the GitHub platform, and their open-source or commercial status. After a comprehensive evaluation for the IE task, *universal information extraction* (UIE) was selected because of its strong sample-less capability, support for Chinese and English cross-extraction, and ability to deploy the model open-source locally to eliminate the risk of data leakage.[37]. For the knowledge-based Q&A task, ChatGLM was selected for its ability to generate logical and consistent answers based on context for intelligent Q&A scenarios, as well as its support for Chinese Q&A and, most importantly, its ability to deploy the model open source locally to eliminate the risk of data leakage[38].

A LLM for standardized information extraction of locomotive data

In this section, we utilize a comprehensive dataset of locomotive maintenance data to fine-tune a general LLM with the goal of enhancing its performance in extracting relevant information from the locomotive maintenance data. The detailed process and procedural steps involved in the information extraction task, performed by this fine-tuned LLM on the locomotive data, are illustrated in Fig. 2. First, locomotive maintenance data are subjected to comprehensive data preprocessing to build corpora specifically for this purpose. The UIE model uses P-tuning of the corpora to develop a customized LLM optimized for locomotive maintenance-related object information extraction. Finally, this customized LLM is used to obtain accurate information extraction results from the maintenance data, which include the system name, component name, accessory name, fault label, and number name.

Data preprocessing

In this section, the data preprocessing phase primarily includes data augmentation, data clustering, and data transformation. The foundational data are sourced from revised locomotive preshattering repair records collected in the field from 2017-23. These data are then integrated with the maintenance specification documentation of the smallest maintenance unit. A manual process of data cleansing, screening, and labeling is subsequently conducted to ultimately curate an original dataset of locomotive maintenance data that includes labeling information related to components, accessories, and fault types. This original dataset comprises 4,970 standardized data entries, predominantly covering eleven subsystems, such as the traction motor. The content primarily consists of system names, component names, accessory names, fault labels, and number names.

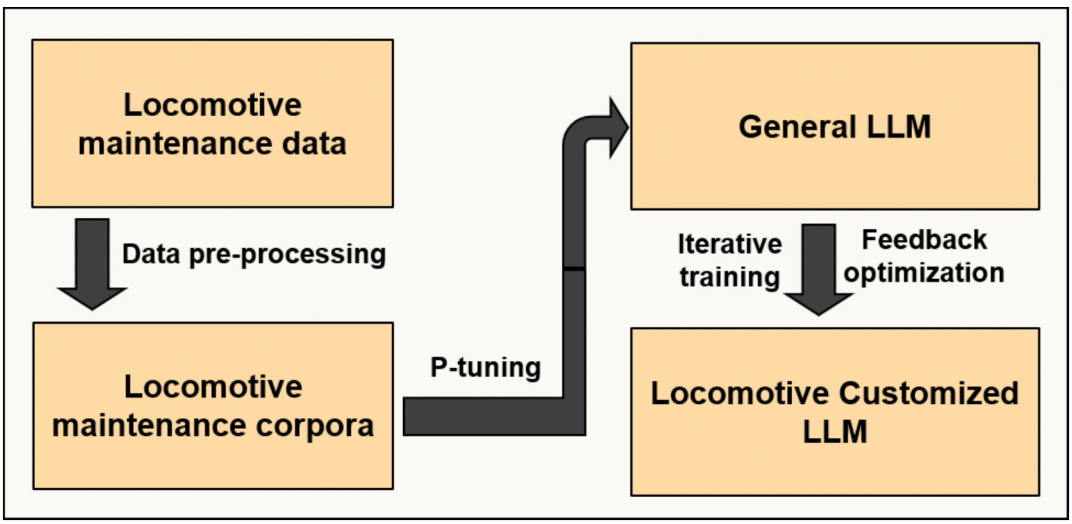


Fig. 1. Process of constructing an LLM for locomotive data standardization. Process of constructing an LLM for locomotive data normalization. First, through the data preprocessing designed in this paper, the locomotive maintenance data are generated in the locomotive maintenance corpora. Then, based on the corpora, the *prompts are fine-tuned* (P-tuned) to the generalized LLM. Subsequently, through iterative training and feedback optimization, a customized LLM suitable for the locomotive domain is finally generated.

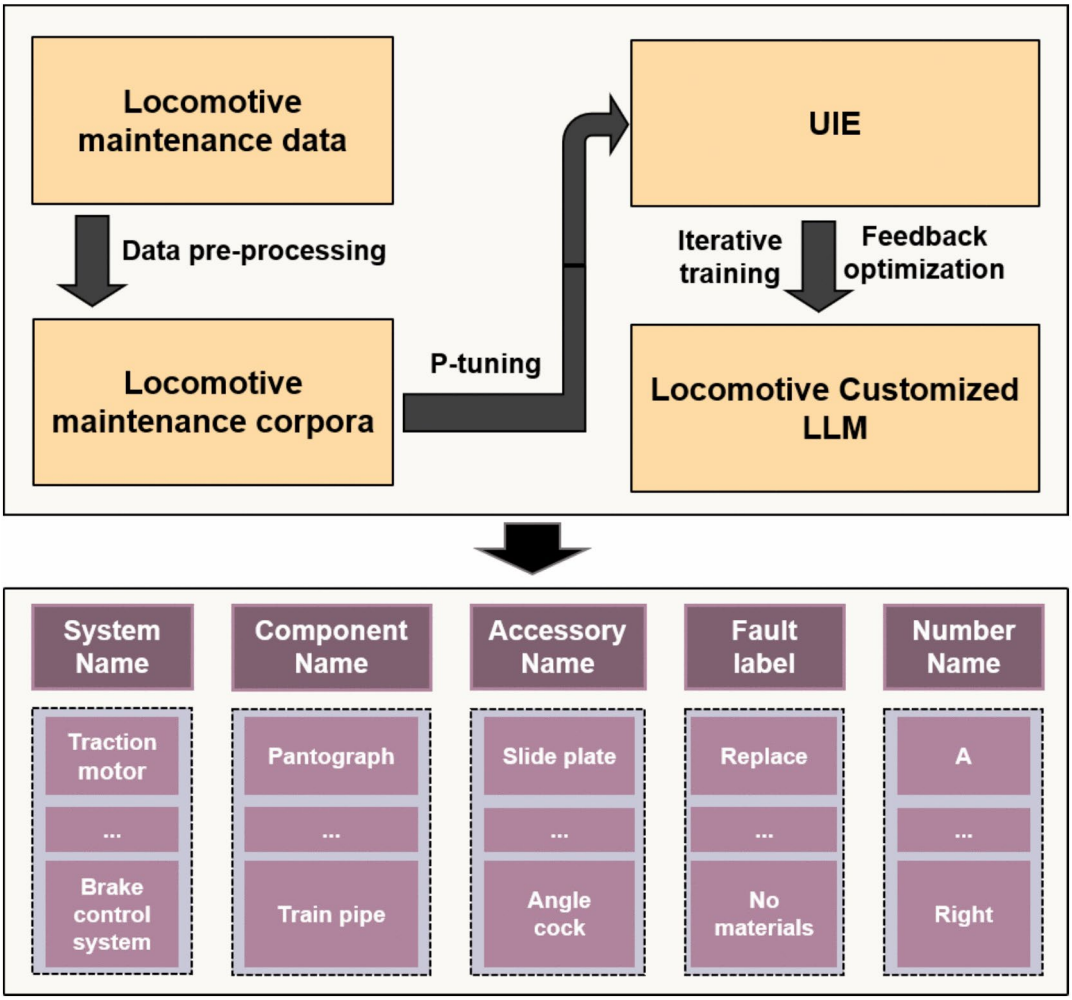


Fig. 2. Process of constructing a customized LLM for locomotive data standardization. First, locomotive maintenance data are subjected to comprehensive data preprocessing to build corpora specifically for this purpose. The UIE model performs P-tuning via corpora to develop a customized LLM optimized for locomotive maintenance-related object information extraction. Finally, this customized LLM is used to obtain accurate information extraction results from the maintenance data, which include the system name, component name, accessory name, fault label, and number name.

The original dataset has a limited volume of data, characterized by its small scale, with a predominance of fault label types classified as replacements. Consequently, this study aims to expand the dataset's size by incorporating fault documentation from video security systems collected in the field between 2020 and 2023. The methodology for data expansion is outlined as follows: First, individual maintenance records for each month within the specified timeframe (2020–2023) are compiled. Next, the maintenance specification document is referenced to identify the minimum maintenance unit, which facilitates the manual annotation of critical information, including component names, accessory names, fault labels, and number names. Finally, these annotated data are merged with the original dataset to create an expanded dataset.

The application of data expansion techniques increases the amount of data in the expanded dataset by 56% compared with the original dataset, which contains 2,791 video security fault records, 760 maintenance records with fault labels, and 51 records without fault labels. This process enhances the quantity of locomotive maintenance data and increases the overall size of the dataset.

The high proportion of "replace" among the key fault labels in the supplemental data may affect the performance of the LLM in information extraction. Therefore, this paper employs data clustering to balance the distributions of various types of fault labels. This approach aims to investigate the impact of the proportion of fault label types on the performance of information extraction.

