IMQC: A Large Language Model Platform for Medical Quality Control

Qi Ye[1], Guangya Yu[1], Jingping Liu[1†], Erzhen Chen[2], Chenjie Dong[2], Xiaosheng Lin[3†], Zelei Liu[4†], Han Yu[5†], Tong Ruan[1]

[1] School of Information Science and Technology, East China University of Science and Technology, Shanghai, China
[2] Shanghai Medical Quality Control Management Center, Shanghai, China
[3] Xinhong Community Health Service Center, Minhang District, Shanghai, China
[4] Unicom (Shanghai) Industrial Internet Co.,Ltd., Shanghai, China
[5] College of Computing and Data Science, Nanyang Technological University, Singapore
[†]jingpingliu@ecust.edu.cn, xiao_sheng2001@sina.com, liuzl231@chinaunicom.cn, han.yu@ntu.edu.sg

Abstract

Medical quality control (MQC) indicators are essential for evaluating the performance of healthcare institutions to ensure high-quality patient care. In this paper, we report the design, implementation, and deployment of the Intelligent EMR-LLM platform for Medical Quality Control (IMQC), a large language model (LLM)-empowered system for automatically computing MQC indicators for enhancing the quality of medical services in Shanghai. It consists of an LLM (i.e., EMR-LLM) developed using electronic medical records (EMRs). With EMR-LLM, IMQC translates existing MQC indicators into a standardized representation language and automatically computes them based on EMRs. Since its deployment in February 2024, IMQC has been adopted by the Shanghai Medical Quality Management Center and associated hospitals. So far, it has processed 1,245 medical quality indicators for secondary- and tertiary-level hospitals, achieving an MQC evaluation accuracy of 93.31%, which is comparable to human experts. It has significantly improved efficiency, increasing from 10 EMRs per hour per human expert to over 1,000 EMRs per hour on average using one single H800 GPU. Over the first round of deployment in Shanghai, it is estimated that IMQC saves around 3.42 million RMB per month in manpower costs compared to traditional reporting methods. The successful deployment of IMQC sets a precedence for other regions to adopt similar AI-driven solutions to enhance medical quality control.

Introduction

Medical quality control (MQC) refers to the systematic procedures ensuring that healthcare service provision meets established standards of quality and safety based on analyzing EMRs (Williams, Parry, and Schlup 1992). It is crucial for maintaining high standards of patient care, enhancing service efficiency, and ensuring patient satisfaction and safety.

Quality control indicators serve as tangible expressions of MQC, offering measurable benchmarks for assessing the quality of healthcare service provision (Øvretveit 2001). For instance, the adenoma detection rate and sessile serrated lesion detection rate are critical quality indicators for endoscopists, as they are strongly associated with the risk of post-colonoscopy colorectal cancer and related mortality (Anderson et al. 2017). By examining the insights gained from these indicators, healthcare institutions can identify opportunities for improvement to refine their treatment procedures, ultimately leading to better patient outcomes and improved efficiency (Wang et al. 2018).

Currently, the calculation of quality control indicators primarily relies on heuristic rules defined by experts (Tamang et al. 2015; Hsu et al. 2016). However, this approach is often inefficient (Ross et al. 2015) and lacks generalizability (Raju et al. 2015), significantly limiting its practical application, especially when managing a large number of quality control indicators. In this approach, engineers first design computer scripts to extract specific field contents from EMRs related to the indicator (Raju et al. 2013). For structured content, they manually create heuristic rules to calculate the quality control indicators. For unstructured content, they manually design extraction rules and summarize the extracted texts (Bae et al. 2022). The process of defining these rules requires significant expertise. Each time a new quality control indicator is introduced, the rules must be redefined as automatic adaptation to different indicators is challenging. Therefore, there is an urgent need for a more automated and accurate method for calculating quality control indicators to enhance the effectiveness of medical quality control for standardization of medical procedures.

In this paper, we report the design, implementation and deployment of the Intelligent EMR-LLM platform for Medical Quality Control (IMQC), an intelligent medical quality control platform for automating the calculation of quality control indicators based on a large language model (LLM), to address the aforementioned challenges. Firstly, we collected a substantial amount of open-domain and medical-domain corpora to continually pre-train an open-source LLM, thereby equipping it with knowledge related to the medical field. We refer to this pre-trained model as Medical-LLM. Next, we created instruction-tuning datasets based on EMRs to enable the Medical-LLM to understand and calculate quality control indicators. To further enhance the Medical-LLM's understanding of EMR data, we developed four instruction-tuning tasks: 1) EMR integrity detec-

[†]Corresponding authors

tion, 2) content standardization detection, 3) logical consistency checking, and 4) terminology standardization. Based on these data, we proposed an instruction-tuning method based on curriculum learning for the Medical-LLM, which aligns the learning process of the LLM with human cognitive development–beginning with simpler tasks and progressively tackling more complex ones. After instruction tuning, we refer to the resulting LLM as EMR-LLM. Finally, we established a standardized representation language for quality control indicators and leveraged EMR-LLM to facilitate the automatic conversion into this standardized language. This framework can make factual judgments on EMRs based on the EMR-LLM, and integrate logical rules to achieve automatic calculation of medical quality control indicators.

