Medical Report Intelligence Assistant (MRIA) for Automated Clinical Documentation and Assistance

Aadithya S[1], Dr. L. Sivagami[2], Mrs. C. Bella Stary Gold[3]

[1]*Student, Department of Applied Electronics, Sriram Engineering College, Tamil Nadu, India*

[2]*School Head, Department of Applied Electronics, Sriram Engineering College, Tamil Nadu, India*

[3]*Assistant Professor, Department of Applied Electronics, Sriram Engineering College, Tamil Nadu, India*

[1]*adithyaadhi042@gmail.com*, [2]sivagamil.ece@sriramec.edu.in

***Abstract*— Medical Report Intelligence Assistant is an artificial intelligence assisted healthcare documentation system designed to convert doctor patient conversations into structured medical reports. In conventional clinical practice, doctors spend significant time after consultations recording symptoms, diagnoses, prescriptions, and follow up notes. This manual process increases workload and may result in incomplete or inconsistent medical records. The proposed system captures consultation audio through an embedded Raspberry Pi based device, converts speech into text using speech recognition techniques, and applies natural language processing methods to extract medically relevant information. Patient details, symptoms, diagnoses, medications, dosage information, investigations, and medical advice are organized into a structured format and stored in a database. Historical records are analyzed using vector based similarity search to provide contextual information from previous consultations. A multi agent workflow employing advanced language models and automation tools generates a comprehensive medical report after review and approval by the physician. The system improves documentation efficiency, reduces repetitive data entry, supports digital patient history management, and provides a scalable foundation for intelligent clinical workflow automation.**

Keywords*— Artificial Intelligence, Clinical Documentation, Speech Recognition, Natural Language Processing, Medical Report Generation, Healthcare Automation, Electronic Health Records, Clinical Workflow Management.

I. Introduction

Healthcare documentation is a vital part of medical practice because every consultation must be recorded for diagnosis, prescription, treatment planning and future reference. However, in many hospitals and clinics, report writing still depends on handwritten notes or manual data entry. Doctors are therefore required to divide their attention between patient interaction and documentation, which can reduce consultation efficiency during busy outpatient and emergency conditions.

The Medical Report Intelligence Assistant (MRIA) addresses this challenge by automating the conversion of doctor-patient conversations into structured medical reports. The system captures speech through a digital microphone connected to a Raspberry Pi-based hardware unit. The audio stream is converted into text, processed by Natural Language Processing algorithms, validated into medical fields and finally presented as a readable report.

MRIA is not designed to replace clinical judgement. Instead, it acts as an intelligent digital assistant that reduces repetitive writing, improves consistency of records and helps doctors maintain a clear history of patient information. By combining embedded edge computing, AI agents, vector search and automated report generation, the system supports faster and more organized healthcare documentation.

II. NEED FOR THE STUDY

The need for this study arises from the practical difficulties faced in clinical documentation. Manual report preparation consumes time, depends on typing speed and may miss important details such as symptom duration, medicine dosage, allergy history or follow-up advice. In high-patient-load environments, even small documentation delays can affect workflow and patient waiting time.

Medical data also needs to be stored in a structured and searchable form. When past reports are available only as handwritten notes or unorganized text, retrieving useful patient history becomes difficult. MRIA improves this situation by creating digital records that can be stored, searched and reused during future consultations.

Another important requirement is the availability of a system that can work close to the source of data. The use of a Raspberry Pi platform allows audio capture and local preprocessing at the edge. This reduces dependence on large desktop systems and makes the solution suitable for clinics, telemedicine setups and rural healthcare centers.

The project therefore focuses on building a reliable, affordable and scalable assistant that can capture consultation speech, extract medical meaning, generate reports and support the doctor with similar case references while keeping the final decision under human control.

