Tsinghua Science and Technology
Doi: To be confirmed

Research Article

Information retrieval in pre-hospital care with visualization-oriented natural-language interface via LLMs

Xin Gao[1,2], Zhengye Zhu[1,2], Xinyu Ma[5], and Yasha Wang[1,4] (✉)
Junfeng Zhao[1,3], Xu Chu[1,3], and William Cheng-Chung Chu[6] (✉)

Abstract With the popularization of Electronic Health Records (EHR), the emergency system has stored a large number of historical dispatch records, which can provide valuable insights for the optimization of current pre-hospital care. However, the inconvenient interaction manner of current information retrieval systems hinders researchers from exploring these historical records. To address this issue, we propose a novel framework that leverages the language understanding and code generation ability of Large Language Models (LLMs) to build an information retrieval system with Visualization-oriented Natural-language-based Interfaces (V-NLI). To incorporate both domain-specific and task-related prior knowledge, we generate the instruction datasets based on the ability of closed-source LLMs in a multi-stage manner and conduct supervised fine-tuning on open-source LLMs. We also devised various mechanisms for augmenting the capabilities of open-source LLMs in query interpretation and code generation. To validate the effectiveness and generalizability of our framework, we conducted experiments on a public dataset NLV. More significantly, we performed more detailed experiments on a dataset including over 1 million pre-hospital emergency historical records in ten years. The performance of our method surpasses all baseline methods and achieves comparable results even with some SOTA closed-source models.

Keywords pre-hospital emergency care; table visualization; natural language interface; large language models

1 National Engineering Research Center for Software Engineering, Peking University, Beijing 100871, China.
2 Department of Computer Science, Peking University, Beijing 100871, China. E-mail: xingao@pku.edu.cn, zzy99@pku.edu.cn.
3 Key Laboratory of High Confidence Software Technologies, Ministry of Education, Peking University, Beijing 100871, China. E-mail: zhaojf@pku.edu.cn, chu_xu@pku.edu.cn.
4 Peking University Information Technology Institute (Tianjin Binhai), Tianjin, China. E-mail: wangyasha@pku.edu.cn.
5 Seed, ByteDance, Beijing 100098, China. E-mail: maxinyu@pku.edu.cn.
6 School of Computing and Artificial Intelligence, Fuyao University of Science and Technology, Fuzhou 350109, China. E-mail: cchu.william@gmail.com.
* To whom correspondence should be addressed. E-mail: wangyasha@pku.edu.cn; cchu.william@gmail.com.
Received/revised/accepted: To be confirmed.

1 Introduction

Pre-hospital emergency care plays a critical role in emergency medical service systems [1–4]. Throughout the evolution of pre-hospital emergency care, the storage paradigm for historical records in numerous cities (e.g., Shenzhen, China) has transitioned from the initial distributed storage across various hospitals to a centralized approach within emergency centers, which facilitates the analysis of emergency events from a macro perspective. The historical records contain information about patients' health conditions, spatiotemporal distribution of emergency events, the process of emergency dispatches, etc. Combining with other data resources in cities, researchers can analyze these data to (1) understand the spatiotemporal occurrence patterns of various diseases, (2) optimize emergency resource allocation, and improve the quality of emergency services. For example, researchers can observe the spatiotemporal distribution of patients with different diseases in cities. These observations enable investigating the impact of different independent variables (e.g., environmental pollution, population density, urban infrastructure, etc.) on the probability of emergency events caused by specific diseases, thereby optimizing resource allocation in emergency grading. In addition, researchers can also analyze the impact of various independent variables (e.g., transportation, patient location, etc.) on emergency response time to optimize emergency response efficiency and improve the quality of emergency services. **A vital requirement in these studies is to associate historical emergency records with specific independent variables and present the results vividly. This poses huge challenges to existing information retrieval systems.**

In the realm of information retrieval systems, users are typically required to perform specific operations on a designated platform to translate their demands into interactions with panels and buttons, thereby accessing relevant tabular data [5, 6]. However, the existing information retrieval systems still CANNOT meet the usage demands in pre-hospital care. **Specifically, on the input side, there are a wide variety of independent variables from various categories of city data.** On the one hand, panels with pre-defined buttons

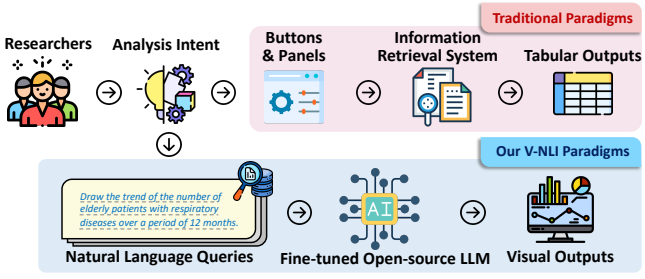

Fig. 1 Comparison between the traditional paradigm of information retrieval systems and our V-NLI-based paradigm.

may struggle to accommodate diverse demands. On the other hand, interactions based on programming languages impose higher degrees of freedom, which brings a significant usage burden to researchers. The consensus is that data analysis is most effective when users can concentrate on their data rather than manipulations on the interfaces [7, 8]. Therefore, a flexible and efficient manner of interaction, such as natural language, is very necessary in the information retrieval system of pre-hospital care. **On the output side, researchers need to observe the data from various perspectives.** Displaying the same data in different ways may lead to completely different results. Therefore, compared to only presenting data in tabular forms, various illustrative chart formats can help researchers extract insights from historical data in an efficient way [9]. Consequently, we characterize the information retrieval task in pre-hospital emergency care as a Visualization-oriented Natural-language-based Interfaces (V-NLI) problem [10, 11]. As illustrated in figure 1, this paradigm takes users' natural language expressions of intent as input and produces visualizations of the retrieved tabular data as output.

Nonetheless, crafting and implementing V-NLI face challenges attributable to **the ambiguous and under-specified nature of human language in query interpretation**, as well as the **intricacies involved in generating correct codes for data transformation** [11, 12]. Over the past two decades, in response to these challenges, numerous systems have emerged through academic research and commercial software development [13–15]. However, due to the limited capability of Natural Language Processing (NLP) tools [16, 17], the efficacy of these methods still faces significant limitations. Thanks to the advent of pre-trained language models, their incredible performance across various NLP tasks presents significant opportunities for enhancing the intelligence of V-NLI [18, 19].

There are already substantial works dedicated to utilizing large language models(LLMs) in the automated analysis of tabular data [20–25]. However, these works still face limitations when directly applied to information retrieval in pre-hospital emergency care. Primarily, most such works [20, 21, 23] currently rely on the in-context learning ability of closed-source LLMs, such as GPT-3.5 [26] and GPT-4 [27], necessitating users to upload data to third-party service providers during usage. However, given that pre-hospital emergency care data often contains sensitive patient information, uploading them to external servers poses risks of privacy breaches [28]. Furthermore, some works attempted to use open-source LLMs. These works [21, 25, 29] are dedicated to addressing various table-related tasks in general domain data, thereby lacking the incorporation of domain-specific and task-related prior knowledge relevant to pre-hospital care. Specifically, domain-specific prior knowledge in pre-hospital emergency care may pertain to some established conventions on under-specified utterances such as "Emergency Response Time" or "Critical Patients". Moreover, task-related prior knowledge may consist of some demonstrations or documents related to various visualization functions. **Incorporating both kinds of prior knowledge can assist LLMs to infer the intentions of researchers and generate accurate data transformation codes.** However, two challenges still remain in leveraging open-source LLMs to V-NLI-based information retrieval in pre-hospital care.

Challenge 1: How to incorporate the prior knowledge into open-source LLMs? Performing supervised fine-tuning with appropriate datasets is an effective way to adapt LLMs to downstream tasks [30, 31]. Although well-established datasets exist in the domain of V-NLI [13, 32], these datasets have simple structures and come from other general domains, which is not compatible with real-world emergency scenarios. Consequently, our first challenge is obtaining an instruction dataset based on the provided emergency historical records to enable LLMs to acquire domain-specific and task-related prior knowledge.

Challenge 2: How to further enhance the code generation ability of open-source LLMs? In comparison to closed-source LLMs which have achieved remarkable performance, open-source LLMs still exhibit a significant capability gap particularly in tasks related to code generation, due to their limited model capacity [33]. Furthermore, this capability gap becomes more pronounced when utilizing relatively smaller LLMs, such as Mistral-7B [34] or BaiChuan-13B [35]. Hence, our second challenge is to improve the capabilities of open-source LLMs in code generation.