The process of data clustering is as follows: each record within the expanded dataset related to locomotive maintenance is treated as a text, encapsulating the fault content and the corresponding processing method. A cosine similarity threshold of 0.8 is subsequently established. Figure 3 shows a histogram of categories derived from various cosine similarity thresholds, revealing that the maximum number of categories is achieved at a threshold of 0.8, thereby justifying this choice. The pseudocode for computing cosine similarity, which employs the Jieba segmentation tool (https://github.com/fxsjy/jieba) and follows Equation 1, is depicted in Fig. 4. Text

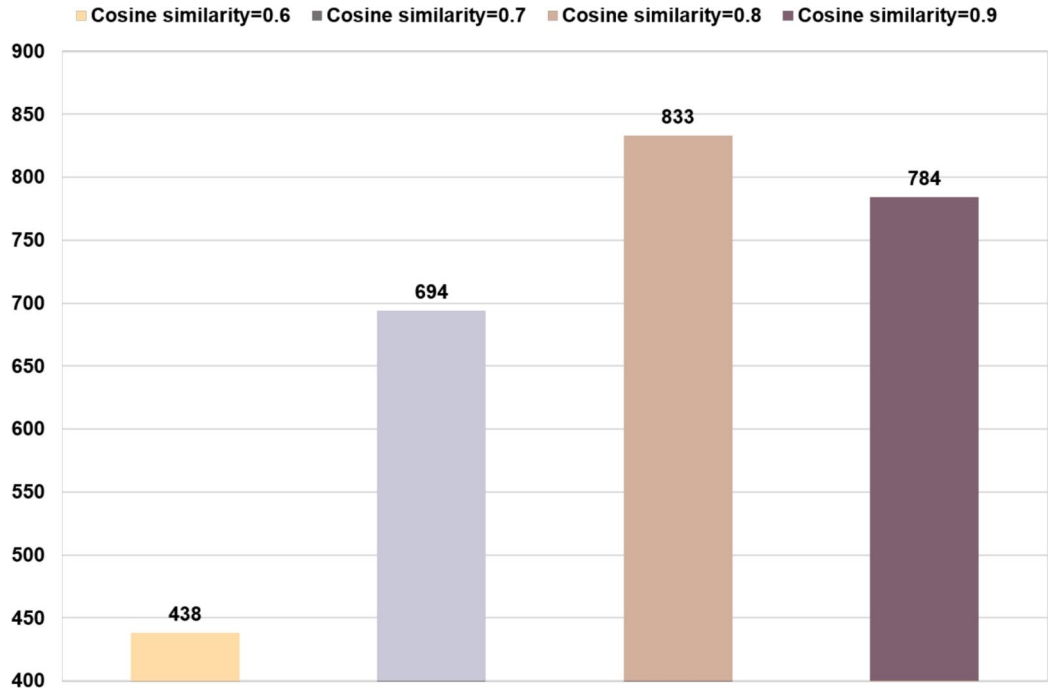


Fig. 3. Histogram of clustering categories with different thresholds.

Input :	two text sentences A and B
Output:	cosine similarity of the angle between word frequency vectors of A,B

1. Using the Segmentation Tool to segment the text sentence A,B into words: A=[A_cut_1, A_cut_2,…, A_cut_i], B=[B_cut_1, B_cut_2,…, B_cut_i]
2. Combine all the non-repeating phrases in A,B into a bag of words: C
3. Calculate the word frequency of A_cut_1 in A, B_cut_1 in B in bag of words C: A=[A_1, A_2,…, A_i], B=[B_1, B_2,…, B_i]
4. Calculate the cosine similarity between A,B word frequency vectors

Fig. 4. Cosine similarity pseudocode.

data with a cosine similarity exceeding this threshold are grouped into the same class, whereas text data falling below the threshold are classified as unclustered, resulting in the identification of distinct file categories. Next, data from the clustered files are extracted based on a 10% upward rounding ratio. These extracted data are ultimately merged with the unclustered data to construct a comprehensive clustered dataset.

$$S_{\text{cosine\_similarity}}(\mathrm{A},\mathrm{B}) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n}(A_i \times B_i)}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}. \tag{1}$$

In the context of data clustering, 3,933 locomotive maintenance data entries are extracted from the expanded dataset. This dataset comprises 1,429 clustered entries and 2,504 unclustered locomotive maintenance data entries. This process ensures a balanced representation of various fault labeling categories within the clustered dataset.

Before the LLM is fine-tuned with the prepared locomotive maintenance corpora, these corpora must be converted from their current Excel format to the JSON format required by the UIE. Additionally, the positions of entities within the text must be annotated. Typically, this conversion and annotation process involves manual data annotation, which requires a high level of business knowledge and expertise from the annotators. However, manual annotation is not only time-consuming but also inefficient, presenting considerable challenges to the overall annotation process.

Consequently, the keyword position matching annotation approach is used in this study to swiftly and automatically transform the locomotive maintenance corpora into a dataset in JSON format that the UIE requires. “content” represents the combined text of the fault content and the treatment, “result_list” represents

the labeled entities and their start and end positions in content, and "prompt" represents information such as the component name, part name, accessory name, fault label, and number name. Figure 5 illustrates an example of the conversion.

To validate structural and qualitative consistency between the original dataset and the extended dataset, we conduct comparative analysis through two complementary approaches: lexical distribution examination and discriminative feature evaluation. First, Chinese word segmentation using Jieba with domain-specific stopword filtering enabled systematic word frequency analysis. The visual representation of term distributions through comparative word clouds (Fig. 6) demonstrates significant pattern alignment. Second, we employ TF-IDF (https://github.com/scikit-learn/scikit-learn) for discriminative feature extraction, with Table 1 presenting the top 10 feature weights from both datasets. The strong correlation in both lexical distributions and feature rankings statistically confirms the structural homogeneity and quality preservation achieved by our dataset extension methodology.

Experiment

Datasets The datasets used for the experiments in this section have three primary components: the original dataset, the expanded dataset, and the clustered dataset. Each of these datasets includes a compilation of text that integrates fault descriptions with their corresponding treatment approaches. Additionally, they provide detailed information such as the component name, the component level 1 number, the component level 2 number, the accessory name, the level 1 accessory name, and the precise start and end positions of the fault labels within the combined text. In this section, the datasets are divided into training, validation, and test sets at a 1:1:8 ratio. Table 2 presents the fundamental characteristics of each dataset, with text segmentation performed via the Chinese segmentation tool Jieba.

Hyperparameters The settings of the hyperparameters of the experiments in this section are shown in Table 3, and these settings include the number of batches, the number of epochs, the number of GPUs, the optimizer, the learning rate, and the longest input length.

Metrics The evaluation metrics for the experiments in this section use *precision* (Pr), *recall* (Re), F1, and runtime, and these evaluation metrics are specified as shown in Equation 2, where the meaning of each symbol is as follows: TP - the number of positive samples correctly identified; FP - the number of negative samples that are misreported; TN - the number of negative samples correctly recognized; and FN - the number of positive samples that are missed.

Input	Output
B节副司机侧辅灯不亮更换辅灯灯泡后试验正常 Section B assistant driver's side auxiliary lamp does not light up, after replace the auxiliary lamp bulb, the test is normal	{"content": "B节副司机侧辅灯不亮更换辅灯灯泡后试验正常", "result_list": [{"text": "辅灯", "start": 6, "end": 8}], "prompt": "部件名称"} {"content": "Section B assistant driver's side auxiliary lamp does not light up, after replace the auxiliary lamp bulb, the test is normal", "result_list": [{"text": "auxiliary lamp", "start": 6, "end": 8}], "prompt": "Component Name"}
	{"content": "B节副司机侧辅灯不亮更换辅灯灯泡后试验正常", "result_list": [{"text": "辅灯", "start": 6, "end": 8}], "prompt": "一级配件名称"} {"content": "Section B assistant driver's side auxiliary lamp does not light up, after replace the auxiliary lamp bulb, the test is normal", "result_list": [{"text": "auxiliary lamp", "start": 6, "end": 8}], "prompt": "Level 1 accessory name"}
	{"content": "B节副司机侧辅灯不亮更换辅灯灯泡后试验正常", "result_list": [{"text": "B", "start": 0, "end": 1}], "prompt": "部件一级编号"} {"content": "Section B assistant driver's side auxiliary lamp does not light up, after replace the auxiliary lamp bulb, the test is normal", "result_list": [{"text": "B", "start": 0, "end": 1}], "prompt": "Component level 1 number"}

Fig. 5. Example of a data conversion result. "content" represents the combined text of the fault content and the treatment, "result_list" represents the labeled entities and their start and end positions in content, and "prompt" represents information such as the component name, part name, accessory name, fault label, and number name.

(a) Original dataset Word Cloud (b) Expanded dataset Word Cloud

Fig. 6. Comparison of word clouds between the original dataset and the extended dataset.

Dataset type/Feature word	Replace	Normal	Test	Inspection	Driver	Malfunction	Storage	Operation	Seat	Good
Original dataset	54.11	50.68	37.74	33.82	15.85	15.23	14.19	10.83	8.47	7.70
Expanded dataset	51.29	62.03	29.78	26.88	14.16	13.26	11.21	9.83	6.68	6.10

Table 1. Comparison of TF-IDF based on original dataset and expanded dataset (In [%]).

Dataset type	Data entries	Fault label type
Original dataset	4970	21
Expanded dataset	7761	51
Clustered dataset	3933	51

Table 2. Basic information about the datasets.

Parameter	Value/Type	Parameter	Value
Batch	8	learning rate	0.00001
Epoch	100	optimizer	AdamW
GPU	8(A30)	maximum input length	512

Table 3. Basic information about the datasets.

