International Journal of Creative and Open Research in Engineering and Management
ISSN: 3108-1754 (Online)
Volume 02 Issue 10 October-2026 | Impact Factor: 3.5



Design and Development of Multi-Agent Healthcare Assistance Framework Using LLM and RAG

Mrs. Pranjali Gopinath Salve, Dr.Archana.S. Vaidya
Department of Computer Engineering
Gokhale Education Society's R. H. Sapat College of Engineering, Management Studies and Research
salvepranjal2000@gmail.com, archana.vaidya@ges-coengg.org

How to Cite this Article:

Salve, P. G. (2026). Design and Development of Multi-Agent Healthcare Assistance Framework Using LLM and RAG. International Journal of Creative and Open Research in Engineering and Management, 02(10), 1-9. https://doi.org/10.55041/ijcope.v2i10.235

License:

This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.







https://doi.org/10.55041/ijcope.v2i10.235

***Abstract*—**Medical clinics field a steady stream of routine questions about reports, appointments, symptoms, and medicines. Answering them quickly and correctly takes staff time that is often scarce, and patients are left waiting or puzzled by medical jargon. This paper reviews how researchers have approached the problem and then describes a multi-agent assistant built on Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG). The literature survey covers more than fifty studies on traditional clinic support, conversational health agents, biomedical language models, medical question answering, RAG, LLM-based agents, evaluation, and safety. Together they show that general-purpose LLMs know a good deal of medicine but still invent facts, that retrieval from trusted sources lowers that risk, and that dividing the work among specialised agents makes an assistant easier to control. Building on these findings, the proposed system places a central orchestrator in front of five agents: medical question answering, medical document analysis, appointment management, symptom checking, and medicine information. Each agent draws on approved clinic knowledge, and every reply passes a safety review before reaching the patient. Quality is to be measured with ROUGE, BLEU, semantic similarity, and LLM-based judging, together with clinician review. The assistant supports routine communication and does not replace professional medical consultation.

Keywords*—Medical clinics, large language models, retrieval-augmented generation, multi-agent AI, medical question answering, medical document analysis, symptom checker, medicine information, appointment management, healthcare AI*

I. INTRODUCTION

A medical clinic is, among other things, an information desk. Patients ask what a laboratory value means, whether a fever needs a visit today, when a particular doctor is free, and whether two medicines can be taken together. Doctors, nurses, and receptionists answer these questions many times a day, usually in between other work. When the queue grows, replies slow down, and patients are left with printed reports they cannot read and instructions they only half remember. Artificial intelligence (AI) has opened a realistic route to easing this load, and the clinical value of language-based AI is now discussed in the leading medical journals [1], [2].

The most visible change came from Large Language Models (LLMs), which can read a free-text question and write a fluent answer. Studies on medical benchmarks show that such models hold a surprising amount of clinical knowledge [3], [4]. The same studies are also clear about the downside: the models sometimes state wrong facts with complete confidence, they cannot see a clinic's own documents, and their knowledge stops at the date of training. Retrieval-Augmented Generation (RAG) was proposed to address exactly this gap by fetching relevant passages from an external source and giving them to the model before it answers [5]. In healthcare, that source can be a set of approved clinic documents rather than the open internet.



A second design question follows. A patient may ask about a report, an appointment, a symptom, or a medicine, and each of these needs different data, different tools, and different safety rules. Recent work on LLM-based agents suggests that it is easier to build and test several specialised components, coordinated by a controller, than one model that does everything [6], [7]. A multi-agent virtual assistant for a medical clinic that combines this idea with RAG has already been reported, and it forms the starting point of this project [8].

This document therefore does two things. Section II is a literature survey that follows the field from traditional clinic support, through language models and medical question answering, to RAG, multi-agent systems, evaluation, and safety. It makes up the larger part of the document, because the design choices of the proposed assistant only make sense against that background. Section III then presents the proposed system: its architecture, the five specialised agents, the RAG workflow, the safety layer, and the evaluation plan. Section IV closes with conclusions and future work. Throughout, the assistant is positioned as support for routine communication. It does not diagnose, prescribe, or replace a clinician.

II. LITERATURE SURVEY

The survey is organised around the questions a designer of such an assistant has to answer. How do clinics handle patient queries today, and where does that break down? What can language models do, and where do they fail? How does retrieval reduce those failures? Why use agents? How should the result be measured, and under what safety rules should it run? Fig. 1 shows how the topics relate.

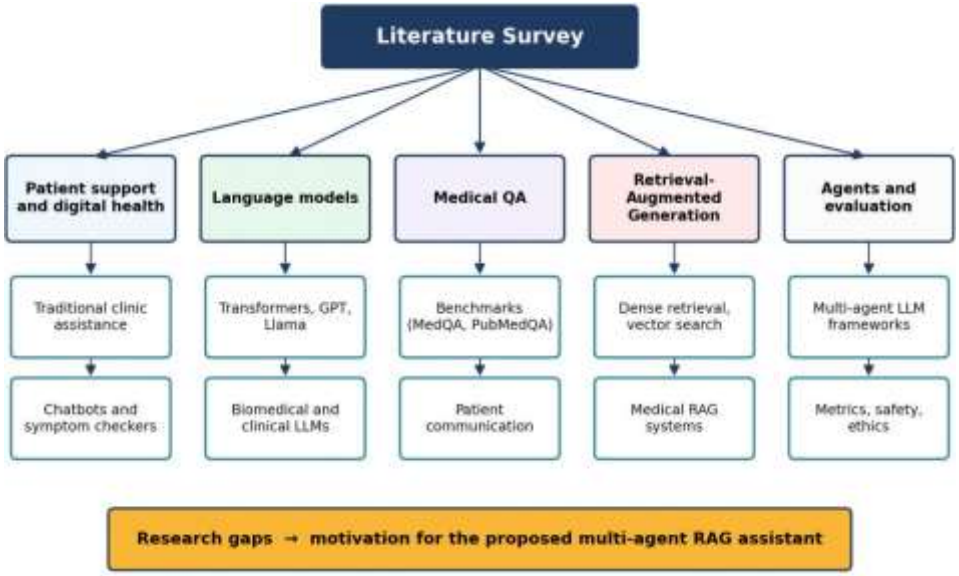


Fig. 1. Organisation of the literature survey.