To solve the first challenge, we have designed a framework that utilizes the ability of closed-source LLMs to generate instruction datasets for the supervised fine-tuning of open-

source LLMs in a multi-stage manner. We exclusively furnish the table header fields to the closed-source LLMs to mitigate the risk of privacy breaches. To overcome the second challenge, we have devised various mechanisms to assist open-source models in decomposing complex problems into a series of fixed-step subproblems, thereby reducing the complexity of a single task. Simultaneously, we have implemented a correction feedback mechanism to enhance the accuracy of the model in generating data transformation and visualization codes. To validate the effectiveness and generalizability of our framework, we conducted experiments on the publicly available dataset NLV. More significantly, we performed more detailed experiments on real-world pre-hospital emergency historical records. We implemented and deployed an information retrieval system supporting V-NLI in a real clinical setting. The experimental results demonstrate the effectiveness of our framework in improving the accuracy of information retrieval in pre-hospital emergency care. The performance surpasses all baseline methods and achieves comparable results with some SOTA closed-source models.

The contribution of this paper can be concluded as follows:

- To assist analysis of historical emergency records, we have devised a framework to build a V-NLI-based information retrieval system tailored for pre-hospital emergency care, which holds great significance, yet has been seldom explored.
- In applying open-source LLMs to this task, we formulated a framework to concurrently incorporate domain-specific and task-related knowledge into LLMs.
- Experiments on public datasets demonstrate the applicability of our framework across various general domains. Our experiments on a real-world dataset including over 1 million pre-hospital emergency historical records in ten years also show that our framework surpasses all baseline methods.

2 Related Works

2.1 Analyzing Pre-hospital Emergency Records

Pre-hospital care is a pivotal facet of emergency medicine, encompassing the provision of medical assistance to patients before their arrival at a hospital or healthcare facility [36]. The adoption of electronic health records serves to standardize the collection of historical records, facilitating seamless data sharing between pre-hospital care providers and researchers. Through the analysis of pre-hospital care interventions and outcomes documented in historical records, researchers can discern areas for enhancement, assess the efficacy of novel treatments, and contribute to the development of evidence-based practice guidelines [37, 38]. A critical challenge encountered in pre-hospital care research lies in the intricate process of selecting a homogeneous study group [36, 39]. The distinctive nature of pre-hospital care, marked by a diverse patient population and a spectrum of emergency scenarios, introduces significant complexities for researchers in forming a cohesive and homogeneous study group. Hence, the establishment of a dedicated information retrieval system for pre-hospital emergency care, aiding researchers in efficiently retrieving homogeneous populations that meet specific criteria, assumes paramount importance. However, this matter has been scarcely investigated.

2.2 Visual-oriented Natural Language Interface

To alleviate the learning burden on researchers regarding the new system, V-NLI is imperative for the information retrieval system of emergency care. Over the past two decades, various systems have been developed in both academic research and commercial software to address these challenges, with a notable surge in recent years. As far back as 2001, Cox [10] introduced an initial prototype of V-NLI, capable of accepting well-structured queries. The emergence of word embeddings [40] spurred advancements in neural networks for Natural Language Processing, renewing interest in V-NLI [41–44]. With recent progress in natural language processing, efforts have been made to leverage deep learning-based language models for visualization. For instance, ncNet [45] utilized a Transformer-based sequence-to-sequence model to convert natural language queries into visualizations. However, the efficacy of these methods faces significant limitations in two crucial steps: query interpretation and data transformation. Query interpretation, being fundamental to all stages, poses challenges for current language parsers due to their limited capability and the ambiguous nature of natural language. The challenge of data transformation primarily arises from the difficulty of generating correct codes based on the user intent derived from previous stages. Recently, LLMs have demonstrated promising abilities in language comprehension and code generation, offering new prospects to address the challenges in V-NLI for information retrieval.

2.3 Large Language Models for Tabular Data

In recent times, there have been notable strides in the development of LLMs, exemplified by closed-source models like GPT-3 [26] and GPT-4 [27], alongside open-source models such as LLaMA [33] and BaiChuan [35]. These models have found application across diverse domains, including code

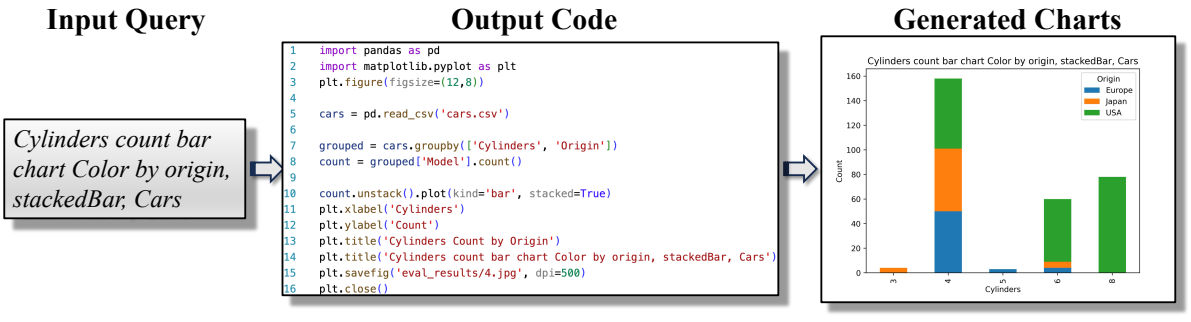

Fig. 2 An example of the input query, output code, and generated chart after executing the code.

generation [46], story generation [47], and web design [48]. Specifically, recent studies have explored utilizing LLMs for tabular data analysis. CHAT2VIS [49], for instance, generates Python visualization code by providing LLMs with table schema, column types, and natural language queries. GPT4-Analyst [50] proposes a framework utilizing prompts to guide GPT-4 in tasks such as data collection, visualization, and analysis. Data-Copilot [24] exhibits the ability to generate requests, select necessary interfaces, and invoke corresponding interface tools sequentially or in parallel without necessitating code. Nevertheless, all these works are reliant on prompt engineering and are contingent on closed-source models like GPT-3 and GPT-4, introducing potential security concerns. Some works [21, 25, 29] have explored the use of open-source models; however, they have predominantly focused on various table-related tasks within general domain data. Lacking domain-specific and task-related prior knowledge, their performance on the V-NLI task in specific domains falls short due to the constrained capacity of open-source LLMs. In contrast to preceding efforts, targeting the V-NLI task for an information retrieval system within the domain of pre-hospital emergency care, we have devised a framework to concurrently infuse domain-specific and task-related knowledge into open-source LLMs.

2.4 Model Finetuning and Problem Decomposition

Fine-tuning has become a pivotal technique for adapting pre-trained language models to specific tasks or domains [51]. Notably, Radford et al. [52] introduced the concept of fine-tuning large transformer models, demonstrating their versatility across diverse domains, from language generation to classification tasks. Fine-tuning LLMs on domain-specific datasets, such as electronic health records [53] or emergency response protocols, allows models to better understand specialized vocabulary and contextual relationships inherent in these domains. However, fine-tuning also comes with challenges, including the risk of overfitting and the need for large annotated datasets. Despite these challenges, fine-tuning remains an essential tool in adapting LLMs to real-world applications, such as pre-hospital emergency care systems, where accuracy and reliability are paramount.

Problem decomposition is an emerging area of research aimed at breaking down complex tasks into smaller, more manageable sub-tasks, enabling more effective model reasoning and improved performance on intricate challenges. For LLMs, this approach involves structuring complex queries or data-related tasks into simpler steps that the model can handle more efficiently. Wei et al. [54] demonstrated that problem decomposition techniques could significantly improve the performance of LLMs in tasks requiring multi-step reasoning or complex calculations. Specifically, problem decomposition has been utilized in areas such as arithmetic problem-solving [55] and decision-making systems [56, 57], where tasks are broken down into logical components that can be addressed independently.

3 Methodology

To build information retrieval systems with V-NLI in specific domains such as pre-hospital care, we propose a framework to incorporate domain-specific and task-related prior knowledge into open-source LLMs via supervised fine-tuning on specific instruction datasets. Here, we detail our framework, which refers to the pipeline of generating and filtering the instruction dataset and then conducting supervised fine-tuning on the generated dataset with several mechanisms in order to enhance the ability of LLMs.