Dataset type	Precision/%	Recall/%	F1/%	Runtime/s
Original dataset	93.65	93.28	93.46	6338.74
Expanded dataset	93.64	91.78	92.70	10953.33
Clustered dataset	90.66	88.81	89.73	4823.33

Table 4. Basic information about the datasets.

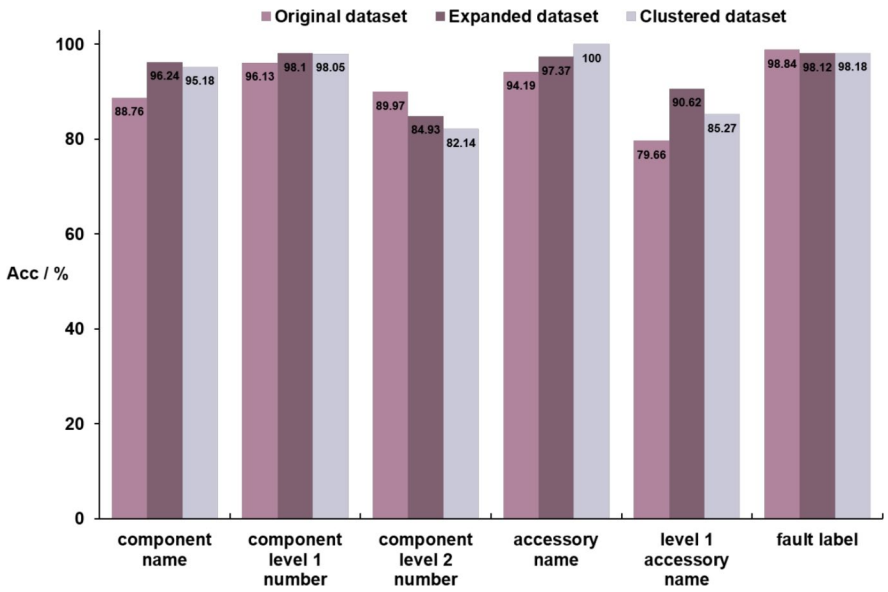


Fig. 7. Comparison of the accuracy of standard records based on different locomotive maintenance data testing sets.

$$
\begin{aligned}
Precision &= \frac{T_{\mathrm{P}}}{T_{\mathrm{P}} + N_{\mathrm{P}}} \times 100\% \\
Recall &= \frac{T_{\mathrm{P}}}{T_{\mathrm{P}} + F_{\mathrm{N}}} \times 100\% \\
F1 &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\%
\end{aligned}
\tag{2}
$$

Results In this section, we conduct a comprehensive analysis using various locomotive maintenance datasets, as detailed in Table 2. By employing the hyperparameters specified in Table 3, the UIE model is fine-tuned on a local server. The aim of this process is to develop a customized LLM specifically designed for standardized information extraction related to locomotive data. Table 4 presents the training performance metrics of the UIE model across different locomotive maintenance datasets. Furthermore, Figs. 7, 8 and 9 illustrate the distinct testing performance of the customized LLM, which is based on the UIE framework, in extracting information from various locomotive maintenance datasets.

Discussion

The training outcomes of the unified information extraction (UIE) model, which utilizes various locomotive maintenance datasets, demonstrate that, with the exception of the execution time, the information extraction task based on the original dataset has superior performance. Specifically, the precision, recall, and F1 score for this task reach 93.65%, 93.28%, and 93.46%, respectively. Notably, the training set used in this instance

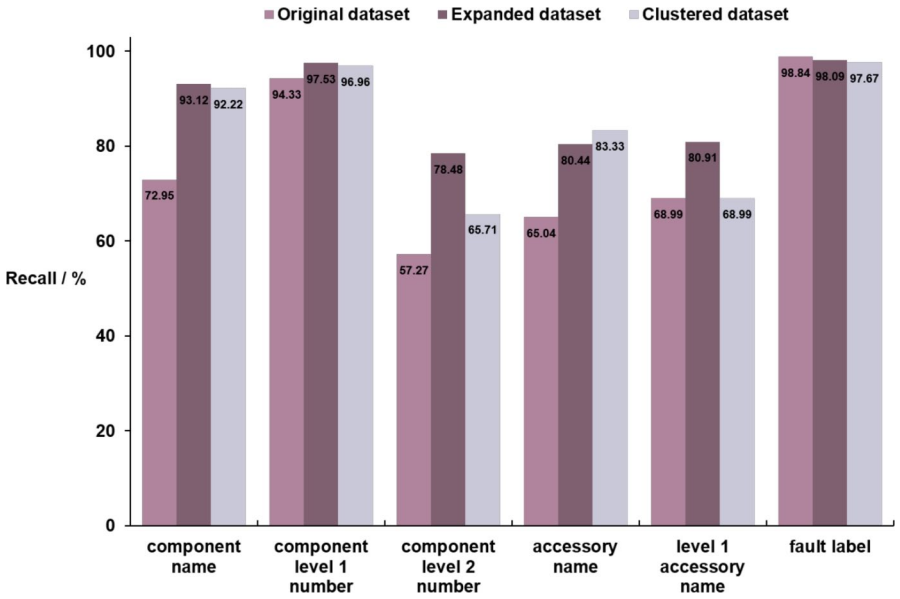


Fig. 8. Comparison of the recall of standard records based on different locomotive maintenance data testing sets.

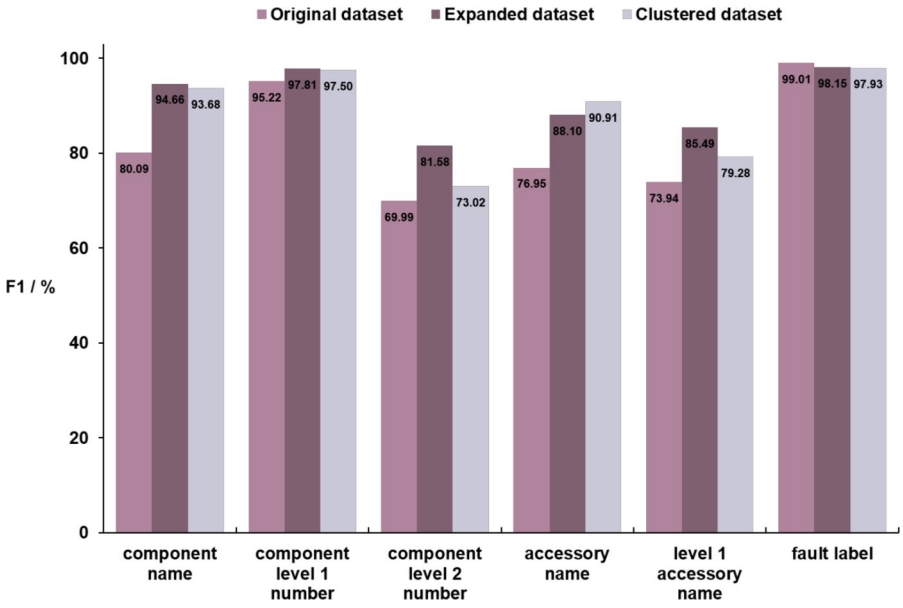


Fig. 9. Comparison of F1 values of standard records based on different locomotive maintenance data testing sets.

comprises only 10% of the entire original dataset, indicating that the UIE model can effectively perform the locomotive maintenance data information extraction task even with a limited amount of data.

Furthermore, the test results from various datasets in the information extraction task of a customized large language model, which is based on the UIE model, were analyzed. This analysis considered the names and quantities of parts and accessories, as well as the importance of fault labels. The findings indicate that information extraction via an expanded dataset produces the most favorable results, whereas information extraction via a clustered dataset is less effective.

From these observations, we can conclude that the use of a large standardized information extraction model based on the UIE framework for extracting information from locomotive maintenance data can benefit from an increase in the number of data entries. Expanding the dataset enhances the training performance of the information extraction task. Conversely, balancing the labels of fault types does not considerably improve extraction performance and may even lead to a decline in effectiveness. These insights have important

implications for the development and optimization of information extraction models customized for locomotive maintenance data.

A LLM for standardized knowledge Q&A of locomotive data

The application of an information extraction LLM for standardizing locomotive data can serve as an effective tool for preprocessing maintenance data. However, this method imposes stringent requirements regarding the dataset format. Specifically, explicit marking of the initial and termination positions of the information within the text is necessary, complicating the marking process. In practical scenarios, it is conceivable that certain information may not be explicitly present in the text, making it impossible to mark. This situation presents a considerable challenge for marking locomotive maintenance data, as it introduces difficulties in accurately identifying and extracting the required information.

Fortunately, ChatGLM has minimal requirements for dataset formatting while demonstrating commendable performance in knowledge Q&A tasks via publicly available datasets such as Gigaword[39] and Xsum[40]. Therefore, in this study, we propose improving the annotation methodology for locomotive maintenance data by converting the existing information extraction format dataset into a structure that resembles knowledge Q&A datasets. This transformed dataset is then used to customize ChatGLM into a standardized knowledge Q&A LLM specifically designed for locomotive data.