A. Patient Support in Clinics: Traditional Practice

Clinics have always relied on people to answer patient questions. A patient walks in, phones the front desk, or catches a nurse after a consultation. Reception staff keep appointment books in registers or spreadsheets, and doctors and pharmacists explain reports and prescriptions face to face. These channels work well at modest volumes, and nothing in the literature suggests they should disappear. Their weakness is scale. Telephone lines become congested, the same answer is repeated dozens of times a day, and manual scheduling invites double bookings and transcription errors.

Medical documents add a further layer of difficulty. A laboratory report is written for clinicians, so abbreviations, units, and reference ranges leave many patients unsure whether a result matters. Medicine labels and leaflets are dense, and patients often retain only part of what was said during the consultation. When professional help is not at hand,



people search online. Semigran et al. tested 23 online symptom checkers on 45 clinical vignettes and found that the correct diagnosis was listed first in only 34% of cases and that urgency advice was appropriate in 57% [9]. The figures show how a self-service tool can sound authoritative and still be wrong much of the time. Topol argues that AI in medicine is most useful when it supports clinicians and patients instead of working in place of them [2], and this survey takes the same view.

B. Conversational Agents and Symptom Checkers

Software that talks to patients is not new. Laranjo et al. reviewed conversational agents in healthcare and found that most systems relied on predefined dialogue flows, that user input was often restricted to menus or short phrases, and that evidence of clinical effectiveness was still thin [10]. Rule-based designs are predictable, which is a virtue in medicine, but they break as soon as a patient phrases a question in a way the designer did not foresee.

General-purpose voice assistants show the opposite problem. Bickmore et al. posed medical scenarios to Siri, Alexa, and Google Assistant and reported that a notable share of the responses could lead to harm if a consumer acted on them [11]. Together, these two findings frame the engineering problem: a patient-facing assistant has to understand free-form language, as modern language models do, while staying inside boundaries that are as dependable as those of a rule-based system.

C. Foundations: Transformers and Large Language Models

The current generation of language models rests on the Transformer, which replaced recurrence with self-attention and made it practical to train on very large text collections [12]. BERT showed that a model pre-trained to read text in both directions could be fine-tuned for many language tasks with little extra data [13]. GPT-3, with 175 billion parameters, then demonstrated that a sufficiently large model can perform new tasks from a few examples placed in the prompt, without any weight updates [14].

Two further developments matter for assistants. Instruction tuning with human feedback, as in InstructGPT, made models far better at following what a user actually asks; in the authors' evaluation, outputs from a 1.3-billion-parameter instruction-tuned model were preferred over those of the much larger GPT-3 [15]. Chain-of-thought prompting showed that asking a model to write out intermediate reasoning steps improves performance on multi-step problems [16]. Capable proprietary and open models have followed, including GPT-4 [17] and the openly released Llama 2 family, which spans 7 to 70 billion parameters [18]. The availability of open weights is relevant for clinics that cannot send patient data to an external service.

D. Biomedical and Clinical Language Models

Medical text has its own vocabulary, so researchers adapted these architectures to it. BioBERT continued BERT's pre-training on PubMed abstracts and PMC full-text articles and improved results on biomedical named-entity recognition, relation extraction, and question answering [19]. Clinical BERT variants were trained on hospital notes from the MIMIC-III database [20]. PubMedBERT went a step further and trained from scratch on biomedical text, arguing that in-domain vocabulary and data beat starting from a general model . BioGPT brought the same idea to generation, pre-training a generative Transformer on millions of PubMed abstracts .

A second wave targeted dialogue. ChatDoctor fine-tuned LLaMA on roughly 100,000 real patient–doctor conversations and added a module that retrieves information from external sources during the chat . HuatuoGPT combined distilled ChatGPT data with real doctors' replies to make answers more informative and more doctor-like. PMC-LLaMA continued the pre-training of LLaMA on biomedical papers and textbooks before instruction tuning , and ClinicalGPT used medical records, domain question answering, and multi-turn dialogues, together with its own evaluation framework . Table I summarises these models.



TABLE I. REPRESENTATIVE BIOMEDICAL AND CLINICAL LANGUAGE MODELS

Model	Training data / approach	Main contribution	Limitation relevant here
BioBERT	BERT further pre-trained on PubMed and PMC text	Better biomedical NER, relation extraction, QA	Encoder only; does not generate answers
Clinical BERT	Pre-trained on MIMIC-III clinical notes	Embeddings suited to clinical language	Notes from one hospital system
PubMedBERT	Trained from scratch on PubMed	Shows value of in-domain vocabulary	Encoder only
BioGPT	Generative Transformer on PubMed abstracts	Biomedical text generation and mining	Knowledge fixed at training time
ChatDoctor	LLaMA tuned on real consultations, plus retrieval	Conversational medical advice	Consultation data may be noisy
HuatuoGPT	Distilled and real doctor responses	Informative, doctor-like replies	Mainly Chinese-language setting
PMC-LLaMA	LLaMA pre-trained on papers and books	Open medical foundation model	No guarantee of factual grounding
ClinicalGPT	Records, QA, and dialogue data	Includes an evaluation framework	Static knowledge

Two observations follow from this body of work. Domain adaptation makes the output read like medicine, but it does not give the model access to a particular clinic's timetables, reports, or protocols. And whatever the model learned is frozen on the day training stopped. Both points push toward architectures in which knowledge lives outside the model and can be updated without retraining.