3.1 Defining Problem Setting and Instructions

We formulate our problem based on the settings in this survey [11]. Given a tabular dataset D, which usually consists of multiple tables, we hope to build a model that can transform analysis intent in natural language n_i into corresponding visualized figure f_i. Our core target is to fine-tune an open-source LLM that takes natural language as input and

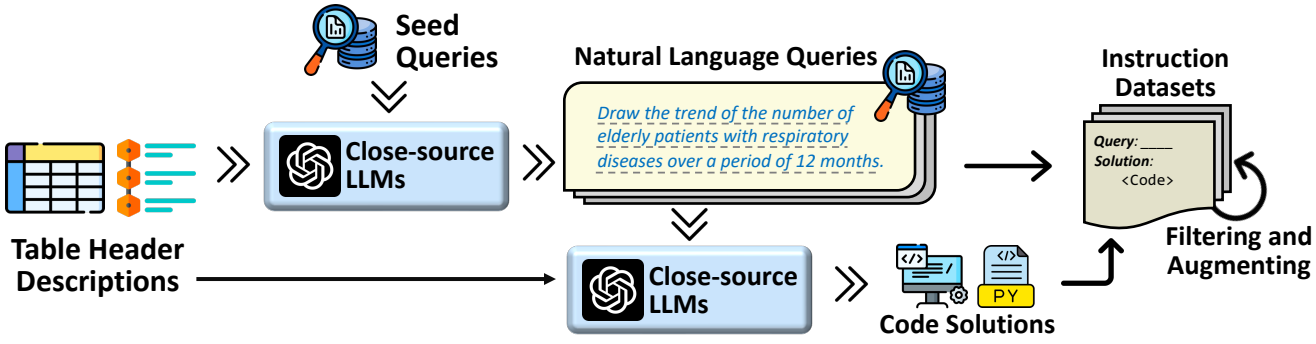

Fig. 3 The pipeline of our framework for generating instruction dataset based on the given tabular data in the specific domain.

generates codes for data retrieval, data transformation, and visualization as output. An Example in the NLV dataset can be found in Figure 2. The instruction can be formulated as n_i, c_i where c_i is the code that can be executed to extract target data from the raw tabular dataset and generate corresponding visualized figure f_i.

3.2 Instruction Data Generation

A well-annotated large-scale instruction dataset is necessary for the supervised fine-tuning of LLMs. The choice of instruction datasets can have a significant influence on fine-tuned LLMs' capability. Though there have been plenty of public instruction datasets, their tasks and domains differ from our setting, which are hard for LLMs to learn relative prior knowledge in our task. Therefore, our first challenge is to obtain a suitable instruction dataset that contains demonstrations transforming analysis intent into corresponding codes. However, collecting such a dataset is challenging for humans because it requires both 1) medical expertise to come up with various analysis intent as queries and 2) programming expertise to write code solutions to each instruction. Hence, we have designed a framework that leverages the ability of closed-source LLMs to generate instruction datasets as shown in figure 3. Our framework consists of three stages: *Natural Language Query Generation*, *Code Solution Generation*, and *Dataset Filtering*.

3.2.1 *Generating Natural Language Queries*

We first generate the natural language queries that contain researchers' potential analysis intents. Our goal is to generate **practical** and **various** queries so that LLMs can learn the underlying patterns from different queries. To ensure the practicality of queries, we ask the closed-source LLMs to generate queries based on *descriptions of table headers* and **seed queries**. As for the variety, we require the LLMs to learn from the **seed queries** and generate new queries different from the provided seeds via prompt. The template of prompts is shown in the upper part of Table 1.

Table Header Descriptions: Table headers, or column names in other words, usually contain information that describes the actual meaning of each column in the table. However, these table headers are usually named in a simplified manner for convenience. It can be even difficult for domain experts to understand the actual meaning of each column only via table headers. Therefore, detailed descriptions of table headers are helpful for LLMs to understand the data, which can effectively prevent LLMs from generating queries that involve attributes not in the table. It should be noted that we do not provide real tabular data to closed-source LLMs but only the descriptions due to the consideration of data privacy.

Seed Queries: We provide the closed-source LLM with some seed queries as demonstrations so that the generated queries could be more practical. Besides, we also hope the vocabulary and structure of generated queries to be similar to actual queries. Therefore, we use the existing queries for public datasets and actual queries written by researchers for pre-hospital care in the experiments.

3.2.2 *Generating Code Solutions*

After generating plenty of queries via closed-source LLMs, we now still leverage the same LLM to generate corresponding code solutions. Table header descriptions are still very helpful in this step since they provide the relationships between the column name and its actual meaning. For example, it could be helpful for LLMs if the description for a column named "DiseaseType" includes its possible values such as "Respiratory Disease". Moreover, we also provide LLMs with several demonstrations of query-code pairs with a consistent and concise code style so that the generated codes could be easy for open-source LLMs to learn from. After collecting the code solutions for each query, we obtain our raw instruction dataset. The template of prompts is shown in the bottom part of Table 1.

Table 1 Templates of prompts in instruction data generation

Instruction: There are several tables as following: [Table Schema and Header Descriptions for each table.] **Instruction:** Now you are a data analyst and want to retrieve information from the table via visualization. Please generate some queries based on the following demonstrations: [Query Demonstrations] **Requirements:** Only provide each generated query without explanation. The generated queries should be different from each other. The generated queries should have diversity in sentence structure, contents, and chart categories. Generated queries cannot involve fields that are not present. Generated queries need to have practical significance. The intention of queries should match the chart category.
Instruction: There are several tables as following: [Table Schema and Header Descriptions for each table.] **Instruction:** Now you are a data analyst and want to retrieve information from the table via visualization. Please use the *pandas* and *matplotlib* in Python to complete the following queries. [Input Queries] **Instruction:** Here are some demonstrations of query-code pairs. [Query-code pair demonstrations] **Requirements:** You need to deeply understand the intention behind the query. Generate codes for beautiful charts and avoid crowding. When setting the color based on a certain column, convert it if not numerical. The generated code of each query needs to contain its own library import and data loading part. You need to split the query into steps before giving the code and then explain the idea of each step. Only after explaining the steps can you give the complete code.

3.2.3 Filtering and Augmenting Instructions

As pointed out in LLaMA-2 [33], not only the number of instructions but also the quality can have large influences on supervised fine-tuning. However, it should be noted that even the most intelligent LLMs such as GPT-4 can not successfully follow the instructions and generate correct codes for each query. Therefore, there could be two potential problems in our collected raw dataset: *insufficient amount* and *low quality*. On the one hand, we have observed that some generated code solutions can not be executed successfully. Such instructions should be deleted from the dataset. Besides, the amount of generated instructions can be limited due to the concerns on cost. On the other hand, the quality of instructions can not be guaranteed. There could be repetitive and abnormal instructions in the dataset. Filtering out such instructions can greatly improve the quality of the whole dataset. As a result, we propose to iteratively perform *filtering* and *augmenting* operations on the instruction dataset to remove instructions in low quality and make sure enough amount of instructions.

Augmenting: This operation makes minor rule-based modifications on both the query and its corresponding code to generate a new instruction. For instance, given a query that *Draw the trend of the number of elderly patients with respiratory disease over a period of 12 months* and its corresponding code solution, we can easily generate an augmented query-code pair via replacing *"respiratory disease"* with *"truma"* in the query and replacing relative lines about *selection* operation in the code. We also can replace *"a period of 12 months"* with *"a period of 24 hours"* in the query and the relative lines about *"aggregation"* operation in the code to get another augmented instruction. It should be noted that instructions generated in this stage are much similar compared with those generated by LLMs since their queries and codes differ only in several words and lines. Nevertheless, the augmenting operation is still important for ensuring the amount of instructions is enough for supervised fine-tuning at low time and money cost.

Filtering: This operation aims to improve the quality of the instruction dataset by filtering out those low-quality instructions. For each instruction, we begin by removing irrelevant texts and extracting codes from the responses of closed-source LLMs to refine the instruction. Then, we delete instructions that contain impractical operations in queries such as involving attributes that do not appear in the table or using complex diagram categories that are rarely used (e.g. butterfly diagram). Besides, we also remove instructions with duplicated queries measured by edit distance. It should be noted that we reserve instructions whose queries are different in edit distance but have similar semantics or similar code solutions. We believe these similar instructions can help open-source LLMs to better understand input queries. Finally, we delete instructions that do not contain code solutions that can be executed successfully.