Our IMQC platform offers three key advantages:

1. **Efficiency**: With the EMR-LLM for automated calculation of quality control indicators, it significantly reduces the need for manual judgment. This enables a small number of experts to review a large amount of results, greatly minimizing costs related to human resources.
2. **Transparency**: The platform retains comprehensive records of the calculation process for each EMR, including the retrieval of EMR field contents, factual judgment by the EMR-LLM, and the integration of facts with logical rules. This transparency enables users to easily understand the rationale behind each step.
3. **Generalization**: We design a standardized language conversion module that is versatile and can adapt to the indicators for different diseases. During the calculation process, we use an LLM (e.g., EMR-LLM) to perform automatic calculations. This general approach makes our method more adaptable to real-world scenarios.

The platform has been adopted by the Medical Quality Control Management Center and hospitals in Shanghai. Previously, clinicians manually calculate quality control indicators with heuristic rules, which are then reviewed by experts from the Medical Quality Control Management Center. This process has now been replaced by the IMQC platform to perform the calculations, while experts continue to conduct the reviews. Since its launch in February 2024, the platform has calculated 1,245 quality control indicators for secondary- and tertiary-level hospitals across Shanghai, with a high accuracy of 93.31%. Using a single H800 GPU card, the system can process 1,000 medical records per hour, far surpassing the manual review rate of 10 medical records per hour per clinician, resulting in a 100-fold increase in efficiency. Over the first round of deployment in the hospitals involved, it is estimated that IMQC saves around 3.42 million RMB per month in manpower costs compared to the traditional approach.

Application Description

As illustrated in Figure 1, the system architecture comprises four modules:

1. **Data Management**: This module is primarily responsible for data collection and cleaning. Through this module, we collected a substantial amount of pre-training corpora, including open-domain data and medical domain data (e.g., medical books, clinical guidelines, expert consensus, and quality control documents). Then, the data went through a data cleaning tool for OCR text recognition, noise processing, and duplication elimination.
2. **EMR-LLM Construction**: The core of this module is the LLM developed using EMRs. Firstly, we carried out continual pre-training on an open-source foundation LLM using collected corpora to obtain the Medical-LLM. Next, we created several instruction-tuning datasets derived from EMRs. Finally, we fine-tuned the Medical-LLM using both the constructed instruction data and open-source instruction data to build the EMR-LLM, and evaluated it with public and proprietary benchmarks to ensure its effectiveness on medical quality control tasks.
3. **Quality Control Indicator Calculation**: This module primarily leverages the EMR-LLM to determine whether each EMR meets the quality control indicator requirements. Firstly, we defined a standardized representation language for quality control indicators and implemented indicator representation conversion using the EMR-LLM. Then, EMR-LLM is used to automatically calculate quality control indicators for EMRs based on this standardized language. Additionally, an annotation tool was developed for experts to review the standardized language, ensuring accurate results.
4. **User Interaction**: This module provides an interactive interface for hospitals and the Medical Quality Control Management Center. The interface includes three main functions: 1) real-time indicator calculation, 2) indicator result search, and 3) data visualization.

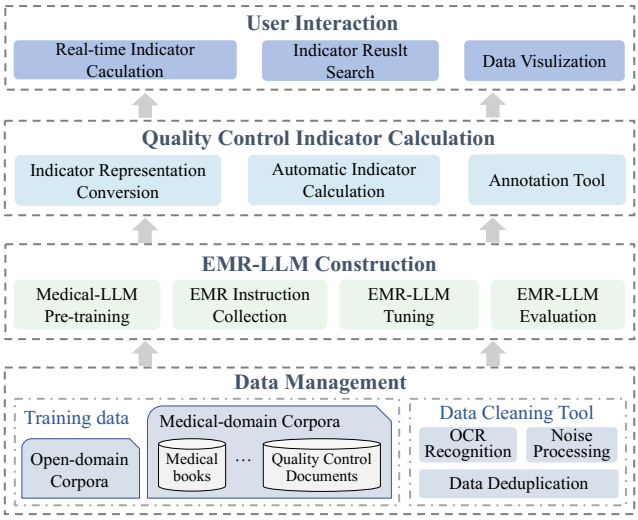

Figure 1: The system architecture of the IMQC platform.

Use of AI Technology

The IMQC platform leverages AI techniques in two primary aspects: 1) creating an LLM using EMRs, and 2) calculating quality control indicators. This section offers a detailed discussion of both aspects.

Phase	Data type	Descriptions	Size
Pre-training	Medical books	Covering disease diagnosis, treatment, medication, etc, with a total of 2,730 books.	0.27G
	Clinical guidelines	Collated medical consensus in healthcare, with a total of 5,155 books.	0.98G
	Expert consensus	Consensus of experts on the state-of-the-science based on evidence, with 2,710 books.	0.17G
	Indicator compilations	Definitions, explanations, and calculation methods for quality control indicators.	0.01G
	Basic medical corpora	CPT, Mimic, Pubmed, etc.	15.17G
	General corpora	Wikipedia, falcon-refinedweb, WuDao, Clue, etc.	66.40G
Instruction-tuning	QCIC	Determine if the contents of EMRs meet the quality control indicators.	2,411
	EMR_ID	Check if any field in EMRs is missing.	1,486
	CSD	Verify if the field content in EMRs complies with official writing standards.	1,753
	LCC	Identify any inconsistency between the contents of two fields.	1,547
	TS	Standardize irregular medical phrases into proper medical terms.	6,641
	Basic medical data	Cblue, MedDialog, BioGPT, Huatuo, etc.	134K
	General data	Alpaca, Alpaca_gpt4_zh, Belle, Firefly, etc.	533K