E. Large Language Models for Medical Question Answering

Progress in this area is usually measured on question sets drawn from exams and the research literature. MedQA collects questions from medical licensing examinations ; PubMedQA asks yes/no/maybe questions about research abstracts ; and MedMCQA offers a very large multi-subject set built from Indian medical entrance examinations . Singhal et al. combined these and other sets into MultiMedQA and added a human evaluation framework covering factuality, comprehension, reasoning, possible harm, and bias. Their instruction-tuned Flan-PaLM reached 67.6% on MedQA-style questions, but human raters found notable gaps in its long-form answers, which led to Med-PaLM [3]. Its successor, Med-PaLM 2, scored about 86% on the same type of questions, and physicians preferred its answers to those of other physicians on most of the axes examined .

General models are strong as well. Nori et al. reported that GPT-4, with no medical specialisation, exceeded the USMLE passing mark by more than 20 points [4]. Liévin et al. examined reasoning on medical questions with GPT-3.5, concluding that step-by-step prompting and retrieval helped, although the generated reasoning could still be flawed .

What matters most for a clinic is how these abilities translate to patient communication. Lee, Bubeck, and Petro described both the promise and the hazards of GPT-4 as a medical chatbot, including its tendency to make errors that are hard to spot . In a widely cited study, Ayers et al. compared physician replies and chatbot replies to 195 patient questions from a public forum; evaluators preferred the chatbot reply in 78.6% of cases and rated it higher on quality and empathy . Mondal et al. looked at LLM use for primary healthcare issues, with remote care and medical education in mind . Broader reviews by Thirunavukarasu et al. and Tian et al. list the same recurring concerns: factual errors, bias, privacy, and the absence of regulatory clarity [1],

A caution is in order. Scoring well on an exam is not the same as being safe with a worried patient. Benchmarks reward the right option on a closed list, whereas real queries are open, incomplete, and sometimes emotional. This is



why the survey now turns to hallucination and grounding.

F. Hallucination and the Need for Grounding

Ji et al. surveyed hallucination in text generation and separated two kinds: content that contradicts the source material and content that cannot be verified from it at all . In a medical setting both are dangerous, since an invented dosage or a made-up reference range reads exactly like a correct one. Fine-tuning reduces the problem without removing it, as the human evaluations behind Med-PaLM show [3]. A more dependable approach is to make the model answer from supplied evidence and to be able to point to it. That is the idea behind retrieval-augmented generation.

The risk is not only invented facts. A model can also give an answer that was correct when its training ended but is out of date today, such as a revised guideline or a withdrawn medicine, and it can phrase a wrong answer with the same fluent confidence as a right one . Both behaviours are hard for a lay reader to detect, which is why the grounding step has to be built into the system and not left to the patient.

G. Retrieval-Augmented Generation

Lewis et al. introduced RAG as a combination of a parametric generator and a non-parametric memory, a dense index of Wikipedia searched with a neural retriever. Their model set new results on several open-domain question-answering tasks and produced answers that were more specific and more factual than a generator working alone [5]. REALM had earlier trained a retriever jointly with a language model so that retrieval became part of pre-training . Dense Passage Retrieval showed that a learned two-encoder retriever could beat the classic BM25 method by 9–19 points of top-20 accuracy .

Practical RAG systems need two supporting pieces. The first is an embedding model that maps text to vectors whose distance reflects meaning; Sentence-BERT made this efficient by producing sentence embeddings that can be compared with cosine similarity . The second is a vector search engine. FAISS provides fast nearest-neighbour search that scales to billions of vectors . Gao et al. organised the growing variety of designs into three paradigms: naive RAG (index, retrieve, generate), advanced RAG (with query rewriting and re-ranking), and modular RAG (with interchangeable and iterative components). Self-RAG added a further step in which the model decides when to retrieve and critiques its own output using special reflection tokens .

Medical applications of RAG have grown quickly. Almanac, a retrieval-augmented system for clinical medicine, was judged by clinicians to be more factual and safer than standard LLM answers on a set of clinical questions . Xiong et al. built MIRAGE, a benchmark of several thousand medical questions, and the MedRAG toolkit; across their experiments retrieval improved accuracy by up to 18% over chain-of-thought prompting, and it helped smaller models approach the performance of larger ones . Sohn et al. proposed rationale-guided retrieval, in which the model's own reasoning is used to filter and steer the retrieved evidence, with gains over earlier medical RAG baselines . A systematic review by Amugongo et al. concluded that RAG is promising in healthcare but that evaluation practice, ethical safeguards, and real-world deployment are still underdeveloped . Table II compares these studies.

TABLE II. SELECTED STUDIES ON RETRIEVAL-AUGMENTED GENERATION IN MEDICINE

Study	Retrieval source / idea	Reported outcome	Lesson for this project
Lewis et al.	Dense Wikipedia index with a neural retriever	More specific, factual answers than generator alone	Separating knowledge from the model works
Almanac	Curated medical sources	Clinicians rated answers more factual and safer	Curated sources matter
MIRAGE / MedRAG	Several corpora and retrievers	Up to 18% accuracy gain over CoT	Choose corpus and retriever deliberately
RAG with rationales	Reasoning-guided query and filtering	Gains over earlier medical RAG baselines	Filter retrieved passages before generation
Systematic review	Review of healthcare RAG studies	Evaluation and ethics under-reported	Plan evaluation and safety from the start



H. Design Choices in Retrieval Pipelines

Beyond the headline idea, the quality of a RAG system depends on many small decisions. Chunk size is one. Passages that are too short lose their context, while very long ones bury the relevant sentence in noise. The retrieval method is another. Sparse methods such as BM25 match exact terms and remain strong for drug names and codes , whereas dense retrievers match meaning . Combining the two, usually called hybrid retrieval, is a common way to get the benefits of both, and Gao et al. list query rewriting and re-ranking among the standard additions of advanced pipelines .