3.3 Enhancement for open-source LLMs

After the aforementioned process, we finally obtain our instruction dataset. Via supervised fine-tuning, open-source LLMs can not only learn to follow the instructions to generate codes for information retrieval but also the domain-specific and task-related prior knowledge contained in the instructions. However, we found that only with supervised fine-tuning, open-source LLMs still could not achieve satisfying performance on our tasks. We speculate that one possible reason is that the ability of open-source models is limited by their own capacity compared with large closed-source LLMs. Therefore, inspired by human programming habits,

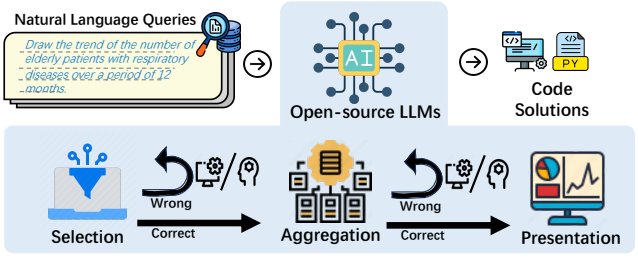

Fig. 4 The basic pipeline of our two mechanisms to improve the code generation ability of LLMs.

we devise two tricks to assist open-source LLMs with generating code solutions. Figure 4 illustrates the basic insight of these two mechanisms.

3.3.1 Problem Decomposition:

When facing complex problems, experienced programmers usually decompose the workflow into several sub-tasks. Specifically, the workflow of V-NLI-based information retrieval can be broken down into three steps: *Selection*, *Aggregation*, and *Presentation*. *Selection* means not only understanding the query and selecting relative attributes from all columns but also filtering out rows via predicates, such as greater than or less than, that meet the requirements in the query, which can be seen as extracting a sub-table from the raw table. *Aggregation* can be seen as transforming the sub-table into other views. It means to apply some function such as count or sum and reorganize the sub-table following the query. *Presentation* means to map the transformed data into visual structures and present them in graphical form. Besides adding these requirements into prompts and demonstrations, we also add some instructions in the training dataset so that the open-source LLM can also learn this pattern during supervised fine-tuning.

3.3.2 Correction Feedback:

Due to the strict grammar of programming languages, it is common that generated codes can not be executed successfully. Even experienced programmers often write programs containing bugs. Good codes are completed through continuous error correction and modification by programmers. Consequently, instead of generating codes only in one turn, we hope LLMs to learn from the previous bugs and generate correct codes. When the generated codes can not be executed, we will input the incorrect code and its error message into the LLM to generate codes again. Due to the context limit, we delete some demonstrations in prompts to incorporate the error message. Another kind of effective feedback can be interactions with human experts in natural language. Such a human-in-the-loop manner can effectively leverage experts' experience in code generation. It should be noted that the effectiveness of human feedback heavily depends on the multi-turn conversational ability of base LLMs.

4 Experiments on NLV Dataset

4.1 Dataset

To validate the effectiveness and generalizability of our framework, we first conduct experiments on a public dataset, NLV. NLV [32] is a small benchmark dataset for the evaluation of V-NLI models. The dataset contains 814 utterance sets from 3 data tables over 30 visualizations curated from an online study. An utterance set is defined as a collection of one or more utterances that collectively map to a specific visualization. We acknowledge that NLV is a relatively small dataset compared with other datasets such as nvBench [58]. The reasons for choosing NLV are from two aspects. On the one hand, our final goal is to achieve a V-NLI-based information retrieval system for pre-hospital care. Experiments on such simple public datasets are conducted just to preliminarily validate our framework and prove its generalizability over other tables. As we will introduce in Section 5, real-world historical datasets are much more sophisticated than all these public datasets. On the other hand, we DO NOT aim at devising a model that can achieve V-NLI-based information retrieval on various domains. We hope that our model can specialize in the given tables. NLV dataset is more related to our settings since all the utterances focus on three given tables. For each line in the original table containing utterances, we use commas to connect three fields, *"Utterance Set"*, *"visId"* and *"Dataset"*, to form a natural language query. We select 60 utterances as seed queries and test our model and other baselines on the remaining utterances, denoted as NLV_{ori}. We also keep 90 queries generated in the construction of our instruction dataset as an extra test set to validate the generalizability, denoted as NLV_{gen}. We carefully check the training instruction dataset to make sure that there are no similar queries in NLV_{gen}.

4.2 Implementation Details

We implement our framework based on two open-source LLMs, LLaMA2-13B [33] and Mistral-7B [34]. We choose the GPT-3.5-turbo as the closed-source LLM to construct our instruction dataset. We note that GPT-4 can be a better but much more expensive choice. Our experiments are conducted on a machine equipped with 2 Nvidia RTX8000 GPUs and Intel Xeon Gold 6230 CPUs. The main used libraries include Python 3.10.13, Pytorch 1.13.1, Transformers 4.36.2, PEFT 0.7.1, and Deepspeed 0.12.6. Without loss

of generality, we choose to generate Python codes for the whole task. We note that other professional programming languages such as SQL and VegaLite can also be used. We generated 10364 queries initially. After generating code solutions, 7516 query-code pairs remain. We finally obtained an instruction dataset including 6092 samples after two iterations of filtering and augmenting. In practice, we gradually increase the scale of our instruction dataset, considering two key insights: 1) ensuring the scale is sufficiently large for LLMs to learn domain-specific and task-related prior knowledge, and 2) avoiding waste in constructing the dataset. We use AdamOptimizer during fine-tuning. The learning rate is set to 5e-5. The batch size is set to 4*4 for Mistral-7B and 1*4 for LLama-13B via gradient accumulation. We train the LLM on the instruction dataset for 3 epochs. To efficiently train the model, we employ LoRA [59] and DeepSpeed training with ZeRO-2 stage [60]. It takes about 3 hours for the training of Mistral-7B and 6 hours for LLaMA-13B. Considering the limited computational speed of RTX8000, it can be much faster to change the GPUs to better devices like V100 or RTX4090. The inference of a single query usually takes less than 10 seconds, which can be further improved via other tools such as VLLM [61]. We spend about 15 dollars on the construction of our instruction dataset with GPT-3.5-turbo API. We do not apply *problem decomposition* and *correction feedback* mechanisms on the NLV dataset since we find that fine-tuned open-source LLMs perform well on it.

4.3 Baseline

Our baselines can be divided into two parts: traditional V-NLI methods including NL4DV, ncNet, and LLM-based methods including GPT-3.5 and GPT-4. NL4DV [62] is a Python package that takes a tabular dataset and a natural language query as input and returns an analytic specification. ncNet [45] is a transformer-based sequence-to-sequence model with several visualization-aware optimizations. GPT-3.5 and GPT-4 are state-of-the-art closed-source LLMs in various tasks. We acknowledge that it is important to incorporate baselines based on open-source LLMs. However, we found that these methods either were not fine-tuned on tasks like table visualization or were not publicly queryable, meaning we are limited to comparing with these methods. As a result, we compare with the public fine-tuned version of base models. Mistral-7B-instruct ① is an instruct fine-tuned version of the Mistral-7B generative text model using a variety of publicly available conversation datasets, denoted as Mistral-C. LLaMA2-13B-chat-hf ② is a fine-tuned version of LLaMA2-13B optimized for dialogue use cases, denoted as LLaMA-C. To compare with Chinese-optimized LLMs, we also choose two famous models, ChatGLM3-6B-instruct and Qwen2.5-14b-instruct, denoted as Chat-GLM-C and Qwen2.5-C. We choose Qwen2.5-coder-14b-instruct as a baseline for code-generation LLMs, denoted as Qwen2.5Coder-C Our models are denoted as LLaMA-V, Mistral-V and Qwen-V.

4.4 Evaluation

Due to the high flexibility of Python programming, it is inappropriate to measure the results using automatic metrics like BLEU and ROUGE. To comprehensively investigate the performance, we carefully design three human evaluation metrics for the generated results, Acc_{val}, Acc_{dat}, and Acc_{fig}. Acc_{val} measures whether the model can produce codes that can be executed successfully to generate charts. Acc_{dat} measures whether the retrieved data is correct with the given query. It should be noted that there could be results with correct codes for data transformation but wrong codes for visualization. As a result, we also ask annotators to check the codes for more accurate Acc_{dat} results. Acc_{fig} measures whether the generated chart meets the demand in the query. We recruited 8 annotators (4 males and 4 females, all of whom possessed experience in Python) to annotate each result on the above metrics. The annotators are fully compensated for their work.