Automated approach to data annotation

Traditional data annotation relies primarily on manual methods, which require annotators to possess extensive professional expertise and operational proficiency. Additionally, these manual processes demand substantial time investments, leading to a time-consuming and highly subjective approach to annotation. To address these limitations, this section introduces a scripting methodology customized to the unique characteristics of locomotive maintenance data. By developing a specialized script, we facilitate batch automated labeling of locomotive maintenance data, utilizing the expanded dataset in Table 2. In practical applications, the accessory name and component level 2 number are rarely utilized. Therefore, this section annotates only the component name, primary number of the component, primary accessory name, and fault label. The method first uses the fault contents and treatment methods in the original data to merge and automatically generate a "sentence" and then uses the index of the original data to find the specific contents corresponding to the "component name, level 1 accessory name, component level 1 name, and fault label" to generate "answers". An illustrative example of data annotation derived from the customized script is presented in Fig. 10. In this figure, the term "sentence" refers to the combined text that includes the fault content and its corresponding solution, whereas "answers" denote the specific details, including the component name, level 1 accessory name, component level 1 name, and fault label.

Experiment

Datasets In this section, the dataset used for the experiments is derived from the locomotive maintenance dataset, which has been processed by a specialized script. This processed dataset contains 7,761 locomotive maintenance records. The dataset has been divided into training, validation, and test sets, following a ratio of 6:2:2. Additionally, Jieba was employed as the Chinese segmentation tool throughout the experimental process.

Hyperparameters The settings of the hyperparameters of the experiments in this section are shown in Table 5, and these settings include the number of batches, number of GPUs, learning rate, number of threads, maximum input length and maximum output length.

Metrics In this paper, Bleu-4, Rouge-1, Rouge-2 and Rouge-L are used as evaluation metrics. Among them, Bleu-4 is based on the accuracy of judging the similarity of two sentences and the fluency of the sentences, and the mathematical expression of the evaluation index is shown in Equation 3, in which the individual symbols in the formula are explained as follows: C_i - machine translation, S_{ij} - reference translation, S_i=s_{i1},s_{i2}...,s_{im} - all neighboring 4 Chinese characters in C_i are extracted to form a set; w_k - the kth phrase in the set, h_k (C_i) - the number of times the kth phrase w_k appears in the machine translation C_i, h_k (S_{ij}) - the number of times the kth phrase w_k occurs in the reference translation s_{ij}, $B_P(C_i, S_{ij})$ - penalty term, P_n - the precision rate of the n-tuple words; l_{ci} - the length of the machine translation; and $l_{S_{ij}}$ - the length of the reference translation. As an index for automatically assessing the similarity between the generated text and the reference text, the Bleu evaluation metric provides a quantitative evaluation standard, making the assessment fairer and more objective. Moreover, it can quickly evaluate many translation results, which greatly improves the evaluation efficiency.

$$
\begin{aligned}
P_n\left(C_i, S\right) &= \frac{\sum_k \min\left(h_k\left(C_i\right), \max_{j \in m} h_k\left(S_{ij}\right)\right)}{\sum_k h_k\left(C_i\right)} \\
B_P(C_i, S_{ij}) &= \begin{cases} 1, l_{ci} > l_{S_{ij}} \\ e^{1-\frac{l_{S_{ij}}}{l_{ci}}}, l_{ci} < l_{S_{ij}} \end{cases} \\
Bleu-4 &= B_p \times e^{\sum_{n=4}^{N} 0.25 lg P_n} \times 100\%
\end{aligned}
\tag{3}
$$

Rouge evaluates the abstract based on the co-occurrence information of n-grams in the abstract, which is an evaluation method oriented to the recall rate of n-grams, and Rouge-1, Rouge-2 and Rouge-L denote the Rouge-N based on 1-gram, based on 2-grams and based on the longest common meta-grams, respectively. The mathematical expression is shown in Equation 4, where the individual symbols in the formula are interpreted as follows: n_{gram} - the number of n phrases common to the reference translation and the machine translation; m_{gram} - the number of n phrases extracted by the reference translation; X - the reference translation; Y - the

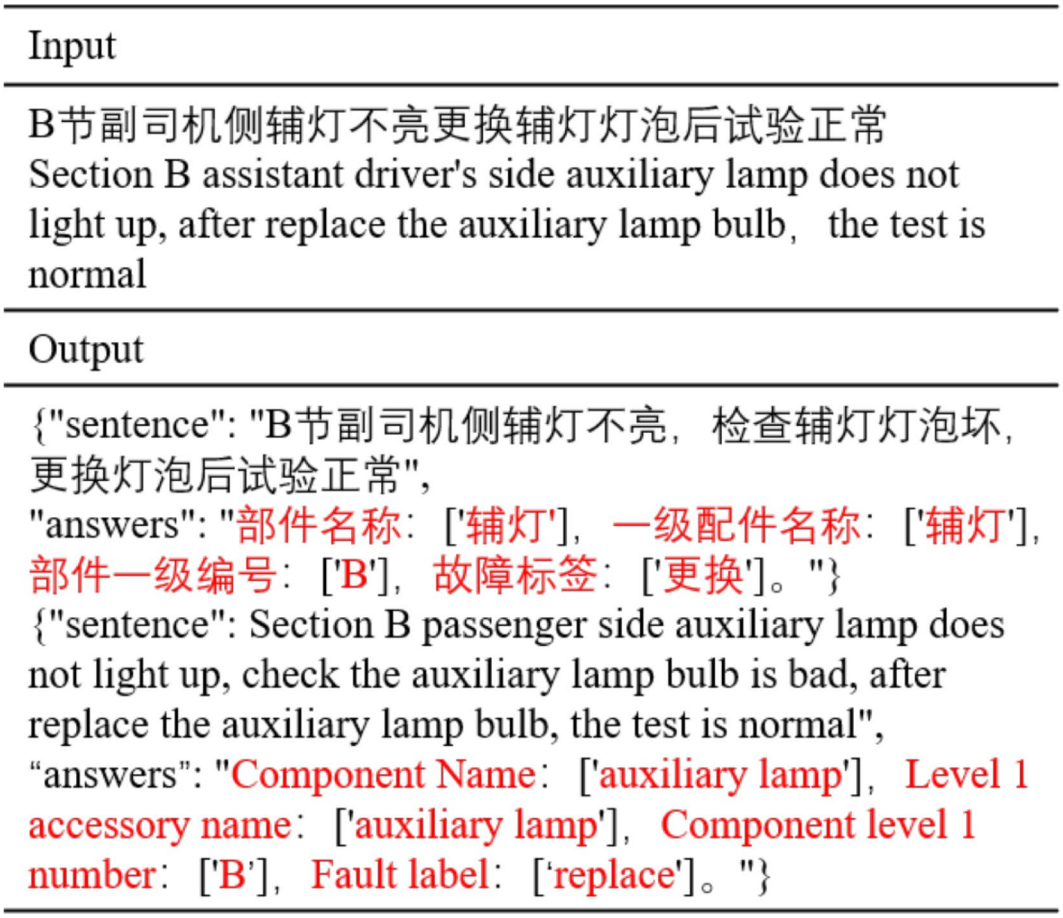


Fig. 10. Example of the data annotation obtained via the customized script. In this example, the term "sentence" refers to the combined text that includes the fault content and its corresponding solution, whereas "answers" denotes the specific details, including the component name, level 1 accessory name, component level 1 number, and fault label.

Parameter	Value	Parameter	Value
Batch	64	Learning rate	0.02
Thread	10	Maximum output length	128
GPU	8(A30)	Maximum input length	512

Table 5. Setting hyperparameters.

machine translation; $l(X, Y)$ - the length of the longest common subsequence of X, Y; m - the extraction length of the reference translation; n - the extraction length of the machine translation; and β - manual setting parameter. In this paper, we take $\beta = 1.2$. Rouge measures the degree of coverage of key information of the reference text by the generated text through the recall rate, which considers the completeness and informativeness of the generated text, thus enabling a more comprehensive assessment of the quality of the generated text.

$$\begin{aligned} Rouge - N &= \frac{n_{gram}}{m_{gram}}, N = 1, 2 \\ Rouge - L &= \frac{(1+\beta^2) \times \frac{l^2(X,Y)}{mn}}{\frac{l(X,Y)}{m} + \beta^2 \times \frac{l(X,Y)}{n}} \end{aligned} \tag{4}$$

Results In this section, we first utilize automated data annotation within the locomotive dataset to create the training and test sets. Next, we employ ChatGLM to fine-tune the local server, using the hyperparameters specified in Table 5 to obtain the customized LLM for standardized knowledge Q&A of locomotive data. This fine-tuning process produces a customized LLM specifically designed for standardized knowledge-based Q&A related to locomotive maintenance data. Then, we conduct inference testing via the customized LLM. Illustrative

examples of the inference performance and corresponding results are presented in Figs. 11 and 12, respectively. In Figure 11, "sentence" refers to the combined text that includes the fault content and the corresponding treatment method. "answers" denotes the original annotated text, which comprises the component name, the level 1 accessory name, the component level 1 number, and the fault label. "prediction" indicates the output produced by the model's predictive capabilities, which also includes the component name, the level 1 accessory name, the component level 1 number, and the fault label. Additionally, to illustrate the effectiveness of the ChatGLM-based customized LLM, this section presents a comparative analysis of ChatGLM's performance against other models using a public dataset (see Tables 6 and 7).