A third decision is how many passages to put in the prompt. Liu et al. found that language models use information best when it sits at the start or end of a long input and often miss material placed in the middle
. That finding favours a small number of well-ranked passages over a large dump of text. Finally, measuring a RAG system is not the same as measuring a language model. Frameworks such as RAGAs score retrieval and generation separately, using reference-free measures such as the faithfulness of the answer to the retrieved context and the relevance of the answer to the question . These choices carry over directly to a clinic, where a missed drug name or an answer built on an irrelevant passage has more serious consequences than in open-domain question answering.

I. LLM-Based Agents and Multi-Agent Systems

An agent, in this literature, is an LLM that can do more than talk: it plans, calls tools, and observes the results. ReAct interleaves reasoning steps with actions such as search calls, which reduced hallucination on question-answering tasks compared with reasoning alone . Toolformer showed that a model can learn when and how to call external APIs such as a calculator or search engine . Generative agents added memory and reflection, so that simulated characters could act consistently over time .

When several agents work together, new issues appear: who decides which agent acts, how information is passed on, and how mistakes are caught. AutoGen offers a framework in which customisable agents converse with each other and with humans to complete tasks [6]. MetaGPT assigns agents human-like roles and standard operating procedures, which reduces confusion between them . Two surveys map the field. Wang et al. cover how agents are constructed and applied , and Guo et al. discuss communication structures and open challenges of multi-agent systems [7]. In the medical domain, MedAgents lets role-playing LLM “experts” from different specialties discuss a case and reach a consensus, improving zero-shot medical reasoning . Table III lists the frameworks and what each contributes to a clinic assistant.

TABLE III. AGENT FRAMEWORKS AND THEIR RELEVANCE TO A CLINIC ASSISTANT

Framework	Core mechanism	Relevance to the proposed assistant
ReAct	Alternates reasoning and tool actions	Each agent can look up evidence before answering
Toolformer	Model learns to call external APIs	Supports scheduling and drug-database calls
Generative agents	Memory, reflection, planning	Session memory for multi-turn conversations
AutoGen	Conversable agents with human-in-the-loop	Orchestration pattern and escalation to staff
MetaGPT	Role assignment with standard procedures	Clear, narrow responsibilities per agent
MedAgents	Expert-role discussion for medical cases	Shows multi-agent reasoning helps in medicine

J. Multi-Agent RAG Assistants for Medical Clinics

The study closest to this project is the one by Tanasă, Oprea, and Bâra, who designed and evaluated an agentic virtual assistant for a medical clinic. A central agent routes each query to specialised tools for retrieval-based question answering, document interpretation, and appointment scheduling. The system was built with LangChain and LangGraph, used Gemini 2.5 Flash, and relied on strict prompt rules to keep responses reliable and limit hallucination. Evaluation with ROUGE, BLEU, LLM-as-a-judge scoring, sentiment analysis, and SBERT-based similarity indicated



that the multi-agent design improved interpretability and response quality, and that clinic-specific RAG improved grounding and coherence [8]. The present project follows this orchestrator pattern and extends it with a symptom-checking agent, a medicine-information agent, and an explicit safety-review stage.

K. Literature on the Five Clinic Services

The five services of the proposed assistant are not equally well covered in the literature, and the amount of evidence available shapes how cautious each agent has to be.

Medical question answering is the best studied. Benchmarks, human evaluations, and the patient-forum comparison discussed earlier all concern this task [3], and retrieval-based variants have been shown to improve it .

Document analysis is less mature. The building blocks exist: biomedical encoders can pull out entities such as test names, units, and conditions from text, and general LLMs can rewrite technical language in simpler words. The clinic assistant reported in [8] includes document interpretation as one of its tools. What is still missing is a systematic evaluation of how accurately a patient-facing system explains laboratory values. That gap is the reason the Document Analysis Agent in Section III relies on fixed reference-range tables for its comparisons instead of the model's memory.

Appointment management is largely an engineering problem of connecting a language model to a scheduling system. The tool-use work described above shows how models can call external interfaces , and the clinic study uses the same idea for booking [8]. The main risk is not medical accuracy but acting on the wrong patient's record, so authentication and a confirmation step matter more here than clever language.

Symptom checking has the longest history and the most cautionary evidence. The audit by Semigran et al. showed modest diagnostic and triage accuracy for online tools [9], and Bickmore et al. documented potentially harmful answers from voice assistants [11]. LLMs read free text more flexibly, but the reviews cited earlier warn about overconfidence [1]. A defensible design therefore limits the output to general information and an urgency level, and it checks for red-flag wording before anything else.

Medicine information sits between question answering and safety-critical advice. Retrieval from drug references can supply general statements about purpose and precautions , but dosage and prescribing decisions belong to a clinician, so the assistant should explain and refer rather than instruct. Across all five services, none of the studies reviewed here evaluates them together in one patient-facing system with a shared safety layer, which is the gap the proposed system targets.