4.5 Results

The results on the NLV dataset are shown in Table 2. The Acc_{val} of NL4DV and ncNet are not reported since they do not generate codes token by token. LLaMA-V, Mistral-V and Qwen-V outperform all the baseline methods and achieve competitive results against closed-source LLMs on both NLV_{ori} and NLV_{gen}. It should be noted that the performance of LLaMA-C and Mistral-C improved greatly after our fine-tuning. We find that our models outperform GPT-3.5 in some cases though most of our training instructions are generated from GPT-3.5. One possible reason is that after filtering out wrong instructions generated and augmenting more correct instructions, the quality of our instruction datasets exceeds the quality that can be generated solely by GPT-3.5. Therefore a small-scale open-source LLM specialized in our task could outperform large-scale closed-source LLMs. Example visualization results of our fine-tuned Mistral-7B model are shown in Figure 5.

① https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

② https://huggingface.co/meta-llama/Llama-2-13b-chat-hf

Table 2 Results on NLV dataset

Datasets	NLV_{ori}			NLV_{gen}		
Models	Acc_{val}	Acc_{dat}	Acc_{fig}	Acc_{val}	Acc_{dat}	Acc_{fig}
NL4DV	-	67.9	61.4	-	56.7	46.7
ncNet	-	75.1	69.0	-	73.3	62.2
LLaMA-C	13.0	10.8	8.6	15.5	8.9	6.7
Mistral-C	47.1	43.9	31.8	24.4	15.6	7.8
ChatGLM-C	57.3	56.5	49.4	45.1	43.3	30.5
Qwen2.5-C	60.1	57.3	51.2	50.4	48.2	39.4
Qwen2.5Coder-C	63.5	60.3	55.8	55.5	50.9	48.8
GPT-3.5	86.7	81.8	72.3	93.3	**85.6**	75.6
GPT-4	92.1	**90.9**	**85.1**	87.8	82.2	**77.8**
LLaMA-V	80.0	76.9	65.2	80.4	76.2	63.3
Mistral-V	94.3	88.6	81.9	94.4	84.4	73.3
Qwen-V	**95.0**	89.0	82.1	**95.0**	85.5	73.2

5 Experiments on Real-world Pre-hospital Care Dataset

5.1 Dataset

Since most public datasets contain less than ten columns and 1,000 rows, they are far away from the complex datasets in reality. We cooperate with Shenzhen Emergency Medical Center in China to evaluate our framework on a real-world dataset. Founded in 1997, Shenzhen Emergency Medical Center has been collecting pre-hospital records since 2009. We have accessed over 1 million historical records of all the emergency dispatches in Shenzhen spanning from 2012 to 2022. Each record encompasses over 100 columns, encompassing attributes pertaining to patients' health conditions, timestamps associated with emergency dispatches, and supplementary information regarding patient outcomes. We remove rows containing any empty data and columns that are not relevant to researchers' interests. After detailed data cleaning, we finally constructed a dataset of historical records with over than 1.2 million rows and 24 columns. We hope to build an information retrieval system with V-NLI on this dataset so that researchers can get insights for assessing the efficacy of current treatments and contributing to the development of evidence-based practice guidelines. The difficulty of this dataset lies not only in its large scale and the complexity of pre-hospital care, but also in its fact that it is not in English, but in Chinese. The performance of lots of models declines when dealing with Chinese, even for GPT-4.

5.2 Experimental Settings

The experimental settings are similar to the NLV dataset. The differences are presented below. We implement our framework on Baichuan-13B [35] due to its strong ability in Chinese. We cooperate with researchers to obtain 100 commonly used queries as seed queries. We first generate 20,000 queries based on seed queries and then obtain about 12,000 query-code pairs. Finally, our instruction dataset contains about 9,000 instructions. We spend about 30 dollars on the construction of the instruction dataset with GPT-3.5-turbo API. The training on the Shenzhen dataset takes about 11.5 hours. The correction feedback mechanism is implemented via one turn re-generation with error message and human feedback. As for the baselines, since NL4DV and ncNet cannot handle Chinese well, we only report the results of ChatGLM-C, Baichuan-C ①, GLM-4-Plus②, GPT-3.5, GPT-4 and our model, denoted as ChatGLM-V and Baichuan-V. The evaluation process remains the same as the NLV dataset.

5.3 Results

The results on Shenzhen pre-hospital care records are shown in table 3. The performances of all the models decline due to the difficulty of the dataset. Our framework still can improve the performance of Baichuan-13B. Our Baichuan-V even outperforms GPT-3.5 and achieves comparable results with other close-sourced LLMs. One of the main reasons for the performance decline in GPT-3.5 comes from misunderstandings about Chinese queries as shown in Figure 6. Results of ablation studies also show that our two mechanisms can enhance the performance of open-source LLMs in this task.

5.4 Case Study

Comparisons of some examples between GPT-3.5, GPT-4, and our model are shown in Figure 6. To enhance display convenience, we have translated the original Chinese queries, coordinate axes, labels, and legends in the generated code

① https://modelscope.cn/models/baichuan-inc/Baichuan-13B-Chat/summary
② https://bigmodel.cn/dev/api/normal-model/glm-4

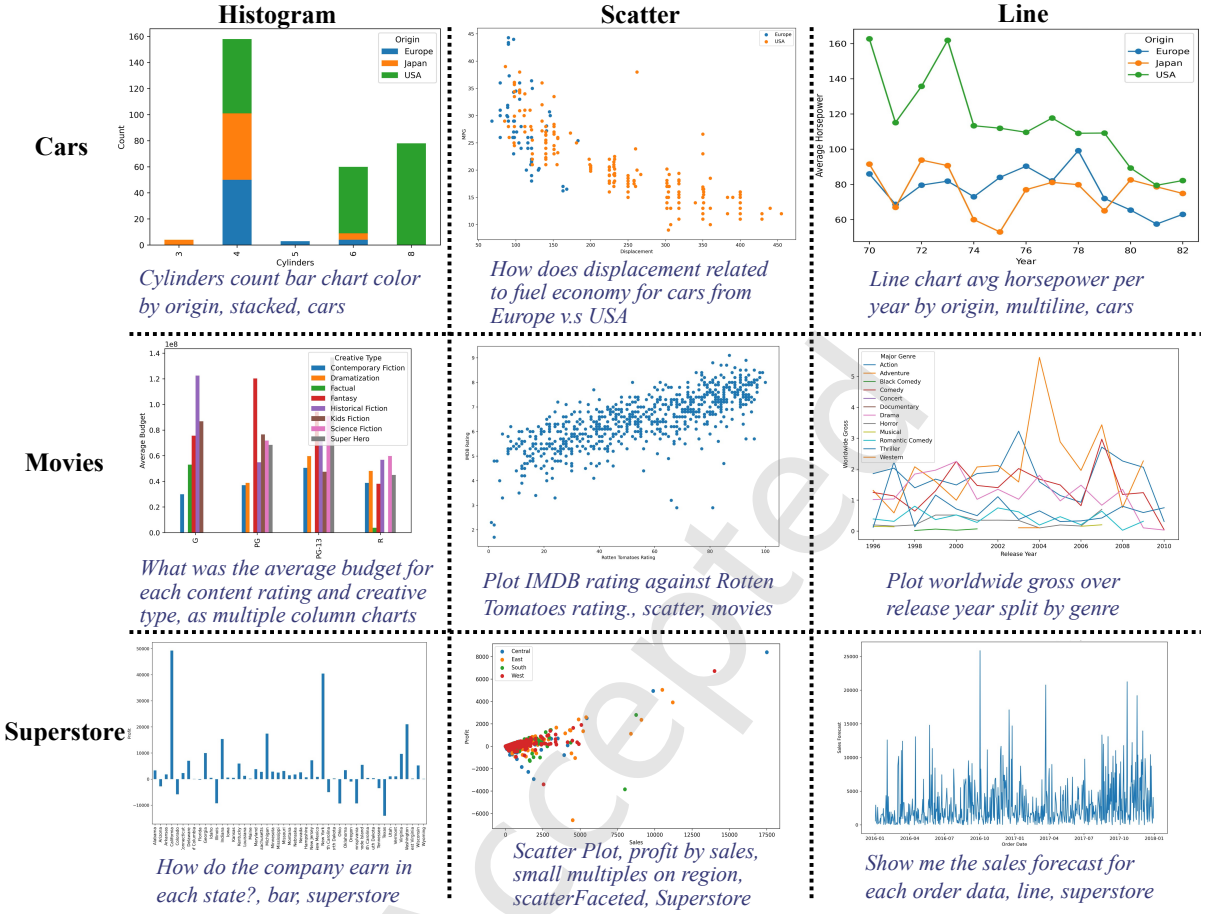

Fig. 5 Example visualization results of our fine-tuned Mistral-7B model on the NLV dataset

