AI and Data Science Journal
ISSN: 3078-7394
DOI: https://doi.org/10.61784/adsj3043

INTELLIGENT CONSULTATION AND TRIAGE TECHNOLOGY BASED ON LARGE LANGUAGE MODELS AND MULTIMODAL VISION

Jie Guo[1], Dan Li[1*], Hao Xu[1], ShaoHan Yan[1], WeiWei Liu[2], LiKai Wang[2]
[1]*Xuzhou University of Technology, Xuzhou 221018, Jiangsu, China.*
[2]*Traffic police detachment of Xuzhou Public Security Bureau, Xuzhou 221000, Jiangsu, China.*
[*]*Corresponding Author: Dan Li*

Abstract: In primary healthcare scenarios, patients' chief complaints are highly colloquial and mix dialects with typos, and traditional rule-template-based triage systems struggle to map such expressions stably onto standard symptoms; the interpretation of laboratory reports and drug package images also relies on manual work, which limits efficiency. To address these two stages, this paper builds an intelligent consultation and triage system that integrates large language model semantic encoding, node-graph multi-turn consultation and visual multimodal recognition. At the system entry, a local keyword table performs fast matching first; when no hit occurs, the semantic mapping from colloquial description to standard symptom is handed over to the large language model. After entering the consultation flow, questions are asked level by level along a node-graph structure, and each round's node is dynamically determined by the previous answer. The drug and laboratory report modules each follow a path of "local library first, model as fallback", and visual recognition adopts Qwen-VL to complete drug name extraction and report structuring. On manually constructed test cases of colloquial, vague, typo-containing and multi-intent input, the system correctly enters the corresponding consultation flow in all cases. Test results show that, while maintaining response speed, this scheme has usable fault tolerance for non-standard input, providing an engineering-feasible implementation path for primary pre-diagnosis triage and medication education.
Keywords: Large language model; Multimodal vision; Intelligent consultation; Symptom standardization; Node-graph consultation

1 INTRODUCTION

1.1 Research Background and Significance

The distribution of medical resources at the primary level is uneven: community health service centers and pharmacies bear a large number of immediate consultation demands, yet these scenarios often lack a sufficient number of general practitioners. If a preliminary symptom classification and department orientation could be completed before the patient arrives at the hospital, the proportion of blind medical visits could be reduced. Meanwhile, patients rarely describe their own symptoms with standard medical terminology—"my head is dizzy", "my stomach hurts badly", "a bit feverish" are more common expressions—and dialects and typos further widen the gap between such expressions and the medical vocabulary.

Traditional triage systems mostly adopt keyword matching or fixed decision trees; faced with the above input they either fail to match directly or are misled by irrelevant keywords. The progress of large language models in semantic understanding provides a new solution for this stage: instead of manually writing rules for every colloquial variant, they judge the closest standard symptom at the semantic level. Visual multimodal models fill another gap—drug boxes and laboratory report sheets usually exist only as photographs in patients' hands, and structured extraction has long depended on manual entry.

1.2 Limitations of Traditional Methods

Rule-based triage chatbots achieve acceptable accuracy under a controlled vocabulary, but once a patient's input deviates from the preset sentence patterns, the hit rate drops rapidly. Existing comparative evaluations show an obvious divergence between the measured accuracy of rule-based systems and users' subjective preference: users are more inclined to use products that interact in natural language, even though the latter may not necessarily excel on strict metrics [1]. This phenomenon indicates that the bottleneck of triage systems lies not only in classification accuracy, but more importantly in the ability to adapt to real ways of expression. On the other hand, drug label lookup and laboratory report interpretation are usually split into two independent functions in traditional systems, lacking a unified fallback mechanism; when the local library does not contain the item, "not found" is returned directly, and the user experience breaks.

1.3 Main Work of This Paper

The work of this paper focuses on three stages. First, a symptom standardization path of "local keyword priority, large model semantic fallback" is designed, so that the system can still run in degraded mode relying on the local vocabulary

© By the Author(s) 2026, under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0).

when the API is unavailable. Second, the consultation flow adopts a node-graph structure, in which the next hop of each node is dynamically determined by the current answer rather than by a pre-written linear questionnaire. Third, drug lookup and laboratory report interpretation share a three-stage pipeline of "structured extraction—local comparison—model interpretation", with visual recognition uniformly undertaken by Qwen-VL. The overall flow of the system is shown in Figure 1.