 ISSN: 3107-6513

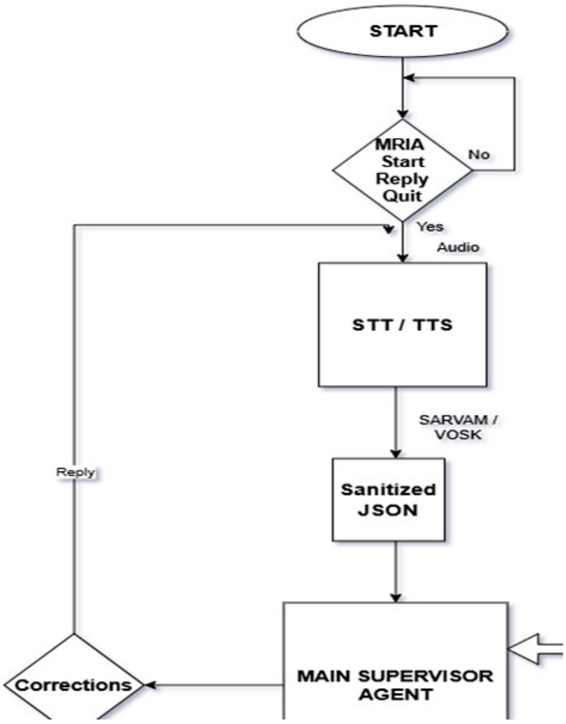

Fig. 1. Working flow of the MRIA system.

A. Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

III. PROPOSED SYSTEM

The proposed MRIA system is organized as a complete pipeline from voice input to final report generation. The consultation begins when the doctor starts the MRIA reply cycle. The audio is captured through the microphone and passed to the Speech-to-Text and Text-to-Speech layer. Sarvam and Vosk models are used for speech conversion depending on connectivity and language requirements.

The transcribed output is cleaned and converted into sanitized JSON. This removes irrelevant words, normalizes clinical expressions and prepares the content for the main supervisor agent. The supervisor agent coordinates the extraction of patient demographics, symptoms, observations, medicines, diagnostic suggestions and follow-up instructions.

For clinical context support, the supervisor agent communicates with a differential diagnosis block that uses ColQwen and Qdrant for retrieval. Patient demographic and historical information is stored and fetched from MongoDB. After validation and correction, the report agent uses Playwright to generate a structured HTML/PDF medical report.

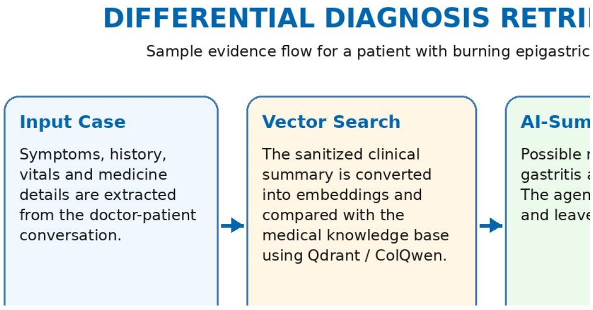

Fig. 2. MRIA differential diagnosis retrieval layer.

A. System Architecture:

The architecture contains four major layers: hardware acquisition, speech processing, AI interpretation and report output. The hardware acquisition layer includes Raspberry Pi, MEMS microphone, RTC and audio amplifier. The speech processing layer converts voice into machine-readable text. The AI interpretation layer extracts medical entities and retrieves similar cases. The output layer generates the final report and displays it for doctor approval.

B. Hardware Prototype:

Compact custom PCB is used to organize the embedded components needed for MRIA operation. The board includes power input sections, USB-C connectivity, Raspberry Pi Compute Module support, audio interface lines, speaker output and headers for development or expansion.

The top and bottom 3D PCB views are shown in Fig. 2 and Fig. 3. The images are placed inside the column layout to maintain continuity of the IEEE-style format.

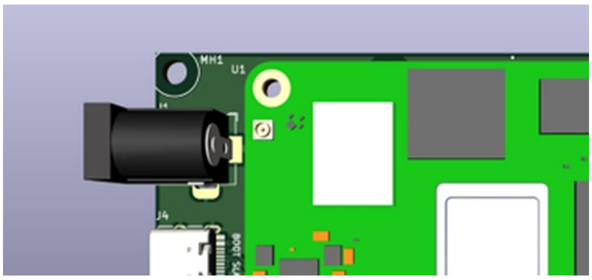

Fig. 3. Top 3D PCB view of the MRIA hardware prototype.

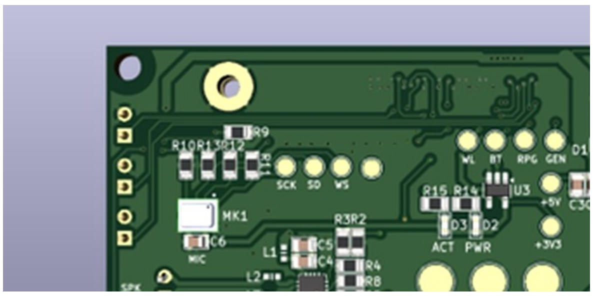

Fig. 4. Bottom 3D PCB view of the MRIA hardware prototype.