L. Evaluation of Generated Medical Responses

No single metric captures whether a medical answer is good. BLEU counts n-gram overlap with a reference and was designed for machine translation . ROUGE is recall-oriented and common in summarisation . Both reward word overlap, so a correct answer phrased differently can score poorly. BERTScore and SBERT-style cosine similarity compare contextual embeddings and tolerate paraphrase better . LLM-as-a-judge uses a strong model to grade answers against a rubric; Zheng et al. found that GPT-4 judgments agreed with human preferences more than 80% of the time, similar to the agreement between humans, while also documenting biases such as favouring longer answers . For medicine, Singhal et al. argued that automated scores should be combined with human review of factuality, harm, and bias [3]. Table IV summarises the trade-offs.

TABLE IV. COMMON EVALUATION METHODS FOR GENERATED ANSWERS

Method	What it measures	Strength	Limitation
BLEU	n-gram precision against a reference	Cheap and reproducible	Penalises valid paraphrase
ROUGE	n-gram / sequence recall	Widely reported	Surface overlap only
BERTScore / SBERT	Embedding similarity to reference	Tolerates rewording	Cannot detect subtle factual errors
LLM-as-a-judge	Rubric-based grading by an LLM	Scales; close to human preference	Length and self-preference bias



Expert review [3]	Factuality, harm, bias	Most trustworthy	Costly and slow

M. Safety, Privacy, and Regulation

The World Health Organization's guidance on AI for health lists six principles: protect human autonomy; promote well-being, safety, and the public interest; ensure transparency and explainability; foster responsibility and accountability; ensure inclusiveness and equity; and promote responsive and sustainable AI. Several of these translate directly into design requirements: citing sources supports transparency, escalating to staff supports accountability, and offering plain language supports inclusiveness.

Regulators are catching up. Gilbert et al. argued that LLM chatbots that give medical advice should be treated as medical devices and should therefore go through approval, while Meskó and Topol made the case for regulatory oversight of generative AI in healthcare more broadly . For a clinic assistant, the practical consequence is to keep the scope narrow. The assistant should provide general information and administrative help, say clearly that it is not a diagnostic tool, and hand over to a professional whenever a question goes beyond that scope. Privacy deserves equal attention: patients upload reports and describe symptoms, so identifiers should be masked before text is sent to an external model, documents should be stored encrypted, and access should be logged.

N. Comparison of Assistance Approaches

Drawing the earlier sections together, Table V compares the main ways a clinic can handle patient queries. The ratings are qualitative judgments synthesised from the studies cited above and the working of each method; they are not measurements from a single experiment.

TABLE V. QUALITATIVE COMPARISON OF CLINIC ASSISTANCE METHODS

Method	Response capability	Cost	Accuracy	Scalability	Availability
Face-to-face consultation	High, but limited by doctor time	High	High (professional)	Low	Medium, needs a visit
Telephone assistance	Medium, basic queries	Low–Medium	Medium	Medium	Clinic hours
Manual appointment management	Medium	Low	Medium–High, depends on data entry	Low–Medium	Working hours
Manual report explanation	High	Medium–H igh	High (professional)	Low	Depends on staff
General online search	Medium	Very low	Low–Medium, often unspecific	High	Always
Single LLM assistant [32], [33]	High	Low–Medium	Medium, may hallucinate [36]	High	Always
RAG-based assistant [5], [44]	High	Medium	High, grounded in sources	High	Always
Proposed multi-agent RAG assistant	Very high, one agent per task	Medium	High, with guardrails and review	Very high	Always, with escalation to staff

O. Research Gaps and Challenges

Reading across the literature, five gaps stand out. First, most medical LLM studies evaluate on exam-style questions, while few examine the routine administrative and explanatory queries that fill a clinic's day [3]. Second, RAG research concentrates on question answering; document interpretation, scheduling, and medicine information, which need tools as well as retrieval, receive less attention [8], . Third, multi-agent frameworks are mostly demonstrated on



software and reasoning tasks, with few designs tailored to patient-facing safety [6], [7]. Fourth, evaluation tends to lean on a single metric type, although overlap metrics, embedding similarity, LLM judges, and human review each miss different errors . Fifth, privacy, regulation, and escalation paths are discussed in policy papers but are rarely built into the architecture .

Practical challenges accompany these gaps. The knowledge base has to be kept current, because stale clinic information produces stale answers. Elderly or low-literacy users need simple language and an easy interface. Symptom checking carries inherent risk, since a symptom can have many causes and some of them are urgent [9]. The proposed system, described next, is designed with these gaps in mind.

Looking ahead, the literature points in three directions. Retrieval pipelines are becoming modular, with components for rewriting, re-ranking, and self-critique that can be swapped as needs change. Agent frameworks are moving toward clearer roles, shared protocols, and a human in the loop [6], [7]. And evaluation is shifting from single scores to combined automatic and expert assessment [3]. A clinic assistant built today can adopt all three trends without waiting for new models.

III. PROPOSED SYSTEM

A. Objectives and Scope

The proposed system is a multi-agent virtual assistant for a medical clinic. It is meant to answer routine patient questions quickly, in plain language, and from information the clinic has approved. Five services are in scope, shown in Fig. 2: medical question answering, medical document analysis, appointment management, symptom checking, and medicine information. All five rest on two shared mechanisms, RAG for grounding and an orchestrator for routing, and all five pass through response evaluation and safety checks. Four principles guide the design, drawn from the survey: answers come from retrieved, approved evidence [5], each task has its own narrowly scoped agent [8], ; every reply is checked before it is shown and the assistant always defers to clinicians for diagnosis, treatment, and emergencies .