Table 1: Statistics of pre-training and instruction-tuning data. CPT: https://github.com/williamliujl/Qilin-Med; Mimic: https://mimic.mit.edu/; Pubmed: https://huggingface.co/datasets/ncbi/pubmed; Wikipedia: https://huggingface.co/datasets/wikimedia/wikipedia; falcon-refinedweb:https://huggingface.co/datasets/tiiuae/falcon-refinedweb; WuDao: https://data.baai.ac.cn/details/WuDaoCorporaText; Clue: https://github.com/CLUEbenchmark/CLUECorpus2020; Cblue: https://github.com/CBLUEbenchmark/CBLUE; MedDialog: https://huggingface.co/datasets/bigbio/meddialog; BioGPT: https://github.com/microsoft/BioGPT; Alpaca: https://huggingface.co/datasets/vicgalle/alpaca-gpt4; Alpaca-gpt4: https://huggingface.co/datasets/llamafactory/alpaca_gpt4_zh; Belle: https://huggingface.co/BelleGroup; Firefly: https://huggingface.co/datasets/YeungNLP/firefly-train-1.1M.

EMR-LLM Construction

The platform is designed to automatically assess hospital diagnostic and treatment processes based on MQC indicators. At its core is a powerful LLM for electronic medical records, known as EMR-LLM, which accurately understands and processes medical data, thereby enhancing assessment performance. The development of EMR-LLM involves three key steps: 1) pre-training a general model with medical data to create Medical-LLM, 2) developing the EMR-Instruction dataset using EMR data, and 3) fine-tuning Medical-LLM with this dataset to produce EMR-LLM.

Pre-Training of Medical-LLM During the continual pre-training phase, we first built large medical corpora (including medical books, clinical guidelines, expert consensus, and compilations of medical quality control indicators as shown in Table 1). To ensure the quality of pre-training data, we developed an automatic data cleaning tool that performs OCR text recognition, noise processing, and data de-duplication. After data processing, we employed Seq-Model (Zhang et al. 2021) to semantically segment the texts and generate logically coherent paragraphs, thereby enhancing the information density and contextual semantic relevance of the data. Finally, we leveraged an unsupervised auto-regressive method (Radford et al. 2018) to continue pre-train a general LLM, which predicts the next word based on one or more previous words. To mitigate catastrophic forgetting, we introduced open-domain corpora during the continual pre-training phase. The training objective of the model is to minimize the following loss function:

$$\mathcal{L} = -\sum_{t=1}^{T} \log P(X_t | X_{t-1}, X_{t-2}, ..., X_1), \qquad (1)$$

where X_t represents the t-th token in the text and T is the total number of tokens. We refer to this pre-trained LLM as Medical-LLM.

EMR-Instruction Collection The core function of the platform is to calculate the predefined quality control indicators for each EMR, and use these results to evaluate the level of service provision of a given hospital. To achieve this, we ask clinicians to manually write instruction samples for this task, where the input is the EMR and quality control indicator, and the output is the calculation process and conclusion (whether it meets the requirements, or cannot be judged). A total of 800 samples were written for this task. Based on these data, we leverage an LLM to rewrite each input and output multiple times, and then ask the same group of clinicians to review the rewritten sample pairs. After manual review, we obtain 2,411 samples. Finally, we manually write several instructions for this task, ask the LLM to rewrite these task instructions, and combine them with the input and output pairs to form an instruction dataset.

To enhance the understanding of EMR data by the LLM, we further design four additional instruction tasks: 1) EMR integrity detection (EMR_ID), 2) content standardization detection (CSD), 3) logical consistency checking (LCC), and 4) terminology standardization (TS). For EMR_ID, the inputs are EMR data, and the output indicates whether there are any missing fields. If fields are missing, their names need to be specified. For CSD, the input is the content of an EMR field, and the output determines whether it meets the requirements for medical record writing. If it does not, the reason must be provided. For LCC, the input consists of the content of two related fields in the EMR, and the output assesses whether there are contradictions between the two fields. If contradictions exist, they are identified. The data construction for the above tasks follows the same process as quality

control indicator calculation (QCIC): 1) expert annotation, 2) large model expansion, and 3) expert review. After reviewing, the number of instruction data for EMR_ID, CSD, and LCC are 1,486, 1,753, and 1,547, respectively. For TS, the inputs are non-standard clinical phrases, and the outputs are standardized clinical terms. This task aims to help the model understand the terminology used in actual clinical processes, using data primarily from CHIP-CDN[1] and Yidu-N7K.[2] To enhance the generalization of the Medical-LLM in different medical tasks and avoid catastrophic forgetting, we also collected a large number of instruction data in the open domain and the medical domain. For convenience, we refer to all the above instruction datasets as EMR-Instruction, and their statistics are presented in Table 1.

Instruction Tuning of EMR-LLM In this section, we propose an instruction tuning method based on curriculum learning (Bengio et al. 2009; Liu et al. 2021). This approach mirrors the human cognitive process, starting with simpler tasks and gradually moving to more complex ones. Specifically, we use the LLM without any training to perform EMR-related tasks, including QCIC, EMR_ID, CSD, LCC, and TS. We rank these tasks from easy to difficult based on the model performance on each task, as defined below:

$$Seq(M) = Ord[LLM(D_i) \mid i = Q, E, C, L, T], \quad (2)$$

where M is the set of the above five tasks. Q, E, C, L and T are the abbreviations of these tasks, respectively. D_i refers to the dataset of a specific task. $LLM(\cdot)$ denotes the score of the LLM on a particular task. $Ord(\cdot)$ represents the ranking of these scores from low to high. The model tuning process is as follows. Initially, we fine-tune Medical-LLM with general domain instruction data and basic medical instruction data to equip it with fundamental medical dialogue capabilities. In the subsequent epochs, we fine-tune the model on EMR tasks in order of increasing difficulty, ensuring that the model gradually masters clinical medical knowledge. Finally, we use all the instruction data to fine-tune the model to prevent it from forgetting previously learned knowledge. After completing fine-tuning using LoRA (Hu et al. 2021), we obtain EMR-LLM.