Table 3 Results on real-world pre-hospital care dataset

Models	Acc_{val}	Acc_{dat}	Acc_{fig}
ChatGLM-C	40.4	32.1	20.5
Baichuan-C	48.5	34.7	21.8
GLM-4-Plus	95.2	**86.1**	71.8
GPT-3.5	89.1	76.2	64.3
GPT-4	<u>94.1</u>	<u>85.1</u>	**74.3**
ChatGLM-V	92.0	75.1	63.5
Baichuan-V	**96.0**	79.2	<u>72.3</u>
w/o problem decomposition	88.1	73.3	63.4
w/o correction feedback	92.1	71.3	62.4

into English. In the first query, all the three models generate the correct charts. However, in the second and third queries, GPT-3.5 generated the wrong charts and reversed the attributes of the x-axis (24-hour in the second query and address types in the third query) from the corresponding attributes of the color (age group in the second query and gender in the third query). GPT-4 and our model both output the correct charts. The difference between the two in the third query is that our model further reflects the proportion of patients with different address types through the height of different bars in the charts. It also should be noted that we observe that the charts generated by GPT-4 are usually more nice-looking than our model, which can be treated as an important direction of our future work.

Figure 7 also gives an example showing how researchers can leverage our model to perform data analysis on V-NLI-based information retrieval. Researchers first observe the 24-hour distribution of all kinds of disease types for some insights. Then they may be attracted by the distribution differences between the two disease types and presented them separately for further observation. After observation, researchers may find out that patients with ”physical and chemical poisoning” more frequently appear at midnight while patients involved in ”traffic accidents” more frequently appear during the day. Then they can take a further look at the address types of these disease types via the heatmaps below. Patients with ”physical and chemical poisoning” more frequently appear in locations such as hotels and residences, indicating that the possible cause of these patients is excessive alcohol consumption. With our model, researchers can explore the records via the interactions in natural language and ob-

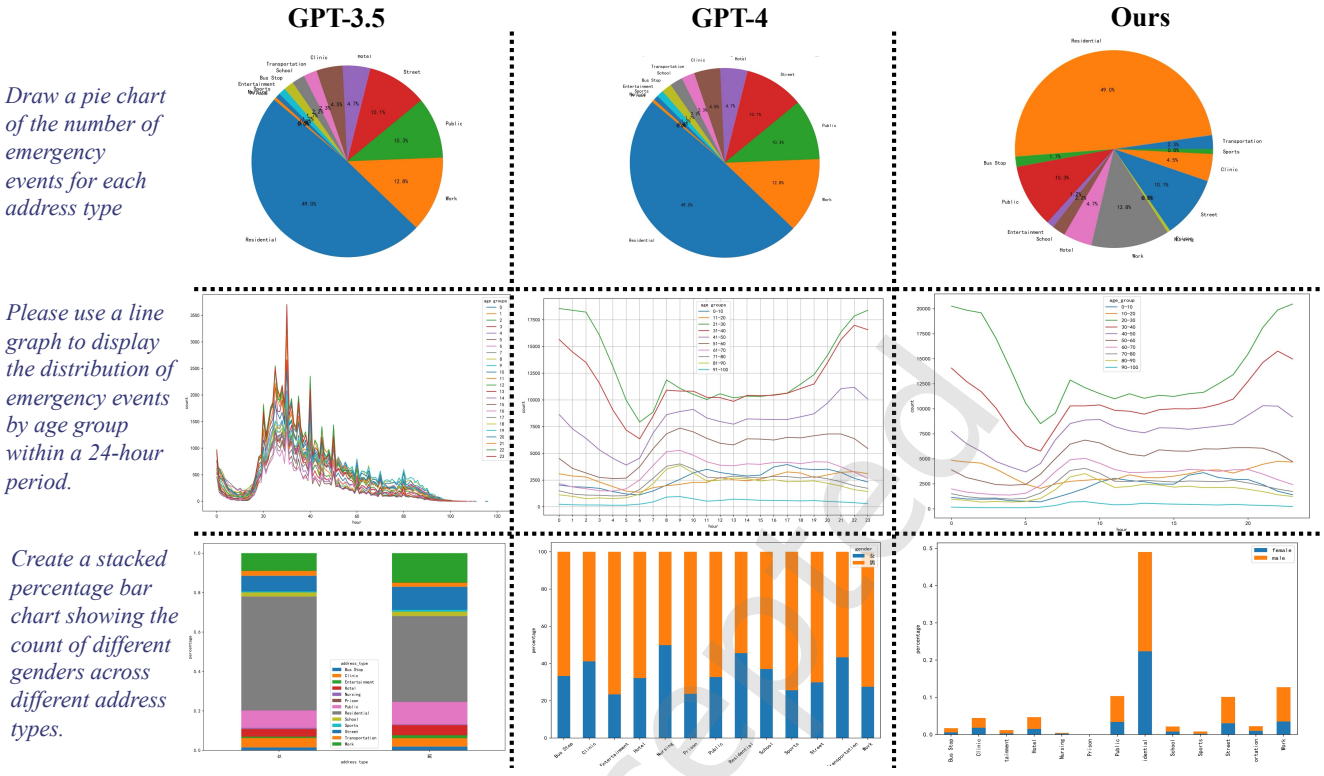

Fig. 6 Comparison of some examples between GPT-3.5, GPT-4 and our model

tain important inspiration about the correlations between interested independent variables and indicators related to pre-hospital emergency care, which can further help optimize resource allocation and improve emergency efficiency.

6 Limitation and Future Work

Experiments on more settings. Though we have tried our best in experiments with an aim to demonstrate the usefulness of our framework, experiments on more datasets can make the results more convincing. We only fine-tuned open-source LLMs to generate Python codes. Using other programming languages such as SQL for data retrieval and VegaLite for visualization can further improve our generalizability and performance. Supported chart categories can also be increased with other programming languages.

Generalizability to Other Domains or Datasets: A key challenge in applying the framework to other fields lies in acquiring high-quality instruction-tuning data. In our current approach, we generate such data using more powerful closed-source models, followed by a filtering process. This strategy has proven effective in our specific context, but we recognize that it may not be directly applicable in all domains. To extend the framework to other fields, one promising alternative is leveraging existing traditional data visualization systems within those domains. These systems already generate accurate data retrieval and visualization code based on user interactions via predefined buttons. Although the diversity of queries generated by these systems is constrained by their predefined nature, they can still provide valuable instruction data. By incorporating these systems, we believe the framework's adaptability and generalizability could be significantly improved. Due to the focus of this work and the constraints of space, we did not explore this approach in depth.