IV. RESEARCH METHODOLOGY

The development of MRIA follows a structured methodology that connects the hardware, software, AI workflow and report generation modules. Each stage is designed to reduce manual documentation effort while maintaining clinical accuracy and user verification.

A. Problem Identification and Requirement Analysis:

The existing documentation process was studied to identify major issues such as repetitive typing, unstructured notes, difficulty in retrieving history and lack of standard report format. Based on this analysis, the functional requirements were defined as voice capture, transcription, data extraction, database storage, similar case retrieval and PDF generation.

B. Problem Identification and Requirement Analysis:

The audio input subsystem was planned around a Raspberry Pi-based edge device with a digital MEMS microphone. The design also includes RTC support for accurate timestamping and audio output hardware for feedback when required. This provides a portable platform suitable for clinical environments.

C. Speech Recognition Processing:

The captured audio stream is cleaned and segmented before conversion into text. Sarvam speech recognition and Vosk are considered for handling online and offline conversion needs. The output of this stage becomes the raw medical conversation text.

D. NLP-Based Medical Extraction:

The transcribed text is processed by AI and NLP modules to extract patient identity, symptoms, duration, diagnosis, medicines, dosage, tests and advice. The system uses a supervisor-agent design so that uncertain fields can be checked and corrected before report generation.

E. Database and Vector Retrieval:

MongoDB stores structured patient information, while Qdrant stores vector embeddings of previous cases and reference knowledge. When a new case is processed, the vector search module retrieves similar records and supports context-aware differential diagnosis.

F. Report Generation and Validation:

After the data is structured, the report agent prepares a standardized medical report in HTML and PDF format. The doctor can review the content, apply corrections and approve the final report before storage or printing.

G. Testing and Evaluation:

The system is evaluated using sample consultation scenarios. Testing focuses on transcription clarity, field extraction accuracy, response time, report completeness, data storage reliability and the readability of the generated output.

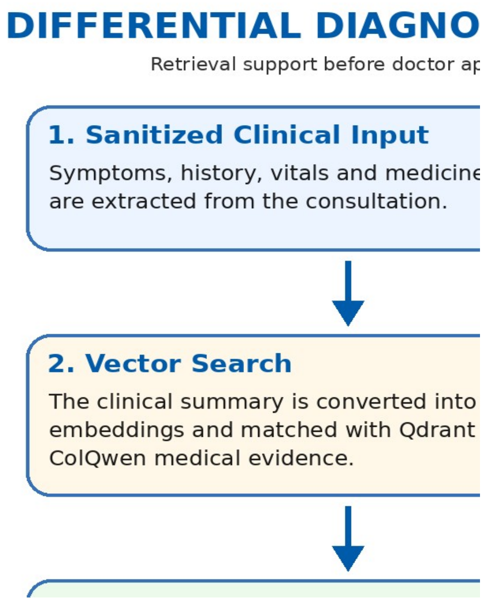

Fig. 5. Column-friendly differential diagnosis retrieval layer.

V. OBJECTIVE

The main objective of MRIA is to develop an intelligent healthcare documentation assistant that can generate structured medical reports from natural doctor-patient conversations. The specific objectives are listed below:

- To capture real-time consultation audio using a compact embedded hardware device.
- To convert doctor-patient speech into accurate text using speech recognition.
- To extract symptoms, diagnosis, medicines, dosage, investigations and advice using NLP.
- To store patient records digitally in a structured and retrievable format.
- To use vector search for retrieving similar previous cases and medical reference context.
- To generate standardized HTML and PDF reports through an automated report agent.
- To reduce doctor workload, documentation time and human error in report writing.
- To provide a scalable system that can be extended to multilingual and hospital-integrated versions.

VI. MODULE DESCRIPTION

The MRIA system is divided into several functional modules so that every stage of clinical documentation can be developed, tested, maintained and improved independently. This modular design makes the system more flexible and reliable because each module performs a specific task in the complete workflow. From capturing the doctor-patient conversation to generating the final medical report, every module contributes to reducing manual documentation effort and improving the accuracy of clinical records. The major modules of the MRIA system are described below.

- *Voice Input Module:*

The Voice Input Module is responsible for capturing the doctor-patient conversation during the consultation. It uses a MEMS microphone connected to the Raspberry Pi hardware unit to receive audio input clearly. This module continuously listens to the consultation and records the spoken interaction without disturbing the normal workflow of the doctor. The quality of this module is important because the accuracy of the later transcription and report generation depends on the clarity of the captured audio.