Quality Control Indicator Calculation

EMR-LLM is leveraged to infer the quality control indicators for EMRs. A straightforward approach is to use in-context learning (ICL) or chain-of-thought (CoT) prompt technology, allowing the LLM to determine whether the EMR meets the requirements of the quality control indicator. However, this method is often not effective enough because the judgment methods for different indicators might be inconsistent, and some indicators involve complex reasoning processes. For example, the "blood transfusion rate" indicator simply identifies whether the patient has received a blood transfusion of 400ml or more, while the "improvement rate at discharge" requires extracting the Glasgow Coma Scale (GCS) score from both the admission and discharge records, and comparing these two scores. To address this issue, we propose a knowledge-enhanced quality control indicator representation language to express these indicators, enabling their automatic conversion and calculation using EMR-LLM.

The prompt of task decomposition

Instruction

As a medical expert, you need to break down the calculation of the indicator into subtasks based on its {definition} and {formula}.

ICL example

Input: "*Definition*: The proportion of successful CAG immediately after CIS, relative to the total number of CIS. A successful CAG is ... residual stenosis of less than 20% ... or less than 50% after PTCA, with a TIMI grade 3..... ;
Calculation formula:
DIR = $\frac{\text{Patients with discharge GCS or NIHSS better than admission score}}{\text{Total ICH patients}}$ "

Output: First, ... assessment. Second, ... evaluated. Hence, Fact 1: PCI or PTCA. Fact 2: Residual stenosis. Fact 3: TIMI grade.

Input: "*Definition*: Proportion of patients with discharge GCS/NIHSS better than admission score relative to total ICH patients.;
Calculation formula:
Success rate of CAG after CIS = $\frac{\text{Number of successful CAG after CIS}}{\text{Total number of CIS}}$ "

Figure 2: Example prompt for indicator calculation decomposition.

Indicator Representation Language Design Given an indicator $I = \langle$ *definition*, *calculation formula* $\rangle$, we expand it to a more comprehensive representation language:

$$I = \langle definition,\ calculation\ formula,\ fact,\ logical\ rule,\ EMR\ field,\ knowledge \rangle. \quad (3)$$

Here, *fact* refers to the statistical item in the calculation formula. *Logical rule* (Xing, Lu, and Yu 2024) denotes the method of integrating these statistical items. *EMR field* corresponds to the field in EMR related to indicator I, which clarifies the mapping between the indicator definition and the EMR field. *Knowledge* includes additional relevant information about indicator I. For instance, a higher GCS score indicates a better patient condition. The advantage of this expanded language (abbreviated as $I = \langle d, c, f, l, e, k \rangle$) is that it clarifies and details the indicator calculation process, making it easier for the LLM to understand and process and enhancing the interpretability of the results.

Automatic Indicator Representation Conversion However, due to the large number of quality control indicators involved, manually crafting a standardized representation for each one is extremely time-consuming and labor-intensive. To address this, we design an automatic conversion method for quality control indicators based on an LLM. Specifically, we first use the *definition* and *calculation formula* of the indicator as the input for the LLM. By employing the least-to-most prompting method (Zhou et al. 2022), as shown in Figure 2, we decompose the indicator calculation into several subtasks, which we refer to as *facts*. Secondly, we input the *facts*, *definition* and *calculation formula* into the LLM

[1]https://github.com/CBLUEbenchmark/CBLUE
[2]http://old.openkg.cn/dataset/yidu-n7k

The prompt of logical rule generation

Instruction

As a medical expert, please generate logical rules for combining {facts} based on the indicator's {definition} and {formula}.

ICL example

Input: "*Definition*: The proportion of successful CAG immediately after CIS, relative to the total number of CIS. A successful CAG is ... residual stenosis of less than 20% ... or less than 50% after PTCA, with a TIMI grade 3.... ;
Calculation formula:
Success rate of CAG after CIS $= \frac{\text{Number of successful CAG after CIS}}{\text{Total number of CIS}}$;
Facts : Fact 1: PCI or PTCA. Fact 2: The residual stenosis percentage. Fact 3: The TIMI grade. "

Output: If PCI, then residual stenosis less than 20%, if PTCA, then residual stenosis less than 50%. TIMI grade 3.
Converted into formal language:
Rule 1: If PCI, Fact2 < 20%; if PTCA, Fact2 < 50%.
Rule 2: TIMI grade is 3.
Rule 3: Rule1 and Rule2.