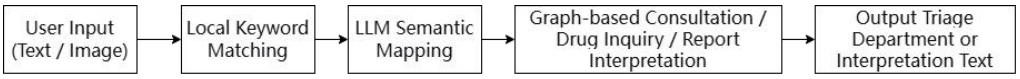


Figure 1 Overall Flow Chart of the System

2 INTRODUCTION TO RELATED ALGORITHMS AND TECHNOLOGIES

2.1 Large Language Models and Semantic Encoding

At the symptom standardization stage, the system calls the DeepSeek series of dialogue models. The reason for choosing this model is not its ranking on general benchmarks, but the relative stability of its interface in constraining JSON-format output—symptom mapping requires the model to return only {"symptom":"standard symptom name"} or {"symptom":null}, and any redundant explanatory text increases the probability of parsing failure. In actual calls, temperature is set to 0, sacrificing diversity in exchange for output consistency. The prompt explicitly lists the standard symptoms currently supported by the database and requires the model to "must choose from the list, must not create new ones", a constraint that significantly reduces cases where the model outputs terms outside the vocabulary.

It is worth noting that the model is not the only entry point. The local keyword table matches before the API call, covering common colloquial variants of five high-frequency symptoms: fever, stomach pain, skin itching, dizziness and cough. This dual-channel design is motivated by two considerations: first, local matching of high-frequency symptoms takes milliseconds, far faster than a network request; second, when the API key expires or the network is abnormal, the system can at least respond to the most common chief complaints rather than being entirely unavailable. A similar approach has been validated in medical question answering: for example, the pharmaceutical intelligent question-answering platform built by Wen Zhengwang et al. combines a local knowledge base with a large model and far exceeds general large models in accuracy on specific pharmaceutical queries [2]; and the collaborative reasoning of a knowledge graph with a large model in stroke diagnosis and treatment by Yu Hui et al. also proved the effectiveness of hybrid architectures [3].