Effective adaptation to schema change. In reality, the schema of a database will change during usage, which raises challenges for not only our framework but even humans. In our framework, LLMs learn the mapping relationship between natural language and header attributes via supervised fine-tuning of instructions, remaining difficulties for LLMs to understand new attributes. One potential solution to this issue, especially when schema changes are relatively minimal, is to use prompt-based methods to describe the new schema and provide demonstrations that help the model understand the new fields. This approach could allow the model to handle minor changes without the need for full retraining. However, as the number of new schema changes increases over time, retraining the model with updated data and schema structures would become necessary. This would allow for periodic version updates to ensure that the system remains accu-

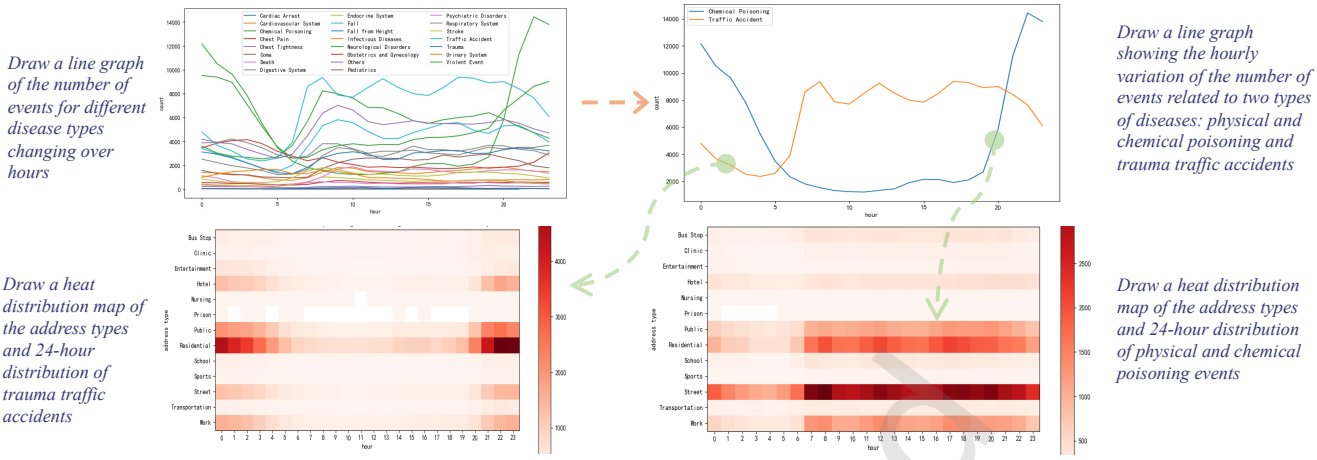

Fig. 7 Case studies on the exploration of pre-hospital care records based on our model.

rate and adaptable. Another is to incorporate ontology models or knowledge graphs during training as domain-specific prior knowledge so that LLMs can understand all possible attributes in this domain.

Multi-turn conversational information retrieval. Our instructions are generated in single-turn conversation with LLMs. However, multi-turn conversations enable users to make modifications to their queries and help LLMs to correct and refine their generated codes. In fact, the case study shown in Figure 7 can be considered a demonstration of single-turn interaction, but it also illustrates how the system can handle follow-up queries within the same interaction. This approach is conceptually similar to multi-turn dialogue in terms of maintaining context across successive prompts. A possible solution is to add general multi-turn conversations into instruction datasets and generate multi-turn instructions for our task, which requires the involvement of experienced experts during the construction of instruction datasets.

7 Conclusion

In facilitating the exploration of historical emergency records, we have developed a framework to build a V-NLI-based information retrieval system for pre-hospital emergency care, an area of considerable importance that has been relatively underexplored. Our approach involves the utilization of open-source LLMs, and we have established a comprehensive framework that seamlessly integrates domain-specific and task-related knowledge via supervised fine-tuning on an instruction dataset generated by closed-source LLMs. Furthermore, our framework's efficacy and practicality shine through experiments on a public dataset and a real-world dataset encompassing over 1 million pre-hospital emergency historical records in Shenzhen spanning from 2012-2022.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 82241052.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

[1] C. Ahl and M. Nyström, To handle the unexpected–the meaning of caring in pre-hospital emergency care, *International emergency nursing*, vol. 20, no. 1, pp. 33–41, 2012.

[2] M. H. Wilson, K. Habig, C. Wright, A. Hughes, G. Davies, and C. H. Imray, Pre-hospital emergency medicine, *The Lancet*, vol. 386, no. 10012, pp. 2526–2534, 2015.

[3] C. W. Seymour, T. D. Rea, J. M. Kahn, A. J. Walkey, D. M. Yealy, and D. C. Angus, Severe sepsis in pre-hospital emergency care: analysis of incidence, care, and outcome, *American journal of respiratory and critical care medicine*, vol. 186, no. 12, pp. 1264–1271, 2012.

[4] W. Huang, T.-B. Wang, Y.-D. He, H. Zhang, X.-H. Zhou, H. Liu, J.-J. Zhang, Z.-B. Tian, and B.-G. Jiang, Trends and characteristics in pre-hospital emergency care in beijing from 2008 to 2017, *Chinese Medical Journal*, vol. 133, no. 11, pp. 1268–1275, 2020.

[5] S. Ibrihich, A. Oussous, O. Ibrihich, and M. Esghir, A review on recent research in information retrieval, *Procedia Computer Science*, vol. 201, pp. 777–782, 2022.

[6] A. Singhal *et al.*, Modern information retrieval: A brief overview, *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 35–43, 2001.

[7] B. Lee, P. Isenberg, N. H. Riche, and S. Carpendale, Beyond mouse and keyboard: Expanding design considerations for information visualization interactions, *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2689–2698, 2012.

[8] E. Hoque, V. Setlur, M. Tory, and I. Dykeman, Applying pragmatics principles for interaction with visual analytics, *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 309–318, 2017.

[9] L. Battle and C. Scheidegger, A structured review of data management technology for interactive visualization and analysis, *IEEE transactions on visualization and computer graphics*, vol. 27, no. 2, pp. 1128–1138, 2020.

[10] K. Cox, R. E. Grinter, S. L. Hibino, L. J. Jagadeesan, and D. Mantilla, A multi-modal natural language interface to an information visualization environment, *International Journal of Speech Technology*, vol. 4, pp. 297–314, 2001.

[11] L. Shen, E. Shen, Y. Luo, X. Yang, X. Hu, X. Zhang, Z. Tai, and J. Wang, Towards natural language interfaces for data visualization: A survey, *IEEE transactions on visualization and computer graphics*, 2022.

[12] S. K. Card, J. Mackinlay, and B. Shneiderman, *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[13] Y. Luo, N. Tang, G. Li, C. Chai, W. Li, and X. Qin, Synthesizing natural language to visualization (nl2vis) benchmarks from nl2sql benchmarks, in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 1235–1247.

[14] K. Affolter, K. Stockinger, and A. Bernstein, A comparative survey of recent natural language interfaces for databases, *The VLDB Journal*, vol. 28, pp. 793–819, 2019.

[15] F. Őzcan, A. Quamar, J. Sen, C. Lei, and V. Efthymiou, State of the art and open challenges in natural language interfaces to data, in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2020, pp. 2629–2636.

[16] K. Chowdhary and K. Chowdhary, Natural language processing, *Fundamentals of artificial intelligence*, pp. 603–649, 2020.

[17] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, Natural language processing: an introduction, *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 544–551, 2011.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805*, 2018.

[19] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, Unified language model pre-training for natural language understanding and generation, *Advances in neural information processing systems*, vol. 32, 2019.

[20] H. Gong, Y. Sun, X. Feng, B. Qin, W. Bi, X. Liu, and T. Liu, Tablegpt: Few-shot table-to-text generation with table structure reconstruction and content matching, in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 1978–1988.

[21] P. Li, Y. He, D. Yashar, W. Cui, S. Ge, H. Zhang, D. R. Fainman, D. Zhang, and S. Chaudhuri, Table-gpt: Table-tuned gpt for diverse table tasks, *arXiv preprint arXiv:2310.09263*, 2023.

[22] L. Zha, J. Zhou, L. Li, R. Wang, Q. Huang, S. Yang, J. Yuan, C. Su, X. Li, A. Su *et al.*, Tablegpt: Towards unifying tables, nature language and commands into one gpt, *arXiv preprint arXiv:2307.08674*, 2023.

[23] H. Li, J. Su, Y. Chen, Q. Li, and Z. Zhang, Sheetcopilot: Bringing software productivity to the next level through large language models, *arXiv preprint arXiv:2305.19308*, 2023.

[24] W. Zhang, Y. Shen, W. Lu, and Y. Zhuang, Data-copilot: Bridging billions of data and humans with autonomous workflow, *arXiv preprint arXiv:2306.07209*, 2023.

[25] T. Zhang, X. Yue, Y. Li, and H. Sun, Tablellama: Towards open large generalist models for tables, *arXiv preprint arXiv:2311.09206*, 2023.

[26] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, Language models are few-shot learners, *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[27] R. OpenAI, Gpt-4 technical report. arxiv 2303.08774, *View in Article*, vol. 2, p. 13, 2023.