- *Speech Recognition Module:*

The Speech Recognition Module converts the captured audio into text format. It uses Sarvam, Vosk and real-time WebSocket streaming to process the audio and generate live transcription. This module helps in converting spoken medical discussions into machine-readable text. Real-time streaming allows the system to process the conversation while the consultation is happening, instead of waiting until the entire session is completed. This improves speed and makes the system suitable for practical clinical use.

- *NLP Processing Module:*

The NLP Processing Module is used to understand the clinical meaning of the transcribed text. It uses Gemini, LangGraph and specialized extraction agents to identify important medical entities from the conversation. These entities may include symptoms, complaints, duration of illness, history, medication details, diagnosis and advice. This module does not simply read the text as plain sentences; instead, it tries to understand the context of the conversation and separate medically relevant information from casual or unnecessary speech.

- *Data Structuring Module:*

The Data Structuring Module converts the extracted clinical information into a clean and sanitized JSON format. This structured format contains important fields such as patient demographics, chief complaints, symptoms, history, diagnosis, prescription and treatment advice. By organizing the data in JSON format, the system makes it easier to store, retrieve, validate and use the information for report generation. This module also helps in removing unwanted filler words, repeated phrases and incomplete information from the raw transcription output.

- *Database Storage Module*:

The Database Storage Module stores patient records extracted clinical data and generated report information in MongoDB. This helps in maintaining patient history and allows doctors to access previous consultation details whenever required. Storing the data in a database also supports continuity of care, because old records can be referred to during future visits. MongoDB is suitable for this purpose because it can handle flexible document-based medical data, which may vary from patient to patient.

- *Vector Retrieval Module:*

The Vector Retrieval Module uses Qdrant and ColQwen to retrieve relevant previous cases or reference knowledge. This module supports the doctor by finding similar medical cases, related symptoms or useful reference information based on the current consultation data. The retrieved information can be useful for evidence-based differential diagnosis support. However, the output of this module is only used as supporting information and not as an automatic final diagnosis. The doctor reviews and validates the information before making any clinical decision.

- *Report Generation Module:*

The Report Generation Module converts the structured medical data into a professional medical report. It uses Playwright along with HTML and PDF templates to generate a clean, readable and doctor-verifiable report. The report includes important sections such as patient details, complaints, history, examination findings, investigations, diagnosis, prescription and advice. After verification by the doctor, the report can be exported as a PDF and stored for future use. This module ensures that the final output is properly formatted and suitable for clinical documentation.

VII. WORKING OF MRIA

The working of MRIA begins when the doctor starts the consultation session with the patient. During the consultation, the embedded device continuously captures the doctor-patient conversation through the audio input system. The captured audio stream is passed to the speech recognition layer for further processing. Before transcription, the raw audio is pre-processed to improve clarity and accuracy. This pre-processing stage helps in reducing silence, background noise, repeated pauses and other unwanted audio disturbances that may affect the quality of the converted text.

After the audio is cleaned, the Speech-to-Text (STT) module converts the spoken conversation into textual form. Since medical conversations may contain filler words, repeated statements, incomplete sentences and casual speech patterns, the generated text is further sanitized. This sanitization process removes unnecessary filler words and organizes the meaningful content into a stable JSON-like structure. This structured format makes it easier for the system to identify important clinical information from the conversation.

The sanitized text is then received by the main supervisor agent. The supervisor agent plays an important role in understanding the extracted conversation and identifying the required medical fields. It separates the information into relevant categories such as patient complaints, symptoms, duration of illness, previous medical history, social history, investigations and treatment-related information. If any extracted detail is incomplete, unclear or wrongly interpreted, the correction loop allows the doctor to review, modify or confirm the extracted data before it is used for report generation. This ensures that the final medical record is not created only based on machine interpretation but is verified by the doctor.

The supervisor agent also communicates with MongoDB to obtain patient-related information such as demographics, previous visits, past medical history and other stored records. This helps the system to maintain continuity in patient care and avoid repeated manual entry of already available data. In addition to MongoDB, the system queries Qdrant and ColQwen to retrieve similar medical cases, reference knowledge or clinically relevant supporting information. The retrieved information is used only as an assisting reference for the doctor. It is not treated as a final diagnosis or automatic decision. The doctor remains the final authority in validating the medical findings and deciding the treatment plan.

Once the doctor verifies and approves the extracted content, the report agent converts the structured medical data into a clean and readable medical report. The report is generated in a standard format and contains important sections such as patient identification, chief complaint, history of present illness, history, social history, review of systems, investigations, evidence-based differential diagnosis, treatment plan and medical advice. The system arranges the information in a professional manner so that the report can be easily understood by doctors, patients and future healthcare providers.