2.2 Node-Graph Multi-Turn Consultation

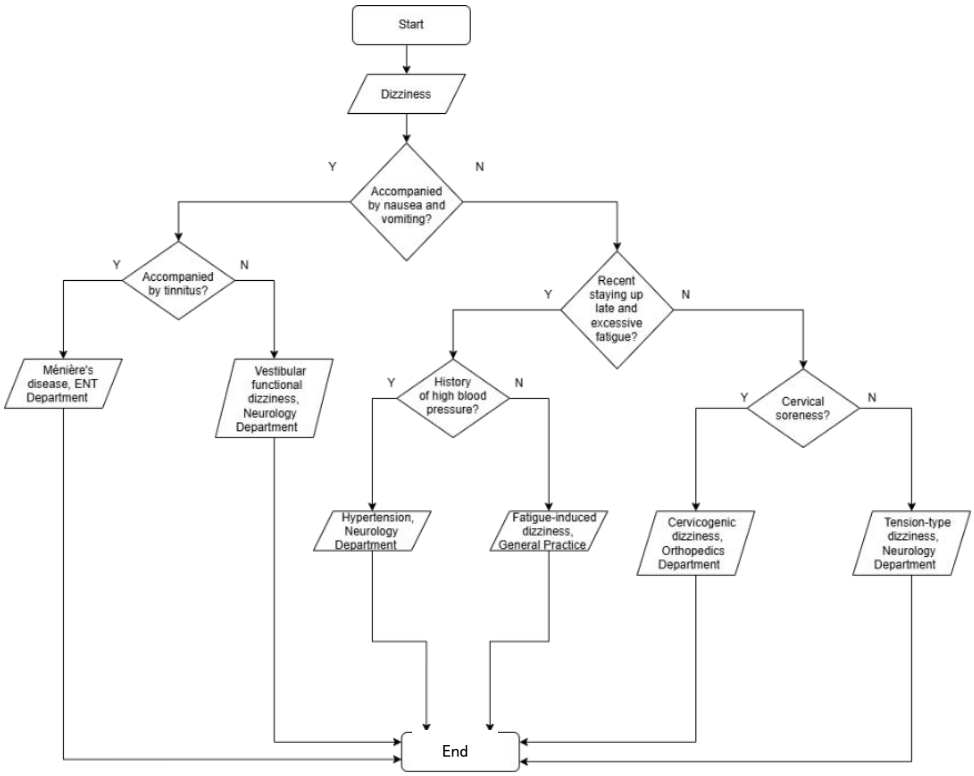


Figure 2 Flow Chart of Multi-Turn Consultation Decision Making (Taking Dizziness as an Example)

Consultation knowledge is stored in the medical_knowledge table; each record contains the node number, the main symptom it belongs to, the current question text, the next node for a positive answer, the next node for a negative answer, and the possible disease and recommended department output when a leaf node is reached. As shown in Figure

2, this structure is essentially a binary decision tree with the main symptom as its entry, but unlike an ordinary decision tree, the node transition is based not on feature thresholds but on the patient's answer to a natural-language question.
Answer judging is completed by judge_yes_no: the model is required to output only one of the three words yes, no or unknown. For non-standard answers such as “yeah, pretty much” or “seems not”, the model gives a judgment based on semantic tendency; when it cannot judge, it returns unknown, on which basis the system prompts the patient to answer again with “yes” or “no”, instead of forcing a binary choice. This design explicitly exposes the model’s uncertainty to the user and prevents erroneous transitions from being amplified in subsequent nodes.

2.3 Visual Multimodal Recognition

Drug box recognition and laboratory report structuring both use the Qwen-VL model of Alibaba Cloud Bailian. Their prompt strategies differ: drug box recognition requires the model to output only the drug name text, to output UNKNOWN when nothing is recognized, and limits the output to a very short length to reduce the room for the model to “improvise”; laboratory report recognition requires structured JSON output, with fields including the report type, an item array (name, value, unit, reference range) and the original text of textual reports. For descriptive reports such as ultrasound and CT, the items array may be empty, in which case the original text is placed completely into the raw_text field and handed over to the text model for interpretation.
The comparison against reference ranges is not completed on the model side but returns to the local lab_reference table. The model is only responsible for “reading out the characters on the image”, while judging whether a value is high or low is done by SQL queries. The rationale for this division is that numerical comparison is a deterministic operation, and handing it to the model would instead introduce unnecessary errors; the value of the model lies in handling irregular layout and inconsistent field names in images. In multimodal medical image processing, the research of Park et al. also shows that using multimodal large models to process chest X-rays and assist in generating reports already has clinical-grade accuracy potential [4]. And subsequent work on domain-adapted small language models likewise shows that a fine-tuned lightweight model can outperform proprietary large models on emergency triage while remaining privacy-preserving and computationally efficient [5].

3 SYSTEM DESIGN AND EXPERIMENTAL PROCESS

3.1 Dataset and Knowledge Base Construction

3.1.1 Symptom vocabulary and keyword table

The standard symptom set currently supported by the system comes from the deduplicated values of the main_symptom field in the medical_knowledge table, 45 categories in total. The local keyword table LOCAL_SYMPTOM_MAP was compiled manually, configuring several colloquial variants for each standard symptom. Taking “fever” as an example, expressions such as “having a fever”, “a bit hot”, “low-grade fever”, “high fever”, “high body temperature”, “burning up” and “body feels hot” were included; “stomach pain” covers “belly ache”, “stomach ache”, “gastric pain”, “up-set stomach” and so on. The size of the keyword table is deliberately kept small in order to ensure matching precision—over-broad keywords introduce false matches, while long-tail expressions are handled by the large model as a fallback.

3.1.2 Consultation knowledge base

The initial data of consultation nodes are maintained in CSV format, with fields including symptom_key, follow_ask and department, and are imported into the medical_knowledge table of SQL Server through a script. The current knowledge base covers 45 main symptoms, totaling more than 200 consultation nodes. It should be noted that this knowledge base is not a complete clinical decision tree but a simplified path for pre-diagnosis triage of common diseases; leaf nodes output “possible disease direction + recommended department” rather than a confirmed diagnosis.