Discussion

Figure 11 shows that the results are accurate in most cases, but there is an error in the "Level 1 part name". Therefore, the system is accurate in answering "component name, component level 1 number and fault label" but poor in answering "level 1 accessory name". We analyzed the possibility that "component name, component level 1 number and fault label" may appear in the "sentence", but "level 1 part name" needs to be linked to specific maintenance documents, and there are difficulties in extracting information. Therefore, based on the analysis of this error, in future research, we will explore techniques such as human-in-the-loop and adversarial training to improve the accuracy. An analysis of the results presented in Table 6 clearly reveals that the use of ChatGLM for knowledge-based Q&A on the Gigaword dataset yields optimal performance, with Rouge-1, Rouge-2, and Rouge-L scores reaching 38.9%, 20.0%, and 36.3%, respectively. These scores significantly surpass the corresponding performances achieved by MASS and UniLMv2. Furthermore, Table 7 demonstrates that when ChatGLM is applied to the Xsum dataset for knowledge Q&A, the highest performance is also attained, with Rouge-1, Rouge-2, and Rouge-L scores of 45.5%, 23.5%, and 37.3%, respectively. These scores are notably higher than those obtained by the BART and T5. Additionally, Fig. 12 illustrates that a ChatGLM-based knowledge Q&A system customized for an LLM achieves exceptional Rouge-1, Rouge-2, and Rouge-L scores of 89.6%, 87.54%, and 94.26%, respectively. These scores are markedly superior to the performance of ChatGLM on other datasets. Based on these findings, the ChatGLM-based knowledge Q&A system customized for LLM is well suited for the standardization of locomotive maintenance data.

Supplementary experiments

In previous experiments to ensure fairness, the hyperparameters and tuning strategies are set the same as chatGLM on the public dataset. In order to explore the impact of different hyperparameters on the customized

input	Output
{"sentence": "A节显示器按键卡死，重启主机后正常", "answers": "部件名称：['视频安防系统']，一级配件名称：['主机软件']，部件一级编号：['A']，故障标签：['事后测试正常']" } {"sentence": "The button on the monitor in section A is stuck, it works normally after restarting the host", "answers": "Component name: ["Video security system"], Level 1 accessory name: ["Host software"], Component Level 1 Number: ["A"], Fault label: ["Post test normal"] "}	{"sentence": "A节显示器按键卡死，重启主机后正常", "answers": "部件名称：['视频安防系统']，一级配件名称：['主机软件']，部件一级编号：['A']，故障标签：['事后测试正常']", "predict": "部件名称：['视频安防系统']，一级配件名称：['主机软件']，部件一级编号：['A']，故障标签：['事后测试正常'] "} {"sentence": "The button on the monitor in section A is stuck, it works normally after restarting the host", "answers": "Component name: ["Video security system"], Level 1 accessory name: ["Host software"], Component Level 1 Number: ["A "], Fault label: ["Post test normal"] ", "predict": "Component name: ["Video security system"], Level 1 accessory name: ["Host software"], Component Level 1 Number: ["A "], Fault label: ["Post test normal"] "}
{"sentence": " A节司机侧座椅不通电，检查为座椅110V控制盒故障，更换控制盒后，库内试验作用良好" , "answers": "部件名称：['司机室座椅']，一级配件名称：['座椅控制器']，部件一级编号：['A']，故障标签：['更换'] " } {"sentence": "section of the driver's side seat does not power, check for seat 110V control box failure, replace the control box, the library test works well", "answers": "Component name: ["Driver's seat"], Level 1 accessory name: ["Seat Controller"], Component Level 1 Number: ["A "], Fault label: ["Replace"] "}	{"sentence": "A节司机侧座椅不通电，检查为座椅110V控制盒故障，更换控制盒后，库内试验作用良好", "answers": "部件名称：['司机室座椅']，一级配件名称：['座椅控制器']，部件一级编号：['A']，故障标签：['更换']。", "predict": "部件名称：['司机室座椅']，一级配件名称：['电机控制盒']，部件一级编号：['A']，故障标签：['更换']。"} {"sentence": "section of the driver's side seat does not power, check for seat 110V control box failure, replace the control box, the library test works well", "answers": "Component name: ["Driver's seat"], Level 1 accessory name: ["Seat Controller"], Component Level 1 Number: ["A "], Fault label: ["Replace"] ", "predict": "Component name ["Driver's seat"], Level 1 accessory name: ["Motor control box"], Component Level 1 Number: ["A"], Fault label: ["Replace"]"}

Fig. 11. Example of the ChatGLM-based customized LLM for knowledge Q&A. "sentence" refers to the combined text that includes the fault content and the corresponding treatment method. "answers" denotes the original annotated text, which comprises the component name, the level 1 accessory name, the component level 1 number, and the fault label. "prediction" indicates the output produced by the model's predictive capabilities, which also includes the component name, the level 1 accessory name, the component level 1 number, and the fault label.

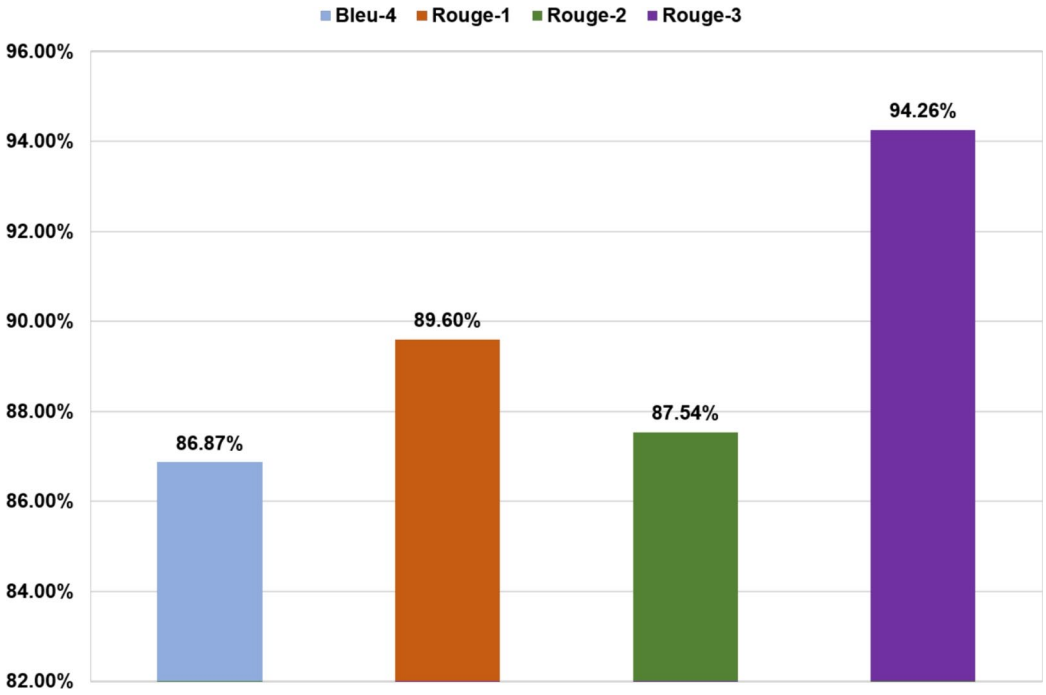


Fig. 12. Histogram of evaluation metrics for customized large language models based on ChatGLM.

Model	Rouge-1/%	Rouge-2/%	Rouge-L/%
Mass[41]	37.7	18.5	34.9
UniLMv2[42]	38.5	19.5	35.8
ChatGLM[38]	38.9	20.0	36.3

Table 6. Knowledge Q&A results based on the Gigaword test set.

Model	Rouge-1/%	Rouge-2/%	Rouge-L/%
BART[43]	45.1	22.3	37.3
T5[44]	40.9	17.3	33.0
ChatGLM[38]	45.5	23.5	37.3

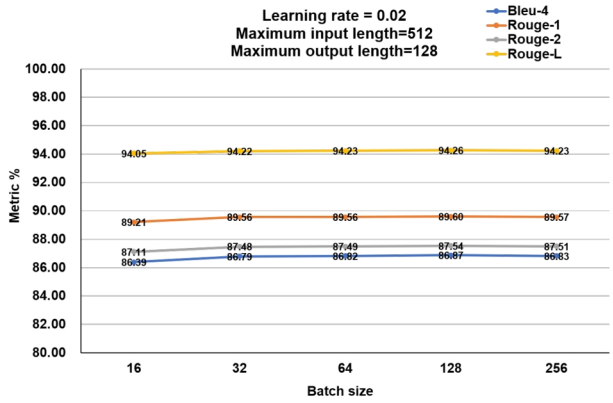
Table 7. Knowledge Q&A results based on the Xsum test set.

llm based on chatglm, this section conducts supplementary experiments based on the settings of previous experiments by respectively changing the learning rate, batch size, maximum input length and maximum output length. The model performance obtained with different hyperparameters is shown in Fig. 13. Through the analysis of the different results in Fig. 13, the performance of the model is shown to be less affected by setting different hyperparameter configurations.

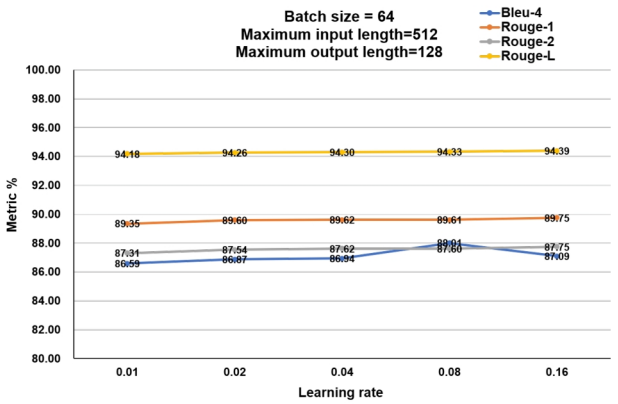
A locomotive maintenance data standardization auxiliary tool

Given the remarkable performance demonstrated by the ChatGLM-based, custom-designed LLM in the field of locomotive maintenance data standardization, this paper presents the development of an auxiliary tool that integrates this LLM and is suitable for real-world applications. The entire process of developing this tool consists of three primary stages: dataset processing, fine-tuning of the LLM, and encapsulation of the results.