Finally, after the report is reviewed and confirmed, it is exported as a PDF document. The generated PDF can be stored in the database for future reference and can also be shared or printed when required. By combining embedded audio capture, speech recognition, structured data extraction, clinical knowledge retrieval and doctor verification, MRIA reduces manual documentation work and improves the speed, consistency and quality of medical report generation.

VIII. ADVANTAGES

- Reduces repetitive documentation work for doctors.
- Improves consistency and structure of medical reports.
- Supports faster retrieval of patient history and similar cases.
- Can operate with edge hardware for compact deployment.
- Create PDF reports suitable for storing, printing and sharing.
- Allows doctor verification before final report approval.

IX. LIMITATIONS

- Speech recognition accuracy may be reduced in noisy hospital environments.
- Mixed-language conversations and strong accents may require additional training.
- AI-generated extraction must always be verified by a qualified medical professional.
- Internet-based AI modules may require stable connectivity unless offline models are configured.
- Clinical safety depends on proper validation, authentication and privacy handling.

X. FUTURE SCOPE

Future versions of MRIA can include multilingual support for Tamil and other Indian regional languages. This will make the system more useful in rural and semi-urban healthcare centers where patients may not communicate fully in English.

The system can also be integrated with hospital management platforms so that prescriptions, lab requests, billing and patient history are connected automatically. Additional improvements may include doctor’s voice adaptation, better medical vocabulary correction, digital signature support, QR code verification and secure patient mobile access.

Advanced clinical validation modules can be added to detect missing dosage, abnormal vital values, drug interactions or incomplete follow-up advice. With these improvements, MRIA can develop into a complete AI-powered documentation and clinical decision-support platform.

XI. CONCLUSION

The Medical Report Intelligence Assistant provides a practical solution for automating medical report generation from doctor-patient conversations. By integrating speech recognition, Natural Language Processing, vector search, database storage and PDF generation, the system reduces manual documentation effort and improves the organization of patient records.

MRIA supports doctors by creating structured reports that can be reviewed, corrected and approved before final storage. The system also improves the availability of patient history and provides similar case retrieval as supporting context. The embedded hardware design makes the solution suitable for clinics, hospitals, telemedicine units and rural healthcare centers.

Overall, MRIA represents a step toward intelligent healthcare documentation. It does not replace doctors, but it reduces their clerical workload and allows them to focus more on diagnosis, treatment and patient care.

REFERENCES

[1] Y. Bazi, M. M. Al Rahhal, L. Bashmal, and M. Zuair, “Vision-language model for visual question answering in medical imagery,” *Bioengineering*, vol. 10, no. 3, Art. no. 380, 2023, doi: 10.3390/bioengineering10030380.

[2] I. Hartsock and G. Rasool, “Vision-language models for medical report generation and visual question answering: A review,” *Frontiers in Artificial Intelligence*, vol. 7, Art. no. 1430984, 2024, doi: 10.3389/frai.2024.1430984.

[3] A. E. W. Johnson, T. J. Pollard, N. R. Greenbaum, M. P. Lungren, C. Deng, Y. Peng, Z. Lu, R. G. Mark, S. J. Berkowitz, and S. Horng, “MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports,” *Scientific Data*, vol. 6, Art. no. 317, 2019, doi: 10.1038/s41597-019-0322-0.

[4] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, “BioBERT: A pre-trained biomedical language representation model for biomedical text mining,” *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020, doi: 10.1093/bioinformatics/btz682.

[5] J. J. W. Ng, E. Wang, X. Zhou, K. X. Zhou, C. X. L. Goh, G. Z. N. Sim, H. K. Tan, S. S. N. Goh, and Q. X. Ng, “Evaluating the performance of artificial intelligence-based speech recognition for clinical

documentation: A systematic review," *BMC Medical Informatics and Decision Making*, vol. 25, Art. no. 236, 2025, doi: 10.1186/s12911-025-03061-0.

[6] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning*, vol. 202, pp. 28492–28518, 2023.

[7] M. M. van Buchem, H. Boosman, M. P. Bauer, I. M. J. Kant, S. A. Cammel, and E. W. Steyerberg, "The digital scribe in clinical practice: A scoping review and research agenda," *npj Digital Medicine*, vol. 4, Art. no. 57, 2021, doi: 10.1038/s41746-021-00432-5.