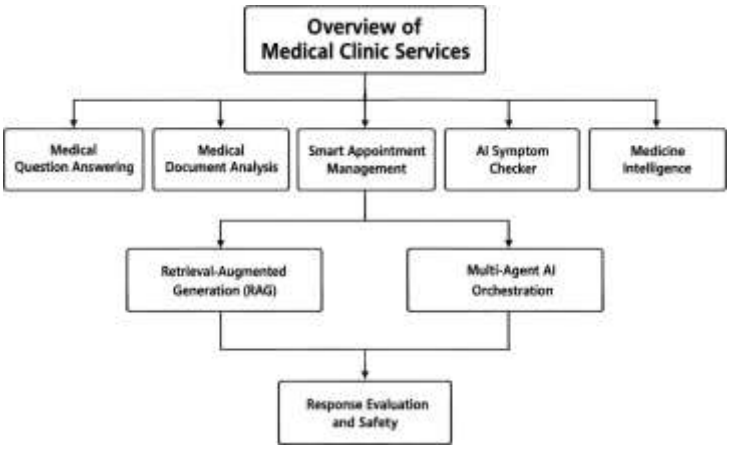


Fig. 2. Services offered by the proposed clinic assistant and the shared mechanisms behind them.

B. System Architecture

The architecture has four layers (Fig. 3). The presentation layer offers a web or mobile chat for patients, a document-upload option, and a dashboard for clinic staff. The application and safety layer authenticates users, masks personal identifiers, screens for emergency wording, and later applies output guardrails. The intelligence layer holds the orchestrator and the five agents. The knowledge and data layer contains the vector database with its embedding model, the approved knowledge base, the scheduling database, an encrypted store for uploaded documents, the LLM service, and the evaluation and audit logs. Keeping these layers separate lets the clinic replace the language model, the vector store, or the user interface without redesigning the rest.

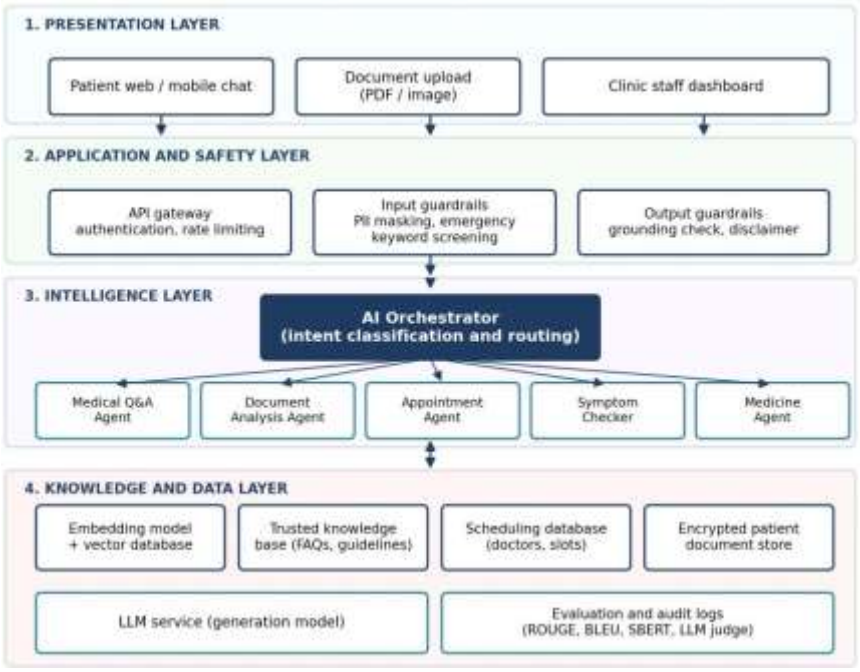


Fig. 3. Layered system architecture of the proposed assistant.

C. Multi-Agent Architecture

The orchestrator is the entry point to the intelligence layer (Fig. 4). It reads the patient's message, classifies the intent, and passes the request to the matching agent. If a message mixes intents, for example "Can I move my appointment, and is this tablet safe with my fever?", it splits the message and calls two agents in turn. It also keeps short-term session memory so that follow-up questions make sense, and it assembles the final reply. This pattern follows the orchestrator-and-tools design reported for clinic assistants [8] and the conversable-agent idea of AutoGen [6]. Agents never talk to the patient directly; every draft goes to a safety reviewer first.

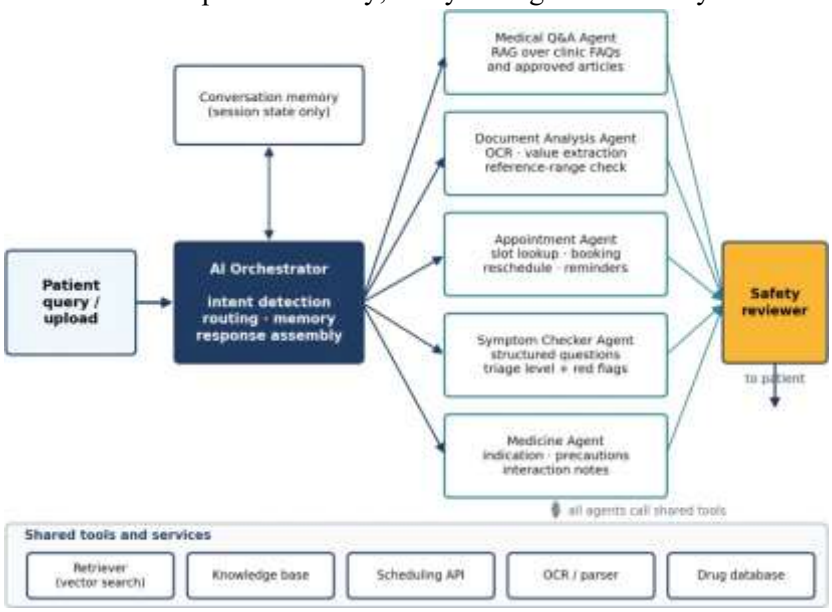


Fig. 4. Multi-agent architecture with a central orchestrator, five specialised agents, and a safety reviewer.