Input: "*Definition*: Proportion of patients with discharge GCS/NIHSS better than admission score relative to total ICH patients.;
Calculation formula:
DIR $= \frac{\text{Patients with discharge GCS or NIHSS better than admission score}}{\text{Total ICH patients}}$;
Facts : Fact1: Admission GCS/NIHSS score; Fact2: Discharge GCS/NIHSS score"

Figure 3: Example prompt for logical rule generation.

and use the ICL method (Wies, Levine, and Shashua 2024) to obtain the *logical rule* (Zhang and Yu 2024), as illustrated in Figure 3. Next, we apply the sentence-transformer algorithm (Reimers and Gurevych 2019) to calculate the matching score between the indicator definition and the content of each field in EMR, selecting the Top-K highest-scoring fields as the *EMR field*. Finally, vector matching is performed (Zhang et al. 2024) to retrieve the top 1 passage most relevant to each indicator's fact from quality control documents and medical books as *knowledge*. After this, the results are manually reviewed to finalize the representation language of the indicators. The pseudo-code for the above process can be found in Lines 2-14 of Algorithm 1.

Indicator Calculation Based on this standardized language, EMR-LLM can determine whether each EMR meets the indicator requirements. Firstly, the relevant field content in the EMR related to the indicator is identified. Using this content, CoT-prompting (Wei et al. 2022b) guides the LLM in generating the statistical results for each fact, as shown in Figure 4. These results are then processed through predefined logical rules and knowledge to produce the final judgment for the EMR. The pseudo-code for the above calculation process can be found in Lines 16-20 of Algorithm 1.

Application Development and Deployment

The IMQC platform was implemented using the Pytorch framework. To verify the effectiveness of our proposed EMR-LLM and quality control indicator calculation method, we compared it offline with six pre-trained language models (PLMs) and five LLM-based reasoning methods prior to making the deployment decision. The PLMs includes BERT (Devlin et al. 2019), ChatGLM3-6b,[3] Mistral-7B-insturct,[4] Llama3-8b-instruct,[5] Huatuo2-7B,[6] and PULSE-20B[7] (the base model of our EMR-LLM). The reasoning methods are:

The prompt of fact calculation

Instruction

As a medical expert, please provide the result of subtask {Fact} based on the {EMR_content}.

ICL example

Input: "*Fact* : Admission GCS/NIHSS score.;
EMR_content : Admission Record--Departmental Examination: Somnolent, in a supine position, ..., normal muscle tone, bilateral Babinski sign negative, GCS 13."

Output: *Intermediate result:* The observation of the medical record reveals "GCS 13," indicating that the patient's admission score was 13.
Final_result: GCS, 13

Input: "*Fact* : Admission GCS/NIHSS score.;
EMR_content : Admission Record - Departmental Examination: Somnolent, supine position, ..., unable to walk in a straight line, GCS 13. "

Figure 4: Example prompt for fact calculation.

- **Zero-shot** (Wei et al. 2022a): This method provides only instructions, and the model responds with them.
- **ICL** (Wies, Levine, and Shashua 2024): We select samples whose indicators match the indicator of the current input, and randomly choose one as the ICL sample.
- **CoT** (Wei et al. 2022b): We add "Let's think step by step" to the original instruction to guide the LLM's reasoning.
- **Least-to-Most** (Zhou et al. 2022): To mitigate the shortcomings of CoT in handling complex tasks, this method breaks down complex tasks into multiple subtasks.
- **Self-Consistency** (Wang et al. 2023): This method uses multiple CoT reasoning paths to generate various results and employs majority voting to produce the final answer.

To evaluate the effectiveness of our proposed method, we built a manually verified test set for the QCIC task. The test set[8] covers 108 indicators of 42 single diseases, and the original EMR data come from the open-source EMR dataset[9], and de-identified EMRs provided by medical institutions. We utilized different baseline PLMs and inference methods

[3]https://github.com/THUDM/ChatGLM3
[4]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
[5]https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct
[6]https://huggingface.co/FreedomIntelligence/HuatuoGPT2-7B
[7]https://huggingface.co/OpenMEDLab/PULSE-20bv5
[8]Due to privacy concerns regarding hospital data, de-identified samples of the EMR task datasets can be requested by contacting the author via email.
[9]https://github.com/FreedomIntelligence/CMB

Algorithm 1: Quality Control Indicator Calculation

Input: Indicator $I = \langle d, c, f, l, e, k \rangle$, EMR, medical corpora

Output: 0/1 (whether EMR meets the indicator requirement)

// Automatic Indicator Representation Conversion
$\{f_i\}_n \leftarrow \texttt{Least-to-Most}(d,\ c)$
Examples $\leftarrow \texttt{ICL}(\{f_i\}_n,\ d,\ c)$
$\{l_j\}_m \leftarrow \texttt{LLM}\,(\{f_i\}_n,\ d,\ c,\ \textit{Examples})$
for each field $\in$ EMR **do**
    $Score_{field} \leftarrow \texttt{Sen\_tran}(d,$ field's content$)$
end for
$e \leftarrow \{field \mid \texttt{Top-K}(Score_{field})\}$
for $f_i \in \{f_i\}_n$ **do**
    for each passage $\in$ medical corpora **do**
        $Score_{passage} \leftarrow \texttt{Vec\_match}(f_i,\ passage)$
    end for
    $k_{f_i} \leftarrow \{passage \mid \texttt{max}(Score_{passage})\}$
end for
// Indicator calculation for EMRs
e's content $\leftarrow \texttt{EMR}(e)$
for $f_i \in \{f_i\}_n$ **do**
    $Answer_i \leftarrow \texttt{LLM}(f_i, \text{e}'scontent)$
end for
Result $\leftarrow \texttt{Calculation}(\{Answer_i\}_n,\ \{l_j\}_m)$