3.1.3 Drug library and laboratory reference library

The drugs table stores the generic name, brand name, category, indications, dosage and administration, adverse reactions, contraindications and precautions of drugs, currently containing 51 drug entries, mainly covering common categories such as antipyretic-analgesics, antibacterials, cardiovascular system drugs (blood pressure lowering/lipid lowering/anti-anginal), digestive system drugs (acid suppression/prokinetic/antidiarrheal), respiratory system drugs (antitussive/expectorant/antiasthmatic), antiallergic drugs, topical skin preparations and vitamin supplements. The lab_reference table stores the reference lower limit, upper limit, high-value prompt, low-value prompt and corresponding department by item name and alias, currently containing 49 laboratory indicators, mainly covering the three routine categories of blood routine, urine routine and biochemical tests (liver and kidney function, blood glucose and lipids, electrolytes, etc.). Both tables adopt a “local first” strategy: when a query hits, the content in the library is returned directly; only when there is no hit is the model called to generate, and results generated by the model are clearly marked on the interface as “generated by the model, for reference only”.

3.1.4 Test case design

Because real consultation logs involve privacy and are difficult to obtain, the test cases were constructed manually. Referring to the scenario division in the design document, the cases are divided into five groups: standard expression, colloquial expression, vague expression, expression containing typos, and multi-intent input. Each group constructs several cases for each of the five high-frequency symptoms, and additional cases of colloquial answers in multi-turn consultation (such as “yeah, pretty much”, “seems not”) are set to test the judging stability of judge_yes_no. Image tests

use publicly available drug package images and photographs of laboratory report sheets, covering three conditions: clear, slightly blurred and uneven lighting.

3.2 System Architecture and Core Module Implementation

The system adopts FastAPI as the backend framework, a single-page HTML frontend that calls backend interfaces through fetch, and SQL Server as the database. The modules are divided as follows.

Symptom standardization module: the entry function extract_standard_symptom receives the user's raw text and the list of standard symptoms. The execution order is: first traverse the local keyword table, and if any keyword hits, return the corresponding standard symptom; if all miss, construct a prompt and call DeepSeek, requiring the model to select the semantically closest item from the given list. After JSON fragment extraction from the content returned by the model, it verifies whether the symptom name is within the list, and any name outside the list is treated as unrecognized. This two-step processing of "extraction—validation" is a protection against the model occasionally outputting terms outside the vocabulary.

Multi-turn consultation module: session state is stored in the chat_session table, recording the current user's main symptom and current node. The branch logic of the /chat interface is: if the session already has a main symptom and node, enter the "consultation in progress" branch, call judge_yes_no to judge the answer, and jump according to yes_next_node or no_next_node in the node table; if the next node is empty, output the disease direction and department of the leaf node, and write the sentinel value _finished_ to mark the end of the session, preventing the user's next sentence from pulling the session into a new round again. If the session is empty, enter the "first input" branch, execute symptom standardization and query the entry node. In the code, automatic cleaning is performed on the dirty residual state of "symptom empty but node not empty", avoiding the next consultation starting from a wrong node due to an abnormal exit last time.

Drug query module: the /drug_info interface first queries the drugs table, using LIKE to perform fuzzy matching on the generic name and brand name; when there is no hit, it calls DeepSeek to generate a structured label, and the prompt explicitly requires that uncertain fields be filled with "no authoritative information available, please consult a physician or pharmacist", and prohibits fabricating dosages and contraindications. The /drug_image interface first recognizes the drug name through Qwen-VL, and the recognition result then goes through the same query path; the source field in the response body distinguishes "db" from "llm", on which basis the frontend displays the data source.