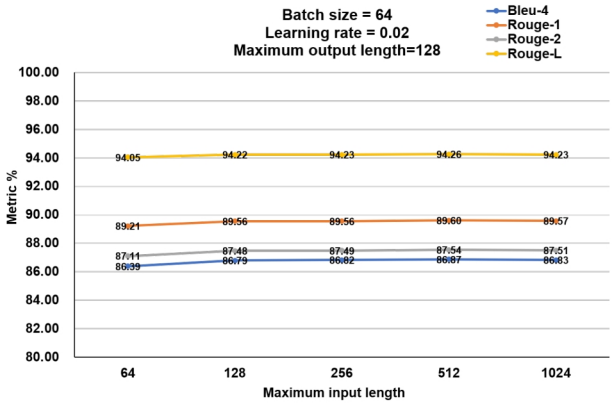
These stages are illustrated in Fig. 14, which offers a comprehensive overview of the tool development process for locomotive maintenance data standardization. The auxiliary tool system comprises a data preprocessing module and an LLM fine-tuning module. The data preprocessing module is responsible for transforming the original locomotive maintenance data into three core files: an index file, a locomotive maintenance dataset and an original dataset. In particular, the locomotive maintenance dataset integrates the fault content and treatment measures of each data item, which is designed to serve the fine-tuning process of the LLM; the original dataset comprehensively retains all the information of the original records to facilitate efficient retrieval through the index file; and the file records the corresponding index of each data item, ensuring that the results extracted



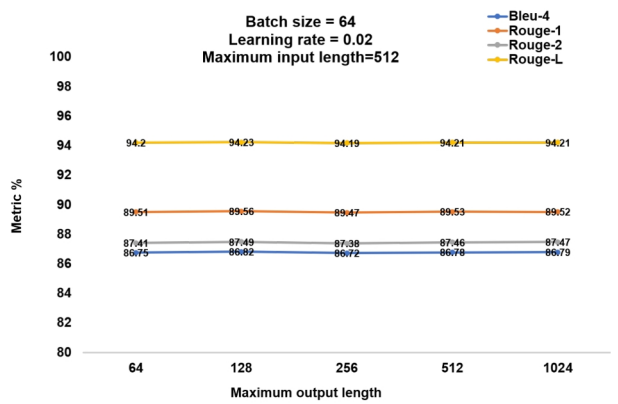
(a) Fix the values of Learning rate, Maximum input length, Maximum output length and change only the value of Batch size.



(b) Fix the values of Batch size, Maximum input length, Maximum output length and change only the value of Learning rate.



(c) Fix the values of Batch size, Learning rate, Maximum output length and change only the value of Maximum input length.



(d) Fix the values of Batch size, Learning rate, Maximum input length and change only the value of Maximum output length.

Fig. 13. Performance of customized models based on different hyperparameter configurations.

by the LLM can be accurately traced back to the relevant contents in the original dataset. The LLM module is based mainly on the ChatGLM framework and uses the locomotive maintenance dataset for P-tuning to obtain the customized LLM for the locomotive maintenance domain. The process of using the auxiliary tool is as follows: first, some locomotive maintenance data are input. Through the data preprocessing module, this data information is subsequently processed into the index file, the locomotive maintenance dataset, and the original dataset. Then, the text in the locomotive maintenance dataset is extracted via the locomotive customized large language model for information extraction. Next, accurate retrieval is performed in the original dataset based on the information in the index file. Finally, the indexing results and the information extraction results are merged as outputs to obtain a complete and accurate final result.

Based on the comparison presented in Fig. 15, the implementation of the locomotive maintenance data standardization auxiliary tool effectively retains comprehensive information related to locomotive maintenance data. This tool facilitates the conversion of verbal maintenance data into standardized records, encompassing details such as the component name, the component level 1 number, the component level 2 number, the level 1 accessory name, and the fault label. Notably, at the ShuoHuang site, the manual standardization of 4,913 records required four person * weeks of labor. In contrast, the use of the auxiliary tool lasted only 15.06 hours (one person * week for review). Consequently, the adoption of this auxiliary tool not only streamlines the standardization process for onsite locomotive maintenance data but also results in considerable time savings. Additionally, it provides a practical solution for standardizing locomotive maintenance practices, ultimately enhancing the analysis of locomotive RCM data.

Intelligent Q&A system for locomotive maintenance

Drawing upon an LLM specifically designed for standardizing locomotive maintenance data, this paper presents the development of an intelligent Q&A system for locomotive maintenance. The workflow of this system, which is based on the customized LLM, is illustrated in Fig. 16. First, the user inputs a question in natural language. The system then segments this question into individual words via Jieba, which facilitates the identification of named entities. These entities are subsequently scored via the customized LLM. Based on the highest-scoring entity, the system retrieves the relevant answer. Next, the system employs big data technology to collect and

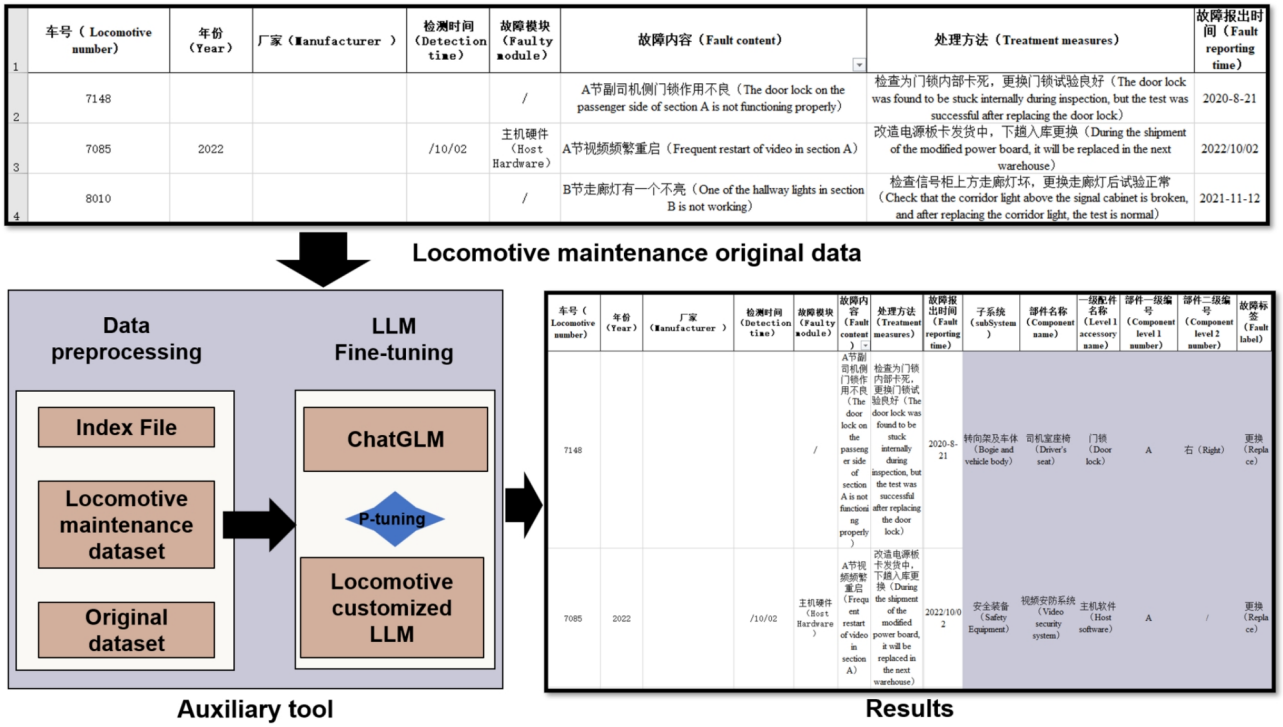


Fig. 14. Workflow of the auxiliary tool: First, some locomotive maintenance data are input. Through the data preprocessing module, this data information is subsequently processed into the index file, the locomotive maintenance dataset, and the original dataset. Then, the text in the locomotive maintenance dataset is extracted via the locomotive customized large language model for information extraction. Next, accurate retrieval is performed in the original dataset based on the information in the index file. Finally, the indexing results and the information extraction results are merged as outputs to obtain a complete and accurate final result.