to calculate each quality control indicator, and evaluated the performance of the models in terms of accuracy and the F1 score. The experimental results are shown in Table 2. It can be observed that EMR-LLM with our calculation method outperforms all competitors on the test set in both accuracy and F1 score, demonstrating its effectiveness. Specifically, our method shows an improvement of 4.78% in accuracy and 5.24% in F1 score over the best baseline, EMR-LLM with ICL. Our EMR-LLM performs better than other LLMs on the QCIC task, indicating the effectiveness of our continual pre-training and instruction tuning. For example, our EMR-LLM with ICL achieves 4.15% higher in accuracy and 3.85% higher in F1 score compared to PULSE-20B with ICL. The experimental results demonstrate that the proposed model can more accurately compute quality control indicators, thereby making it suitable for deployment. As a result, our proposed methods were chosen for integration into the AI engine of the IMQC platform.

To verify the effectiveness of EMR-LLM, we compare it with other LLMs across our constructed EMR tasks, including QCIC, EMR_ID, CSD, LCC, and TS. These LLMs were not fine-tuned on any data. We use accuracy as the evaluation metric. The experimental results are presented in Table 3. It can be observed that our constructed EMR-LLM outperforms all competitors on every EMR task, demonstrating its suitability as the base model for our platform. Furthermore, compared to PULSE-20B, our LLM shows significant improvement, indicating the effectiveness of our continual pre-training and instruction-tuning approaches.

Figure 5 shows the homepage of our IMQC platform,

Model	Method	Accuracy	F1 score
BERT	FT	74.31	66.39
ChatGLM3-6B	ICL	71.47	68.64
Mistral-7B	ICL	69.67	68.79
Llama3-8B	ICL	68.47	67.55
Huatuo2-7B	ICL	65.77	64.06
PULSE-20B	ICL	84.38	83.06
EMR-LLM	Zero-shot	85.58	83.56
EMR-LLM	ICL	88.53	86.91
EMR-LLM	CoT	86.49	85.09
EMR-LLM	L2M	87.68	85.84
EMR-LLM	SC	87.38	85.68
EMR-LLM	Ours	**93.31**	**92.15**

Table 2: Comparison results (%) of different baseline PLMs and inference methods on the quality control indicator calculation task. "FT", L2M, and SC stand for "fine-tuning", "Least-to-Most", and "Self-Consistency", respectively.

Model	QCIC	EMR_ID	CSD	LCC	TS
ChatGLM3-6B	56.15	49.12	64.28	71.42	54.48
Mistral-7B	51.35	74.23	57.14	59.29	12.26
Llama3-8B	59.15	65.34	63.14	64.21	16.51
Huatuo2-7B	62.76	48.89	60.71	67.21	29.90
PULSE-20B	65.16	87.34	71.42	79.42	81.49
EMR-LLM	**85.58**	**90.12**	**92.85**	**89.28**	**96.91**

Table 3: Comparison results (Accuracy %) of different LLMs on our constructed EMR tasks in zero-shot setting.

which includes statistics on the number of indicators, the distribution of indicator calculation results, and the calculation results for a specific hospital. 1) Indicator Statistics and Indicators of a Hospital: These two parts provide the total number of indicators for departments and single diseases across the platform, as well as all indicators for a specific hospital. 2) Indicators' Results of Hospitals: This part displays the distribution of calculation results for all indicators across different months. 3) Indicators' Results of a Hospital: This part provides the specific calculation results for all indicators within a particular hospital.

Figure 6 illustrates the details page of our quality control indicator calculation results. This page is designed to assist users in viewing the calculation process and reviewing the results. It comprises three parts: 1) indicator calculation results, 2) indicator representation language, and 3) detailed results for each EMR. Specifically, the indicator representation language part displays the indicator definition, facts, EMR fields, logical rules, knowledge, and source documents. For EMRs, users can also view the calculation process, including the facts and logical rules associated with each indicator. This ensures traceability of the calculation basis, enhancing the transparency and credibility of the results. In addition, an audit function is incorporated to verify the accuracy of the results.

Figure 5: The IMQC platform homepage for medical quality control indicator calculations.

Note that the pages shown in Figures 5 and 6 have been translated into English for readers who do not speak Chinese. The actual system interfaces are in Chinese.

Application Use and Payoff

In this section, we elaborate on the practical impact and pivotal role of the IMQC platform. This platform has pioneered an innovative and effective method for calculating quality control indicators, significantly enhancing the efficiency of this traditionally labor-intensive computation process. The proposed EMR-LLM allows hospitals to accurately and promptly identify and address deficiencies in their medical services. Consequently, the Medical Quality Control Management Center has shifted from traditional periodic inspections to real-time supervision, enabling swift responses to hospital treatment quality assessments. Ultimately, this methodology ensures that the entire process of medical quality control is gradually realized, meeting society's high demand for quality medical services.

The IMQC platform has significantly enhanced the efficiency of medical quality control through the comprehensive integration of LLM technology. Since its deployment at the Medical Quality Control Management Center in February 2024, it has processed 1,245 quality control indicators for secondary- and tertiary-level hospitals. After expert review, our quality control indicator calculation results achieve an accuracy of 93.31%, which is slightly lower than the accuracy of manual calculations by expert clinicians, often exceeding 95% (100% accuracy is difficult to achieve due to energy limitations in large-scale calculations). In an experimental environment using a single H800 GPU card, our platform can process 1,000 EMRs per hour, compared to about 10 EMRs per hour manually, making it 100 times more efficient. In addition, conservative estimates indicate that our method can save 3.42 million RMB per month in manpower expenses across hospitals adopting it, compared to traditional reporting practices.