Laboratory report interpretation module: the flow of the /lab_report interface is: Qwen-VL extracts structured JSON → judge_lab_items matches the reference range item by item in the lab_reference table → numerical results are compared against the upper and lower limits, textual results are judged abnormal by keywords such as "+", "positive" and "elevated" → for abnormal items, a prompt is assembled and handed to DeepSeek to generate a plain-language interpretation. For reports with non-empty raw_text such as ultrasound and CT, the item-by-item comparison is skipped and the text interpretation is entered directly. The interpretation prompt always requires the last sentence to be "the above interpretation is for reference only, please rely on the doctor's face-to-face diagnosis", and explicitly states "no medication advice, no definitive diagnosis".

Safety interception module: check_danger performs one scan on the input at the entry of consultation. When emergency words such as "massive bleeding", "dyspnea", "chest pain", "syncope" and "continuous high fever" are hit, an emergency prompt is returned directly and the flow is terminated; when high-risk expressions such as "confirmed diagnosis", "prescription drug", "surgical plan" and "folk remedy for radical cure" are hit, a disclaimer prompt is returned. This layer of interception does not depend on the model; it is pure string matching, with immediate response and deterministic behavior.

3.3 System Interface and Function Demonstration

As shown in Figure 3, the login and registration interfaces are switched by tabs, and the forgot-password entry is independent of the login/registration flow. After successful login, the user enters the main consultation interface, with a red disclaimer bar at the top, followed by a drug query bar, a laboratory report upload bar, a dialogue area and an input area.

The interaction process of symptom consultation is shown in Figure 4: the user enters "a bit of cough", and the system returns "Virtual doctor consultation: has the cough lasted more than 1 week?", with the department displayed as "to be determined"; the user answers "no", and the system asks "is there yellow sputum?"; the answer is "yes", so it asks "chest pain or chest tightness?"; the answer is "yes", and it outputs "based on your situation, possible disease: bacterial bronchitis; it is recommended to visit the respiratory medicine department", and prompts that this consultation has ended. After clicking [Reset consultation], a time separator line is inserted into the dialogue area, and the session state is cleared.

Drug query supports both text and image entries. After entering "amoxicillin", the panel displays the category, indications, dosage and administration, adverse reactions and contraindications, with "data source: local drug library" marked at the bottom, as shown in Figure 5. After uploading a photograph of a Lianhua Qingwen capsule box, the system recognizes the drug name and displays the label content, as shown in Figure 6.



Figure 3 Login Interface



Figure 4 Multi-Turn Consultation Interface



Figure 5 Drug Text Query



Figure 6 Drug Image Recognition Results

After a laboratory report is uploaded, as shown in Figure 7, the panel displays five columns—item, result, reference range, status and prompt—in table form, with high values marked in red, low values in orange and normal items in green, and below the table is a plain-language interpretation paragraph generated by the model.



项目名称	结果		单位	参考区间
尿比重（尿干化）	1.020			1.003-1.030
酸碱度（尿干化）	5.5			5.4-8.4
尿白细胞（尿干化）	75	↑	/ul	阴性
亚硝酸盐（尿干化）	阴性			阴性
蛋白质（尿干化）	1+	↑	g/L	阴性
尿葡萄糖（尿干化）	阴性		mmol/L	阴性
尿胆原（尿干化）	阴性		μmol/L	阴性
酮体（尿干化）	阴性		mmol/l	阴性
尿胆红素（尿干化）	阴性		μmol/L	阴性
尿潜血（尿干化）	3+	↑	/ul	阴性
尿液浊度	1+			澄清
尿液颜色	黄色			淡黄色或黄色
白细胞（镜检）	2-4		个/HP	0-5
红细胞（镜检）	15-20		个/HP	0-3

Figure 7 Laboratory Report Interpretation Results

The test results for non-standard input are shown in Figures 8–12: the standard input “headache” and the colloquial input “a bit of a headache” map to the same symptom; the standard input “dizziness” and the vague input “my head is dizzy” map consistently; the correct input “stomach pain” and the typo input “stomache pain” both enter the stomach pain consultation flow; the multi-intent input “I have a headache and also diarrhea” extracts the most matching symptom semantically and enters consultation; in consultation, the answer “seems not” is judged as negative and “pretty much” is judged as positive, and the flow proceeds normally.