Table VI lists the responsibilities and limits of each agent. The limits are as important as the duties: they are written into each agent's prompt and checked again by the safety reviewer.



TABLE VI. RESPONSIBILITIES AND BOUNDARIES OF THE SPECIALISED AGENTS

Agent	Inputs and tools	What it does	What it must not do
Medical Q&A	Text question; retriever over FAQs and approved articles	Answers general clinic and health questions with source tags	Give answers not supported by retrieved text
Document Analysis	Uploaded report; OCR and parser; reference-range table	Extracts values, flags out-of-range items, explains terms simply	Interpret results as a diagnosis
Appointment	Scheduling API; doctor and slot database	Shows availability, books, reschedules, sends reminders	Reveal other patients' data
Symptom Checker	Structured follow-up questions; red-flag rules; retriever	Collects symptoms, gives general information and an urgency level	Name a final diagnosis
Medicine	Drug-information base; retriever	Explains purpose, common precautions, general interactions	Advise starting, stopping, or changing a dose

D. RAG Workflow

The RAG workflow has an offline and an online half (Fig. 5). Offline, approved sources such as clinic FAQs, service descriptions, patient-education leaflets, and drug information are parsed, cleaned, and split into chunks of roughly 300 to 500 tokens with a small overlap, so that sentences are not cut from their context. Each chunk is converted to a vector by a sentence-embedding model and stored in a vector index with metadata such as source, section, and last-reviewed date . Online, the patient's question is rewritten if needed, embedded, and matched against the index; the top five chunks are re-ranked, and those below a relevance threshold are dropped . The remaining text is placed into a prompt that tells the model to answer only from the evidence and to tag each statement with its source.

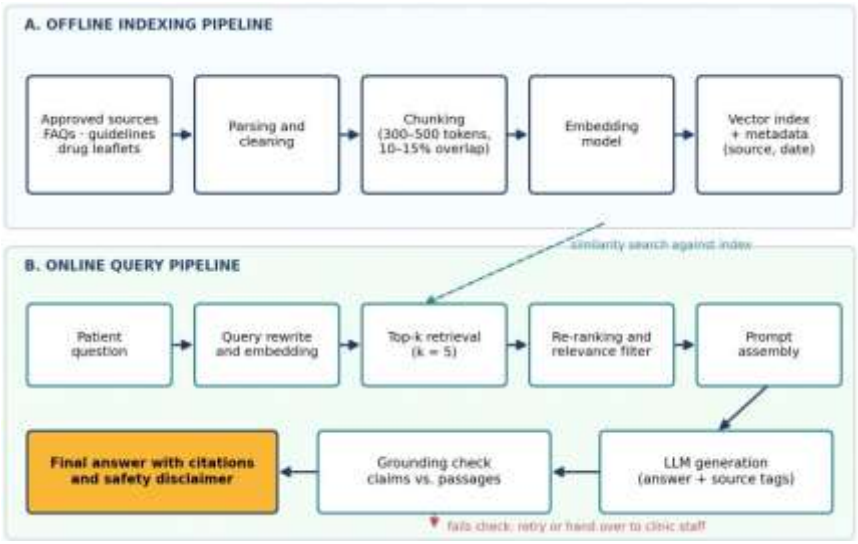


Fig. 5. RAG workflow: offline indexing pipeline and online query pipeline.

A grounding check then compares the draft with the retrieved passages. If a statement has no support, the draft is regenerated once; if that fails too, or if retrieval returns nothing relevant, the assistant says it cannot answer and offers to pass the question to clinic staff. The idea of letting the system critique its own output before answering is borrowed from Self-RAG , though here the check is implemented as a separate step rather than inside the model. The numeric settings (chunk size, overlap, top-k, threshold) are starting values to be tuned on the clinic's own documents.

E. Safety and Privacy Guardrails

Fig. 6 follows one message through the safety path. On the way in, personal identifiers such as names and phone numbers are masked before any text reaches an external model, and an emergency check looks for wording such as chest pain or difficulty breathing; if it fires, the assistant skips generation and shows local emergency contact instructions straight away. On the way out, the safety reviewer verifies that the draft stays inside the agent's boundaries from Table VI, contains no dosage advice or diagnosis, and carries a short reminder that the information is general and does not replace a clinician. Low-confidence cases are handed to staff together with the conversation, so the patient does not have to repeat themselves. Uploaded documents are encrypted at rest, access is role-based, and every request is logged for audit, in line with the accountability and transparency principles discussed in Section II-M.

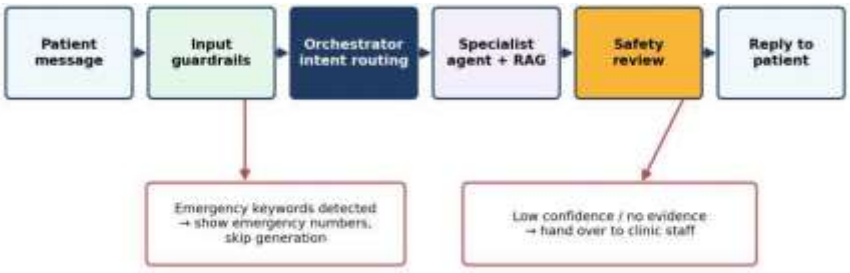


Fig. 6. Request flow with emergency and low-confidence escalation paths.

F. Evaluation Plan

The evaluation follows the lesson of Section II-L that no single metric is enough. A test set will be prepared from the clinic's own material, with a reference answer for each question and a fixed number of questions per agent. The proposed assistant will be compared with two baselines: a single LLM without retrieval, and a single LLM with RAG but without agents. Table VII lists what will be measured.