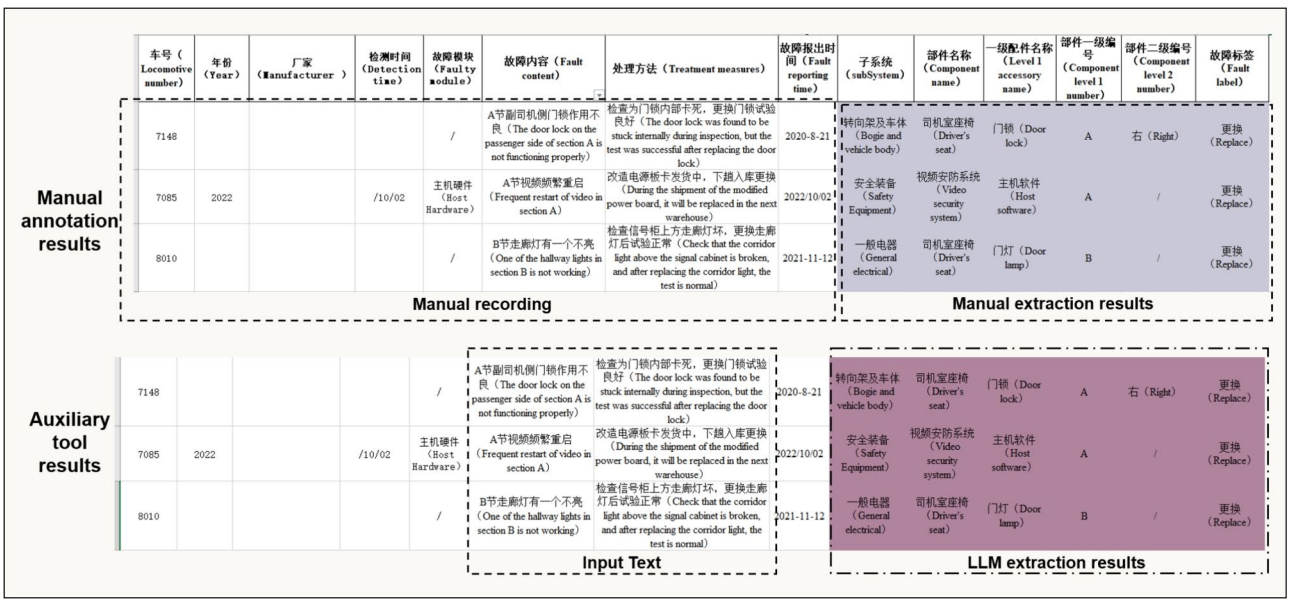


Fig. 15. Comparison of auxiliary tool and manual labeling results.

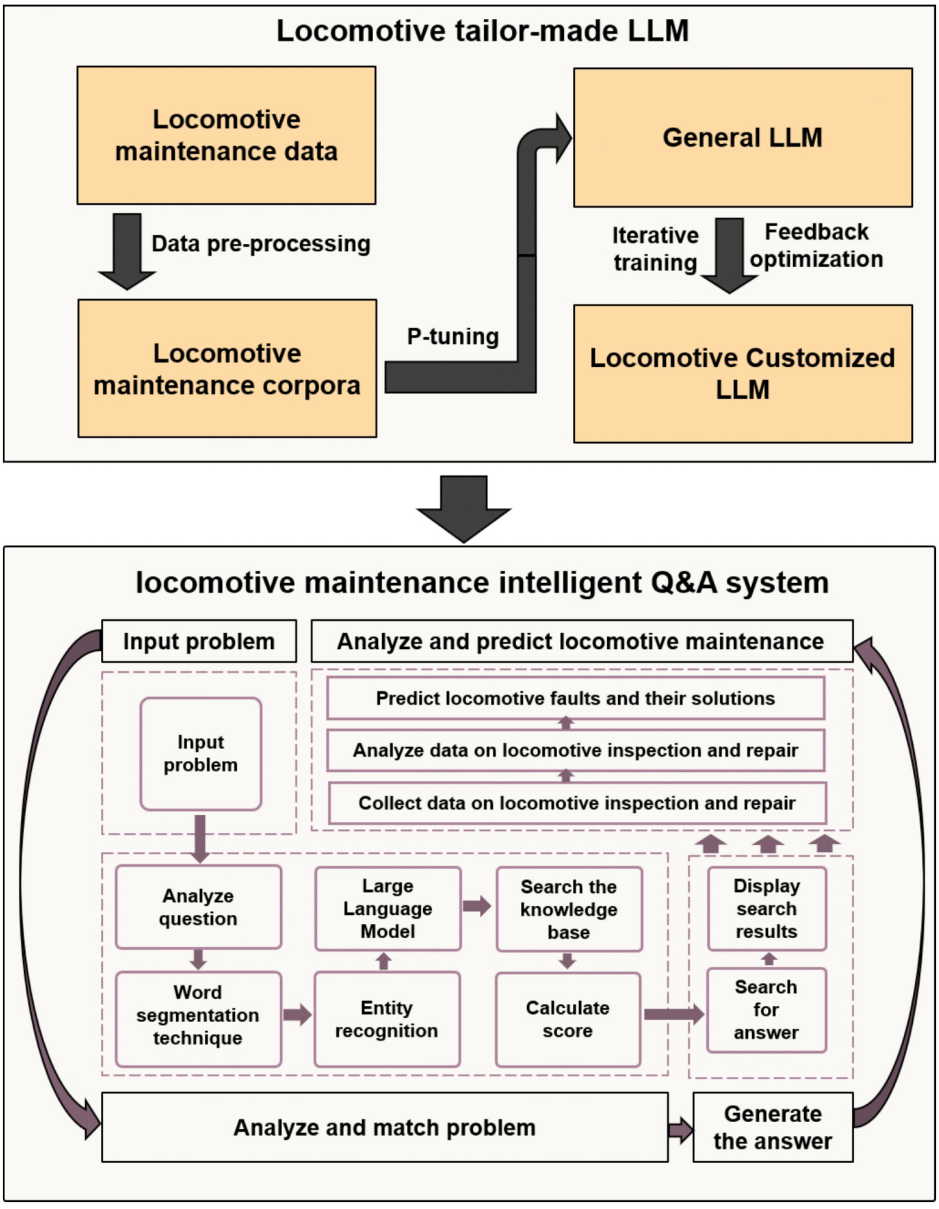


Fig. 16. Workflow of the intelligent Q&A system: First, the user inputs a question in natural language. The system then segments this question into individual words via Jieba, which facilitates the identification of named entities. These entities are subsequently scored via the customized LLM. Based on the highest-scoring entity, the system retrieves the relevant answer. Next, the system employs big data technology to collect and analyze locomotive maintenance data, ultimately providing feedback to the user.

analyze locomotive maintenance data, ultimately providing feedback to the user. An illustrative example of the application of the intelligent locomotive maintenance Q&A system is presented in Fig. 17.

Conclusion

Standardizing the preprocessing of locomotive maintenance data is a crucial step in ensuring data quality for RCM analysis. This paper explores information extraction techniques, knowledge Q&As, and LLMs. Subsequently, we apply UIE and ChatGLM to the standardized preprocessing of locomotive maintenance data, thereby constructing LLMs specifically designed for locomotive maintenance data standardization. This approach considerably improves the efficiency of data standardization tasks.

Within this framework, we investigate the impact of data volume and fault type on the performance of information extraction via UIE. To facilitate this process, we developed a specialized script to automate data annotation. Furthermore, we design an auxiliary tool and an intelligent Q&A system based on a customized LLM for standardizing locomotive data, leveraging ChatGLM. The auxiliary tool can be implemented in real-world scenarios, enabling the conversion of verbal maintenance data into standardized locomotive maintenance records. Moreover, the intelligent Q&A system provides essential maintenance components and corresponding

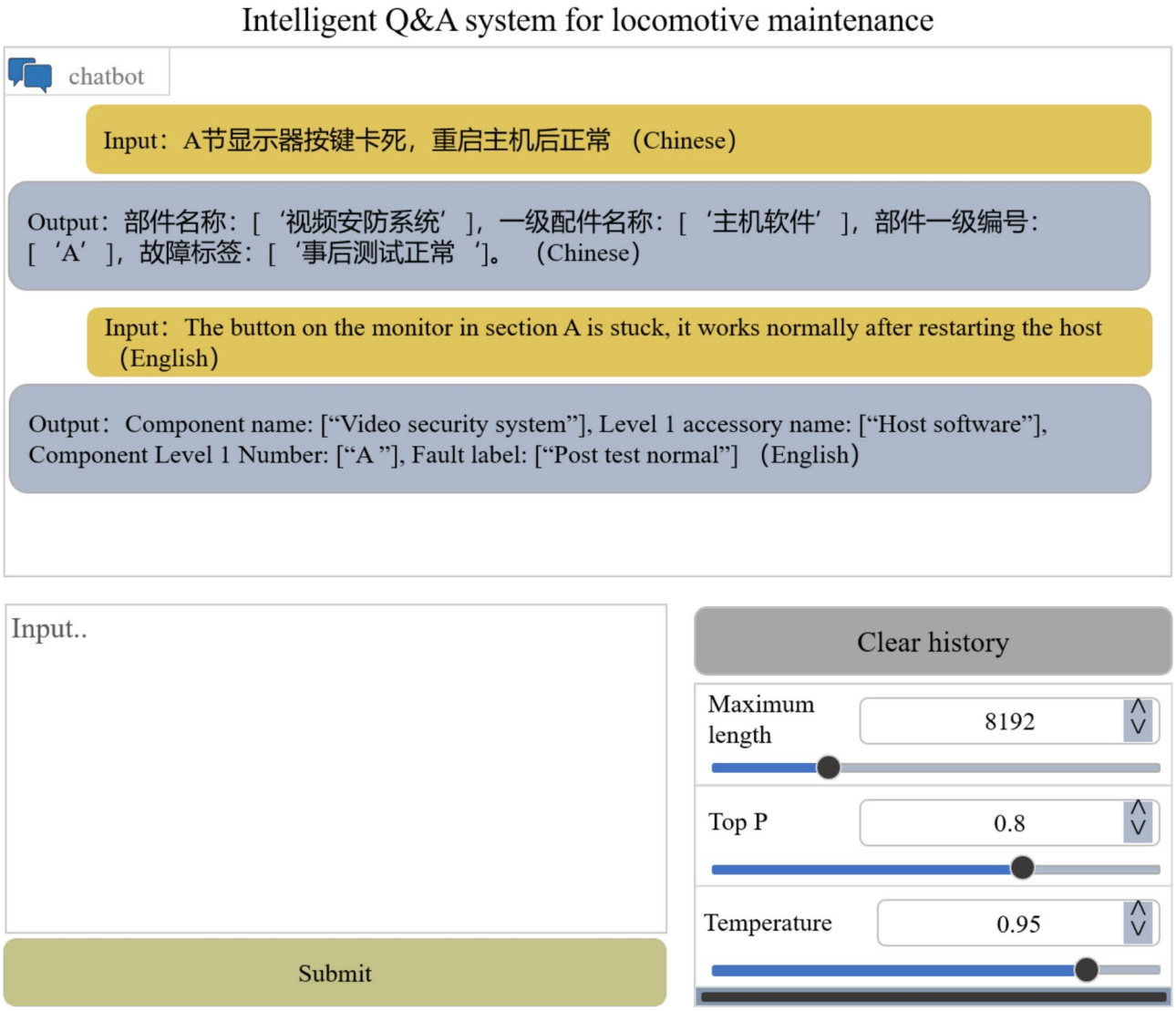


Fig. 17. Example of using the locomotive maintenance intelligent Q&A system.

fault treatment methods. Collectively, these auxiliary tools and the intelligent Q&A system establish a robust foundation for the standardization of locomotive maintenance data.