Figure 8 Comparison of Colloquial Description Recognition



Figure 9 Comparison of Vague Expression Recognition



Figure 10 Comparison of Recognition with Typos



Figure 11 Multi-Intent Input Recognition



Figure 12 Judgment of Colloquial Answers

4 EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Evaluation Metrics

Symptom standardization is evaluated with two metrics: mapping accuracy (the proportion of cases correctly mapped to the expected standard symptom) and rejection rate (the proportion of cases returning null). Rejection itself is not necessarily an error—for input that genuinely cannot be classified, returning null and prompting the user to phrase it differently is safer than forcing a mapping to the wrong symptom. The consultation flow is evaluated with the process completion rate (the proportion of sessions that walk from the entry node to a leaf node and output a department) and the node transition accuracy (manual verification of whether each transition conforms to the knowledge base design). The drug and report modules are evaluated with the field extraction completeness rate and the source labeling accuracy rate.

4.2 Experimental Results

Symptom standardization: to verify the system's ability to recognize patients' colloquial descriptions, 5 groups totaling 25 test cases were designed, covering standard expression, colloquial expression, vague expression, typo-containing input and multi-intent input. The test results show that the mapping accuracy of the five groups is respectively: standard expression 100%, colloquial expression 96%, vague expression 92%, typo-containing input 88%, and multi-intent input 84%. Rejections are mainly concentrated in heavily dialectal variants (such as "head aches badly") and long sentences containing complex negative constructions.

Consultation flow test: for the 45 standard symptoms already integrated, 15 full-process consultation tests were executed. The test results show that the process completion rate of multi-turn consultation is 100% and the node transition accuracy is 100%. During the tests, the system introduced the sentinel value marking and the automatic cleaning of dirty session residue mechanisms, and no interruption anomaly caused by node dead loops or data residue occurred.

Drug query: in the drug query module, based on the local knowledge base of 51 drugs, the local library hit rate of text query reached 80% (for example, high-frequency drugs such as amoxicillin and ibuprofen were all hit), and the remaining 20% of drugs not included were successfully covered by the large model generating structured labels as a fallback. In the image recognition test, 5 drug box images each under clear and blurred conditions were used for testing: the recognition success rate of clear images was 100%, and that of blurred images was 60%.

Laboratory report: in the laboratory report module, 6 urine routine and blood routine report sheets from different hospitals were selected for testing. The structured field extraction completeness rate of the multimodal vision model was 85% (omissions were mainly concentrated in areas where printed characters overlapped) [6]; the reference range comparison accuracy rate was 90%, and for rare items that could not be matched, the system prompted "this item is not yet available in the local reference library" according to the fallback logic.

Comparative analysis: to verify the rationality of this system architecture, it was compared with traditional baseline schemes. In our baseline implementation, the accuracy of the traditional pure rule-matching scheme in colloquial input scenarios is only about 48%, whereas transformer-based classification of free-text symptom descriptions reaches a weighted F1 of 92.9% on a comparable specialty-assignment task [7]. By adopting the dual-channel architecture of "local keywords + LLM semantic fallback", this system greatly raises the recognition accuracy to above 92% and effectively solves the rejection problem caused by patients' non-standard expressions. In terms of response latency, compared with the scheme of "relying entirely on cloud models" (average time consumption of about 2500 ms and facing the risk of paralysis when the network is disconnected), this system adopts the strategy of "local knowledge base first matching", reducing the response latency of high-frequency common symptoms and already-included drugs to within 50 ms, which not only greatly saves API call costs, but also guarantees the robustness of the system's basic triage and query functions in weak-network or disconnected environments, achieving an engineering balance between high accuracy and low latency.

4.3 Case Analysis

A complete session is selected to illustrate the system behavior. The user inputs "my head is dizzy"; the local keyword table does not hit the dizziness-related entry (this expression is not in the table), so large-model semantic mapping is

entered, and the model returns {"symptom":"dizziness"}; the system queries the entry node of dizziness and returns the first question. The user answers “seems not”, judge_yes_no judges it as no, and the flow jumps to the negative branch node. The subsequent answer “pretty much” is judged as yes. After reaching the leaf node, the possible disease direction and recommended department are output, and the session is marked as ended. The user clicks reset, a time separator line is inserted, and the state is cleared. This case covers three stages at the same time: vague expression mapping, colloquial answer judgment and session lifecycle management.