Maintenance

As time goes on, IMQC may introduce more quality control indicators and integrate additional EMR data. The platform's modular design ensures that updates can be implemented without affecting the AI engine. Hence, our AI algorithms have not required any modifications since the platform's deployment in February 2024.

Lessons Learned During Deployment

Throughout the deployment process of the IMQC platform, several key lessons have been learned.

Firstly, base model selection and fine-tuning are important. The core of the IMQC platform relies on its powerful LLM support. With numerous open-source LLMs available, selecting a suitable base model is crucial as different LLMs exhibit varying performance on EMR tasks. Open-source LLMs that have not been fine-tuned often fail to meet the

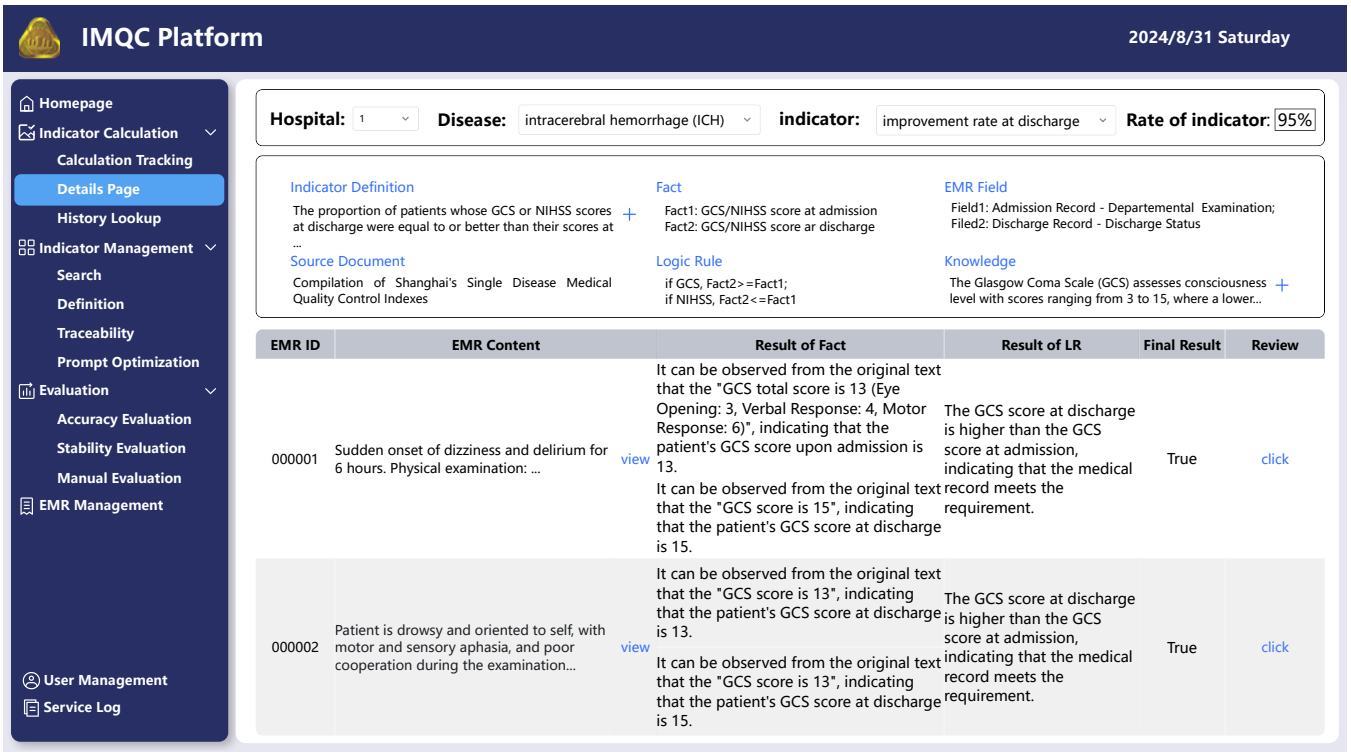

Figure 6: The IMQC platform interface for quality control indicator calculation process.

specific needs of downstream tasks. Therefore, it is essential to manually annotate a significant amount of domain-specific data to fine-tune the LLM.

Secondly, the transparency of quality control indicator calculation is important. Quality control indicators are vital for measuring the level of hospital treatment. Ensuring that users trust the results of these calculations is crucial. To enhance the interpretability of model results, we prioritize allowing the LLM to combine facts and logical rules through multi-step reasoning, despite the additional time costs. This approach significantly improves user trust in the results.

Last but not least, User-friendly interface design is important. The primary users of the IMQC platform are medical experts rather than IT professionals, who may lack relevant technical knowledge. Consequently, we emphasize clear modularity in platform design and prioritize designing the interface from the perspective of medical experts. This ensures that users can quickly understand the operation process and interpret the output results effectively.

Conclusions and Future Work

In this paper, we developed an approach to leverage LLM technology to address the challenges in calculating medical quality control indicators. We have developed an IMQC platform to assist hospitals, the Medical Quality Control Management Center, and the Health Management Institute in automating these indicator calculations. IMQC analyzes EMR data based on the constructed EMR-LLM, enabling the reasoning of quality control indicators. Since its deployment in February 2024, the platform has been widely adopted for medical quality control management in secondary- and tertiary-level hospitals in Shanghai. In addition to ensuring the reliability of quality control indicator calculations, it significantly reduces labor costs.