In this research, the locomotive maintenance data are limited, necessitating the need to expand the dataset as a crucial next step to improve the comprehensive standardization system for locomotive maintenance data. Additionally, the current intelligent Q&A system deviates from traditional Q&A paradigms. To address this issue, future efforts will focus on creating a standardized dataset, which will then be refined and optimized by integrating advanced techniques, including large language models, prompt mechanisms, and retrieval-augmented generation methodologies. Moreover, the automatic data annotation script requires complete locomotive maintenance records; otherwise, it is prone to data bias and other problems. In the future, we will eliminate some serious missing data, supplement simple missing records, and optimize the automatic data annotation script. Furthermore, when the model is applied to larger and more diverse locomotive overhaul datasets, in the future, we plan to use multiple GPUs for parallel training in our training strategy, which will help to reduce the training time for larger datasets, and we also plan to use data augmentation and batch processing, which will help to manage the memory usage and increase the efficiency of the training, and in addition, we will explore the model optimisation techniques such as pruning and quantisation to reduce the computational load without reduce computational load without significantly impacting performance. Simultaneously, the auxiliary tool needs to update the index of new data constantly in the actual deployment, which is a considerable challenge, and in the future, we will optimize the mechanism via prompts. Intelligent Q&A systems must be constantly updated in actual deployment, which has a great demand for energy, such as more powerful or more GPUs to efficiently handle large matrix operations, more GPU memory to load and process data, and more storage space to save models and intermediate results, and in the future, we will integrate energy and continue to conduct in-depth research.

Data availability

The original dataset, extended dataset and clustered dataset data are obtained from CRRC Corporation Limited and Shuohuang Railway Development Limited Liability Company. These datasets contain closed scenarios of various models of railroad cars, and most of the sensitive data must be kept confidential, so most of the original data of the dataset are not open to the public, and only a small amount of the data are openly available (for academic research only): https://github.com/anheqiao-neu/LLM-based-Intelligent-Q-A-System-for-Railway-Locomotive-Maintenance-Standardization. However, the data are available from the corresponding author upon reasonable request.

Accession codes

The data used for this study are published in LLM-based Intelligent Q&ASystem for Railway Locomotive Maintenance Standardization..

Received: 27 November 2024; Accepted: 26 March 2025
Published online: 15 April 2025

References

1. Givoni, M. Development and impact of the modern high-speed train: A review. *Transp. Rev.* **26**, 593–611 (2006).
2. Gupta, G., Mishra, R. & Mundra, N. Development of a framework for reliability centered maintenance. In *Proceedings of the International Conference on Industrial Engineering and Operations Management, Bandung, Indonesia*, 6–8 (2018).
3. Moubray, J. *Reliability-Centered Maintenance* (Industrial Press Inc., 2001).
4. Navair, N. *Guidelines for the Navalaviation Reliability-Centered Maintenance Process* (Naval Air Systems Command, 2005).
5. Campbell, J. D. *The Reliability Handbook* (Clifford-Elliott, 1999).
6. Conachey, R. M. & Montgomery, R. L. *Application of Reliability-Centered Maintenance Techniques to the Marine Industry* (SNAME, 2003).
7. Geisbush, J. & Ariaratnam, S. T. Reliability centered maintenance (rcm): Literature review of current industry state of practice. *J. Qual. Maint. Eng.* **29**, 313–337 (2023).
8. Wang, H., Zhang, L., Ma, X. & Wen, L. Preliminary study on reliability-centered maintenance of high-speed train. In *2009 8th International Conference on Reliability, Maintainability and Safety*, 633–638 (IEEE, 2009).
9. Zhao, W. X. *et al.* A survey of large language models. arXiv preprint arXiv:2303.18223 (2023).
10. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. *et al.* Improving language understanding by generative pre-training (2018).
11. Radford, A. et al. Language models are unsupervised multitask learners. *OpenAI Blog* **1**, 9 (2019).
12. Brown, T. et al. Language models are few-shot learners. *Adv. Neural. Inf. Process. Syst.* **33**, 1877–1901 (2020).
13. Achiam, J. *et al.* Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023).
14. Touvron, H. *et al.* Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023).
15. Taori, R. *et al.* Stanford alpaca: An instruction-following llama model (2023).
16. You, Y. colossalchat: An open-source solution for cloning chatgpt with a complete rlhf pipeline (2023).
17. Chiang, W.-L. *et al.* Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See https://vicuna.lmsys.org (accessed 14 April 2023)**2**, 6 (2023).
18. Grishman, R. Twenty-five years of information extraction. *Nat. Lang. Eng.* **25**, 677–692 (2019).
19. Cowie, J. & Lehnert, W. Information extraction. *Commun. ACM* **39**, 80–91 (1996).
20. Tartakovskyi, E., Ustenko, O., Puzyr, V. & Datsun, Y. *Systems Approach to the Organization of Locomotive Maintenance on Ukraine Railways, 217–236* (Springer International Publishing, 2017).
21. Hodgson, T. *Locomotive Management-Cleaning, Driving and Maintenance* (Read Books Ltd, 2013).
22. Piskorski, J. & Yangarber, R. Information extraction: Past, present and future. *Multi-source, Multilingual Information Extraction and Summarization* 23–49 (2013).
23. Bikel, D. M., Miller, S., Schwartz, R. & Weischedel, R. Nymble: A high-performance learning name-finder. arXiv preprint cmp-lg/9803003 (1998).
24. Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J. & Scholkopf, B. Support vector machines. *IEEE Intell. Syst. Appl.* **13**, 18–28 (1998).
25. Settles, B. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications (NLPBA/BioNLP)*, 107–110 (2004).
26. Zhang, Y. & Yang, J. Chinese ner using lattice lstm. arXiv preprint arXiv:1805.02023 (2018).
27. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).
28. Sun, Y. *et al.* Ernie: Enhanced representation through knowledge integration. arXiv preprint arXiv:1904.09223 (2019).
29. Wang, M. et al. A survey of answer extraction techniques in factoid question answering. *Comput. Linguist.* **1**, 1–14 (2006).
30. Riloff, E. & Thelen, M. A rule-based question answering system for reading comprehension tests. In *ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems* (2000).
31. Yani, M. & Krisnadhi, A. A. Challenges, techniques, and trends of simple knowledge graph question answering: A survey. *Information* **12**, 271 (2021).
32. Sharma, Y. & Gupta, S. Deep learning approaches for question answering system. *Procedia Comput. Sci.* **132**, 785–794 (2018).
33. Shanahan, M. Talking about large language models. *Commun. ACM* **67**, 68–79 (2024).
34. Chen, J. et al. When large language models meet personalization: Perspectives of challenges and opportunities. *World Wide Web* **27**, 42 (2024).
35. Wei, W. *et al.* Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 806–815 (2024).
36. Borji, A. A categorical archive of chatgpt failures. arXiv preprint arXiv:2302.03494 (2023).
37. Lu, Y. *et al.* Unified structure generation for universal information extraction. arXiv preprint arXiv:2203.12277 (2022).
38. Du, Z. *et al.* Glm: General language model pretraining with autoregressive blank infilling. arXiv preprint arXiv:2103.10360 (2021).
39. Rush, A. M., Chopra, S. & Weston, J. A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015).
40. Narayan, S., Cohen, S. B. & Lapata, M. Don't give me the details, just the summary. *Topic-Aware Convolutional Neural Networks for Extreme Summarization ArXiv, abs* (1808).
41. Song, K., Tan, X., Qin, T., Lu, J. & Liu, T.-Y. Mass: Masked sequence to sequence pre-training for language generation. arXiv preprint arXiv:1905.02450 (2019).
42. Bao, H. *et al.* Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International Conference on Machine Learning*, 642–652 (PMLR, 2020).

43. Lewis, M. *et al.* Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019).
44. Raffel, C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**, 1–67 (2020).

Author contributions
Ao Chen conceived and conducted the experiments, and Ao Chen and Ye Tian analyzed the results. Ao Chen, Ye Tian, Jinyi Zhang, Chen Li and Huiyuan Zhang reviewed the manuscript.

Declarations

Competing interests
All authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to A.C.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.