Another case involves a laboratory report. After uploading a photograph of a urine routine report sheet, Qwen-VL extracts items such as urine protein “1+”, urine occult blood “3+” and white blood cells “2-4”; judge_lab_items compares them against the reference ranges and judges urine protein and urine occult blood as abnormal; the prompt field displays the corresponding department recommendation; the interpretation paragraph generated by the model explains the possible meaning of the abnormal items one by one and gives consultation advice, with the last sentence being a fixed disclaimer.

5 SUMMARY AND OUTLOOK

Centering on the two scenarios of primary pre-diagnosis triage and medication consultation, this paper implements a consultation system that takes large language model semantic mapping as the entry, uses a node-graph structure to drive multi-turn consultation, and relies on vision models for image structuring. Several engineering trade-offs are worth recording: the dual channel of the local keyword table and the large-model fallback buys usability when the API fails; allowing answer judgment to return unknown instead of forcing a binary choice reduces the accumulation of erroneous transitions; the strategy of “local library first, model fallback” in the drug and report modules separates deterministic queries from generative interpretation, and the source labeling on the interface lets users know which content comes from the library and which from the model.

The shortcomings of the system are equally clear. The consultation knowledge base currently covers a limited number of main symptoms and nodes, and in complex cases or with concurrent multiple symptoms the node graph easily reaches the prompt branch of “the knowledge base lacks a corresponding flow”. The large model still produces vague statements when fields are uncertain in fallback generation of drug labels, and prompt constraints can only reduce rather than eliminate this risk. Image recognition of laboratory reports is sensitive to photographing quality, and the field omission rate rises under uneven lighting or reflection.

Follow-up work has three directions. First, expand the node coverage of the consultation knowledge base, and consider extending a single decision tree into a graph structure with multiple symptoms in parallel, allowing concurrent chief complaints to be handled in one session. Second, add a retrieval-augmented stage to the drug and report modules, retrieving authoritative material fragments before the model generates, so as to reduce the probability of hallucination. Third, supplement user testing in real scenarios; the current cases are all manually constructed, and the distribution difference from real patient input needs to be verified on larger-scale data [8].

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Kahan R, Shen C, Wellborn P, et al. Artificial Intelligence in Triaging Patient Questions: An Evaluation of a Large Language Model for Distal Radius Fractures. Journal of the American Academy of Orthopaedic Surgeons, 2026, 34(1): e106-e115.

[2] Wen Z W, Wang J Y, Yang W Y, et al. Construction and efficacy verification of an intelligent pharmaceutical Q&A platform based on AI hallucination-suppression. China Pharmacy, 2026, 37(2): 226-231.

[3] Yu H, Zu H Y, Wang W D, et al. Research on stroke knowledge graph and intelligent diagnosis and treatment based on Stroke BERT. Computer Technology and Development, 2026, 36(2): 126-131.

[4] Park J, Kim S, Yoon B, et al. M4CXR: Exploring Multitask Potentials of Multimodal Large Language Models for Chest X-Ray Interpretation. IEEE Transactions on Neural Networks and Learning Systems, 2025, 36(10): 17841-17855.

[5] Aljohani M, Ho B, McKinley K, et al. Domain-Adapted Small Language Models for Reliable Clinical Triage. arXiv preprint arXiv:2604.26766, 2026.

[6] Shang F, Xia Y, Yang D, et al. MedRepBench: A Comprehensive Benchmark for Medical Report Interpretation. arXiv preprint arXiv:2508.16674, 2026.

[7] Chahid A, Chahid I, Mrabti W, et al. An AI-Powered Approach for Medical Specialty Triage Using Natural Language Processing and Transformer Models. International Journal of Advanced Computer Science and Applications, 2026, 17(4). doi:10.14569/IJACSA.2026.0170411.

[8] Kopka M, von Kalckreuth N, Feufel M A. Accuracy of online symptom assessment applications, large language models, and laypeople for self-triage decisions. npj Digital Medicine, 2025, 8: 178.