TABLE VII. PLANNED EVALUATION OF THE PROPOSED ASSISTANT

Aspect	Measure	Purpose
Routing	Intent-classification accuracy of the orchestrator	Checks that queries reach the right agent
Retrieval	Recall@k of relevant chunks	Checks that evidence is found
Answer overlap	ROUGE and BLEU against references	Comparability with earlier work [8]
Semantic quality	SBERT cosine similarity / BERTScore	Credits correct paraphrases
Overall quality	LLM-as-a-judge on relevance, completeness, groundedness	Scalable rubric-based scoring
Safety	Clinician review of a sample: factuality, harm, disclaimers [3]	Catches errors automatic metrics miss
Usability	Response time and a short patient and staff survey	Confirms practical benefit

G. Expected Outcomes and Limitations

Based on the evidence reviewed, the multi-agent RAG design is expected to beat a single LLM on groundedness and relevance, and to match or improve on a single RAG pipeline in task handling, because each agent has a narrower job [8]. It should also shorten waiting times for routine queries and reduce repeated work for staff. These are expectations rather than results; they will be tested with the plan above, and numerical findings will be reported only after the



experiments are run.

Several limitations remain. Answer quality is bounded by the quality and freshness of the knowledge base. The symptom checker can only give general guidance and cannot examine the patient [9]. Using several agents adds latency and cost compared with a single call. And any LLM can still make mistakes that a grounding check fails to catch, which is why the human fallback stays in place.

IV. CONCLUSION AND FUTURE WORK

This document surveyed the research behind AI-assisted clinic communication and described a multi-agent assistant built on that foundation. The literature agrees on a few points. Language models know a lot of medicine but cannot be trusted to recall it faultlessly retrieval from approved sources makes answers more factual specialised agents with a coordinating controller make complex assistants easier to build and test and no system of this kind should operate without safety review, privacy protection, and a path to a human The proposed assistant puts these points together: an orchestrator, five agents, a grounded RAG pipeline, layered guardrails, and a mixed automatic and expert evaluation plan.

Future work includes a working prototype and the evaluation described , followed by personalisation based on authorised patient history, voice and image input for easier use, direct integration with the clinic's scheduling software, and support for regional languages so that more patients can use the assistant comfortably. Throughout, the goal stays the same: to give patients quicker, clearer answers to routine questions and to free clinicians' time for the care that only they can provide.

REFERENCES

[1] A. J. Thirunavukarasu et al., "Large language models in medicine," *Nature Medicine*, vol. 29, no. 8, pp. 1930–1940, Aug. 2023.

[2] E. J. Topol, "High-performance medicine: The convergence of human and artificial intelligence," *Nature Medicine*, vol. 25, no. 1, pp. 44–56, Jan. 2019.

[3] K. Singhal et al., "Large language models encode clinical knowledge," *Nature*, vol. 620, pp. 172–180, Jul. 2023, doi: 10.1038/s41586-023-06291-2.

[4] H. Nori, N. King, S. M. McKinney, D. Carignan, and E. Horvitz, "Capabilities of GPT-4 on medical challenge problems," arXiv preprint arXiv:2303.13375, 2023.

[5] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. NeurIPS*, vol. 33, 2020, pp. 9459–9474.

[6] Q. Wu et al., "AutoGen: Enabling next-gen LLM applications via multi-agent conversation," arXiv preprint arXiv:2308.08155, 2023.

[7] T. Guo et al., "Large language model based multi-agents: A survey of progress and challenges," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2024, pp. 8048–8057.

[8] A.-M. Tanasă, S.-V. Oprea, and A. Bâra, "Designing an architecture of a multi-agentic AI-powered virtual assistant using LLMs and RAG for a medical clinic," *Electronics*, vol. 15, no. 2, p. 334, Jan. 2026, doi: 10.3390/electronics15020334.

[9] H. L. Semigran, J. A. Linder, C. Gidengil, and A. Mehrotra, "Evaluation of symptom checkers for self diagnosis and triage: Audit study," *BMJ*, vol. 351, p. h3480, 2015.

[10] L. Laranjo et al., "Conversational agents in healthcare: A systematic review," *J. Am. Med. Inform. Assoc.*, vol. 25, no. 9, pp. 1248–1258, Sep. 2018.

[11] T. W. Bickmore et al., "Patient and consumer safety risks when using conversational assistants for medical information: An observational study of Siri, Alexa, and Google Assistant," *J. Med. Internet Res.*, vol. 20, no. 9, p. e11510, 2018.

[12] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017,



pp. 5998–6008.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[14] T. B. Brown et al., “Language models are few-shot learners,” in *Proc. NeurIPS*, vol. 33, 2020, pp. 1877–1901.

[15] L. Ouyang et al., “Training language models to follow instructions with human feedback,” in *Proc. NeurIPS*, vol. 35, 2022, pp. 27730–27744.

[16] J. Wei et al., “Chain-of-thought prompting elicits reasoning in large language models,” in *Proc. NeurIPS*, vol. 35, 2022, pp. 24824–24837.

[17] OpenAI, “GPT-4 technical report,” arXiv preprint arXiv:2303.08774, 2023.

[18] H. Touvron et al., “Llama 2: Open foundation and fine-tuned chat models,” arXiv preprint arXiv:2307.09288, 2023.

[19] J. Lee et al., “BioBERT: A pre-trained biomedical language representation model for biomedical text mining,” *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, Feb. 2020.

[20] E. Alsentzer et al., “Publicly available clinical BERT embeddings,” in *Proc. 2nd Clin. Nat. Lang. Process. Workshop*, 2019, pp. 72–78.