[28] J. R. Díaz-Palacios, V. J. Romo-Aledo, and A. H. Chinaei, Biometric access control for e-health records in pre-hospital care, in *Proceedings of the joint EDBT/ICDT 2013 workshops*, 2013, pp. 169–173.

[29] Y. Tian, W. Cui, D. Deng, X. Yi, Y. Yang, H. Zhang, and Y. Wu, Chartgpt: Leveraging llms to generate charts from abstract natural language, *arXiv preprint arXiv:2311.01920*, 2023.

[30] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu *et al.*, Instruction tuning for large language models: A survey, *arXiv preprint arXiv:2308.10792*, 2023.

[31] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, Training language models to follow instructions with human feedback, *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[32] A. Srinivasan, N. Nyapathy, B. Lee, S. M. Drucker, and J. Stasko, Collecting and characterizing natural language utterances for specifying data visualizations, in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–10.

[33] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, Llama 2: Open foundation and fine-tuned chat models, *arXiv preprint arXiv:2307.09288*, 2023.

[34] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier *et al.*, Mistral 7b, *arXiv preprint arXiv:2310.06825*, 2023.

[35] A. Yang, B. Xiao, B. Wang, B. Zhang, C. Bian, C. Yin, C. Lv,

D. Pan, D. Wang, D. Yan *et al.*, Baichuan 2: Open large-scale language models, *arXiv preprint arXiv:2309.10305*, 2023.

[36] J. Cimino and C. Braun, Clinical research in prehospital care: Current and future challenges, *Clinics and Practice*, vol. 13, no. 5, pp. 1266–1285, 2023.

[37] Z. Finn, P. Carter, D. Rogers, and A. Burnett, Prehospital covid-19-related encounters predict future hospital utilization, *Prehospital Emergency Care*, vol. 27, no. 3, pp. 297–302, 2023.

[38] T. Li, D. Koloden, J. Berkowitz, D. Luo, H. Luan, C. Gilley, G. Kurgansky, and P. Barbara, Prehospital transport and termination of resuscitation of cardiac arrest patients: A review of prehospital care protocols in the united states, *Resuscitation Plus*, vol. 14, p. 100397, 2023.

[39] R. D. Oanesa, T. W.-H. Su, and A. Weissman, Evidence for use of validated sepsis screening tools in the prehospital population: A scoping review, *Prehospital Emergency Care*, no. just-accepted, pp. 1–21, 2023.

[40] T. Mikolov, K. Chen, G. Corrado, and J. Dean, Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781*, 2013.

[41] W.-D. J. Zhu, B. Foyle, D. Gagné, V. Gupta, J. Magdalen, A. S. Mundi, T. Nasukawa, M. Paulis, J. Singer, M. Triska *et al.*, *IBM Watson content analytics: discovering actionable insight from your content*. IBM Redbooks, 2014.

[42] L. T. Becker and E. M. Gould, Microsoft power bi: Extending excel to manipulate, analyze, and visualize diverse data, *Serials Review*, vol. 45, no. 3, pp. 184–188, 2019.

[43] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios, Datatone: Managing ambiguity in natural language interfaces for data visualization, in *Proceedings of the 28th annual acm symposium on user interface software & technology*, 2015, pp. 489–500.

[44] C. Liu, Y. Han, R. Jiang, and X. Yuan, Advisor: Automatic visualization answer for natural-language question on tabular data, in *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*. IEEE, 2021, pp. 11–20.

[45] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin, Natural language to visualization by neural machine translation, *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 217–226, 2021.

[46] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman *et al.*, Evaluating large language models trained on code, *arXiv preprint arXiv:2107.03374*, 2021.

[47] J. J. Y. Chung, W. Kim, K. M. Yoo, H. Lee, E. Adar, and M. Chang, Talebrush: Sketching stories with generative pretrained language models, in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–19.

[48] T. S. Kim, D. Choi, Y. Choi, and J. Kim, Stylette: Styling the web with natural language, in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–17.

[49] P. Maddigan and T. Susnjak, Chat2vis: Generating data visualisations via natural language using chatgpt, codex and gpt-3 large language models, *IEEE Access*, 2023.

[50] L. Cheng, X. Li, and L. Bing, Is gpt-4 a good data analyst? *arXiv preprint arXiv:2305.15038*, 2023.

[51] G. Dong, H. Yuan, K. Lu, C. Li, M. Xue, D. Liu, W. Wang, Z. Yuan, C. Zhou, and J. Zhou, How abilities in large language models are affected by supervised fine-tuning data composition, in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 177–198.

[52] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, Improving language understanding by generative pre-training, OpenAI, Tech. Rep., 2018.

[53] L. Li, J. Zhou, Z. Gao, W. Hua, L. Fan, H. Yu, L. Hagen, Y. Zhang, T. L. Assimes, L. Hemphill *et al.*, A scoping review of using large language models (llms) to investigate electronic health records (ehrs), *arXiv preprint arXiv:2405.03066*, 2024.

[54] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, Chain-of-thought prompting elicits reasoning in large language models, *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[55] N. Lee, K. Sreenivasan, J. D. Lee, K. Lee, and D. Papailiopoulos, Teaching arithmetic to small transformers, in *The Twelfth International Conference on Learning Representations*, 2024.

[56] Y. Kim, C. Park, H. Jeong, Y. S. Chan, X. Xu, D. McDuff, H. Lee, M. Ghassemi, C. Breazeal, and H. W. Park, Mdagents: An adaptive collaboration of llms for medical decision-making, *Advances in Neural Information Processing Systems*, vol. 37, pp. 79 410–79 452, 2024.

[57] J. J. Jia, Z. Yuan, J. Pan, P. McNamara, and D. Chen, Decision-making behavior evaluation framework for llms under uncertain context, *Advances in Neural Information Processing Systems*, vol. 37, pp. 113 360–113 382, 2024.

[58] Y. Luo, J. Tang, and G. Li, nvbench: A large-scale synthesized dataset for cross-domain natural language to visualization task, *arXiv preprint arXiv:2112.12926*, 2021.

[59] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, Lora: Low-rank adaptation of large language models, *arXiv preprint arXiv:2106.09685*, 2021.

[60] S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He, Zero: Memory optimizations toward training trillion parameter models, in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2020, pp. 1–16.

[61] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, and I. Stoica, Efficient memory management for large language model serving with pagedattention, in *Proceedings of the 29th Symposium on Operating Systems Principles*, 2023, pp. 611–626.

[62] A. Narechania, A. Srinivasan, and J. Stasko, Nl4dv: A toolkit

for generating analytic specifications for data visualization from natural language queries, *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 369–379, 2020.

Author biography

Xin Gao received his Ph.D. degree in Computer Software and Theory from Peking University in 2024 and is currently a postdoctoral researcher at the Department of Computer Science, Peking University, and a member of the CENTRAi Lab. His research interests include text classification, topic modeling, deep representation learning, large language models, and artificial intelligence for healthcare. His recent work focuses on retrieval-augmented generation, pre-hospital emergency data analysis, and large language models for medical information extraction. He has published more than ten papers in leading conferences and journals, including ICDE, ICDM, ICML, ACL, AAAI, and ICLR.

Yasha Wang received his Ph.D. degree from Northeastern University, China, in 2003. He is a Boya Distinguished Professor at Peking University, Director of the National Engineering Research Center for Software Engineering, Principal Investigator of the CENTRAi Lab, and a Changjiang Scholar of China's Ministry of Education. His research interests include artificial intelligence, data mining, big data analytics, and interdisciplinary research in artificial intelligence and medicine. He has published more than 100 papers in leading venues, including ICML, KDD, ICDE, AAAI, IJCAI, and IEEE TMC. His work received the Second Prize of the National Science and Technology Progress Award of China and the First Prize of the Science and Technology Progress Award of the Ministry of Education.

William Cheng-Chung Chu is a Professor and Ph.D. supervisor at the School of Computing and Artificial Intelligence, Fuyao University of Science and Technology, China. He received his M.S. and Ph.D. degrees in Computer Science from Northwestern University, USA, in 1987 and 1989, respectively. Before entering academia, he worked as a research scientist at the Software Technology Center of Lockheed Missiles and Space Company and was a visiting scholar at Stanford University. His research interests include artificial intelligence, large language models, intelligent healthcare, big data analytics, and software engineering. He has published more than 300 research papers and has extensive experience in interdisciplinary research involving software engineering, artificial intelligence, and digital healthcare.