In subsequent research, we plan to explore retrieval-augmented generation (RAG) techniques (Xiong et al. 2024) to supplement medical knowledge and address missing information when new indicators are introduced. In addition, we plan to use LLMs to standardize medical record writing across hospitals to handle diverse styles and human errors, and ensure the objectivity of indicator calculation results. Ultimately, our goal is to calculate indicators for different diseases and EMR data from various hospitals, obtaining reliable and interpretable results. This approach will enhance data accuracy and consistency, leading to more informed and effective healthcare service provision.

Acknowledgments

This research is supported by National Natural Science Foundation of China (No. 62306112); National Research Foundation, Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-019); the RIE 2020 Advanced Manufacturing and Engineering (AME) Programmatic Fund (No. A20G8b0102), Singapore; and Shanghai Sailing Program (No. 23YF1409400), and Shanghai Pilot Program for Basic Research (No. 22TQ1400100-20).

References

Anderson, J. C.; Butterly, L. F.; Weiss, J. E.; and Robinson, C. M. 2017. Providing data for serrated polyp detection rate benchmarks: an analysis of the New Hampshire Colonoscopy Registry. *Gastrointestinal endoscopy*, 85(6): 1188–1194.

Bae, J. H.; Han, H. W.; Yang, S. Y.; Song, G.; Sa, S.; Chung, G. E.; Seo, J. Y.; Jin, E. H.; Kim, H.; and An, D. 2022. Natural language processing for assessing quality indicators in free-text colonoscopy and pathology reports: Development and usability study. *JMIR Medical Informatics*, 10(4): e35257.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Hsu, W.; Han, S. X.; Arnold, C. W.; Bui, A. A.; and Enzmann, D. R. 2016. A data-driven approach for quality assessment of radiologic interpretations. *Journal of the American Medical Informatics Association*, 23(e1): e152–e156.

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

Liu, J.; Wang, M.; Wang, C.; Liang, J.; Chen, L.; Jiang, H.; Xiao, Y.; and Chen, Y. 2021. Learning term embeddings for lexical taxonomies. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 6410–6417.

Øvretveit, J. 2001. Quality evaluation and indicator comparison in health care. *The International journal of health planning and management*, 16(3): 229–241.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.

Raju, G. S.; Lum, P. J.; Slack, R. S.; Thirumurthi, S.; Lynch, P. M.; Miller, E.; Weston, B. R.; Davila, M. L.; Bhutani, M. S.; Shafi, M. A.; et al. 2015. Natural language processing as an alternative to manual reporting of colonoscopy quality metrics. *Gastrointestinal endoscopy*, 82(3): 512–519.

Raju, G. S.; Vadyala, V.; Slack, R.; Krishna, S. G.; Ross, W. A.; Lynch, P. M.; Bresalier, R. S.; Hawk, E.; and Stroehlein, J. R. 2013. Adenoma detection in patients undergoing a comprehensive colonoscopy screening. *Cancer medicine*, 2(3): 391–402.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992.

Ross, W. A.; Thirumurthi, S.; Lynch, P. M.; Rashid, A.; Pande, M.; Shafi, M. A.; Lee, J. H.; and Raju, G. S. 2015. Detection rates of premalignant polyps during screening colonoscopy: time to revise quality standards? *Gastrointestinal endoscopy*, 81(3): 567–574.

Tamang, S.; Patel, M. I.; Blayney, D. W.; Kuznetsov, J.; Finlayson, S. G.; Vetteth, Y.; and Shah, N. 2015. Detecting unplanned care from clinician notes in electronic health records. *Journal of Oncology Practice*, 11(3): e313–e319.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Wang, Y.; Wang, L.; Rastegar-Mojarad, M.; Moon, S.; Shen, F.; Afzal, N.; Liu, S.; Zeng, Y.; Mehrabi, S.; Sohn, S.; et al. 2018. Clinical information extraction applications: a literature review. *Journal of biomedical informatics*, 77: 34–49.

Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022a. Finetuned Language Models are Zero-Shot Learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wies, N.; Levine, Y.; and Shashua, A. 2024. The learnability of in-context learning. *Advances in Neural Information Processing Systems*, 36.

Williams, S. M.; Parry, B. R.; and Schlup, M. 1992. Quality control: an application of the cusum. *BMJ: British medical journal*, 304(6838): 1359.

Xing, P.; Lu, S.; and Yu, H. 2024. Federated Neuro-Symbolic Learning. In *Forty-first International Conference on Machine Learning*.

Xiong, G.; Jin, Q.; Lu, Z.; and Zhang, A. 2024. Benchmarking retrieval-augmented generation for medicine. *arXiv preprint arXiv:2402.13178*.

Zhang, H.; Wang, Y.; Chen, Q.; Chang, R.; Zhang, T.; Miao, Z.; Hou, Y.; Ding, Y.; Miao, X.; Wang, H.; et al. 2024. Model-enhanced vector index. *Advances in Neural Information Processing Systems*, 36.

Zhang, Q.; Chen, Q.; Li, Y.; Liu, J.; and Wang, W. 2021. Sequence model with self-adaptive sliding window for efficient spoken document segmentation. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 411–418. IEEE.

Zhang, Y.; and Yu, H. 2024. LR-XFL: Logical Reasoning-based Explainable Federated Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 21788–21796.

Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.; et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.