19ENG234 – Project Phase 1

Topic:

Medical Scribe

Description:

Enabling Automated Population of Patient-Clinician Dialogue into EHR systems utilising Large Language Models

Done by:

K. Venkata Naga Satya Subhash – AIE21036
amenu4aie21036@am.students.amrita.edu

Thati Ayyappa Swamy – AIE21084
amenu4aie21084@am.students.amrita.edu

K. Sumitha – AIE21040
amenu4aie21040@am.students.amrita.edu

B. Indra Kiran – AIE21078
amenu4aie21078@am.students.amrita.edu

Artificial Intelligence Engineering

Amrita School of Computing

Amrita Vishwa Vidyapeetham, Amritapuri.

Professor/Guide:

Dr. Rahul Krishnan Pathinarupothi

Assistant Professor (Sl. Gd.), Amrita Center for Wireless Networks & Applications (AmritaWNA), Amritapuri

Background:

According to numerous studies, physicians and other healthcare professionals spend around 2-4 hours per shift on manual documentation tasks, which can take 30% of their total work time. This investment of time takes away from direct patient care and increases overall workload, especially during consultations. Research also indicates that data entered manually has an error rate of 10-15%, while automated systems are below 1%. These errors can lead to serious consequences: one study identified that 25% of medical errors are linked to incorrect documentation. For example, to retrieve and interpret a set of paper records compared to only a few minutes using digital systems. They are concerned that adopting a structured and standardized electronic health record (EHR) will lead to more time documenting.

Implementation:

This project aims to streamline this process by recording, transcribing, and automatically summarizing patient-clinician interactions using AI. Integrating these transcriptions, summaries and audio files directly into the Electronic Health Record (EHR) system database. Also, patients can schedule appointments.

Objective:

The primary objective of this project is to significantly reduce the time and effort clinicians spend on documentation by >50%, allowing them to focus more on patient care. By automating the transcription and summarization of patient-clinician conversations through large language models, the project aims to minimize human errors that often occur in manual documentation. This ensures more consistent records.

Intended Audience:

This software is designed specifically for the healthcare providers, particularly doctors, within hospital settings. The application functions in person, allowing doctors to record patient appointments directly during consultations. It is for hospital environments, providing clinicians with an efficient tool to capture, transcribe, and summarize patient interactions.

Functional Requirements:

Authentication Module:

- **Sign-In and Sign-Up:** Secure sign-in and sign-up pages for both clients and doctors.
- **Two-Factor Authentication (2FA):** Added 2FA for password recovery, ensuring enhanced security.
- **Google Sign-in:** If the application is built using fire base and google cloud we sign in using google account.
- **Password Hashing:** Implementing password hashing for improved security.

User Roles and Permissions:

- **Doctor:** Ability to start/stop recordings, review automatically generated summaries and transcriptions, and update, export or save data to the EHR system. Respond for the patient's appointment requests.
- **Patient:** Access personal health records, including summaries of their consultations. Schedule appointments. Get Notifications from the Doctor regarding Scheduled appointments.

Voice Recording Module:

- Start/Stop/Pause recording functionality.
- Check the status of audio recording (failed/successful).
- Secure storage of recorded audio.

Transcription Module:

- Automatic transcription of recorded audio into text.
- We can use whisper/assembly AI / google cloud speech to text model for transcription.
- Display of real-time or near-real-time transcription.
- The transcription enters the patient EHR displays on the patient ID/visit ID.

Summarization Module:

- Automated generation of conversation summaries using a trained language model.
- For summarization we can use GTP-4o/ llama 3.1-8B-fine-tuned/ medical LLM for summarization running on local machine or API calls.
- The generated summary will display on patient ID.
- Option for doctors to update, review and approve the summary before exporting.

Database:

- We are using MySQL for our EMR database to store/create user data - user records. doctor records, user activity, appointments.
- And MangoDB/VectorDB to store and retrieve audio files, transcription and summaries.

Data Export/Integration:

- Export summaries and transcriptions in formats compatible with EHR systems.
- Integration with existing EHR systems via APIs.
- The system should save patient records in a dedicated folder for each patient every time they visit the doctor.
- If the system requires API integration with external EHR systems, consider adding a block or connection in the architecture that explicitly shows "API Integration" for data export and import.

User Interface:

- Intuitive UI for easy navigation and operation.

- **Doctor Dashboard:** shows recent and upcoming appointments, recordings, summaries, and patient details with medical history.
- **Patient Dashboard:** shows recent and upcoming appointments, recordings, summaries and doctor details with specialization.

Non-Functional Requirements:

Performance:

- The primary bottleneck in our application lies with the AI models for speech-to-text and summarization. We can run these models in two ways: on GPUs or cloud infrastructure, or through external APIs.
- Using APIs can reduce the demand for resources, allowing for CPU-based processing. However, it can be costly, even for deployment.
- Based on the model used for transcription, working system of the model and the system resources available for the transcription and summarization.

Security:

- HIPAA compliance for data handling and storage. A federal law that protects privacy and security of sensitive patient data.
- Encryption of stored and transmitted data.

Scalability:

- Ability to handle multiple users and sessions concurrently.
- Scalable architecture to accommodate future growth.
- By deploying local models in the cloud (e.g., AWS or Google Cloud), the system can leverage cloud resources to scale up as needed.
- Utilizing APIs enables flexibility.

Reliability:

- Ensure system uptime and availability. Make sure to less crashes.
- Regular backups and recovery plans.
- Ensure consistent performance of all modules, including voice recording, transcription, summarization, and integration with the EHR system.
- Implement error handling mechanisms to manage and log issues during recording, transcription, and summarization, ensuring minimal disruption to users.

Usability:

- User-friendly interface with minimal training required.
- Accessibility features for differently-abled users.

Technical Requirements:

Technology Stack:

- **Backend:** FastAPI

- **Frontend:** JavaScript, React and FIGMA.
- **APIs:** Integration with third-party services for transcription and summarization:
  - **Transcription:** Whisper model, Assembly AI, or Google Cloud Speech-to-Text
  - **Summarization:** ChatGPT-GPT-4, Llama 3 8B model, or similar LLM's
- **Database:**
  - **Relational:** MySQL for user and appointment data
  - **Non-relational:** MongoDB/VectorDB for audio files, transcriptions, summaries
- **Tools:** GitHub, Docker, Postman

Risks and Mitigation:

Data Privacy Risks:

- Mitigation strategies like encryption and secure authentication.

Technical Challenges:

- Potential issues with transcription accuracy, summarization quality, and model training.
- If the conversation between doctor and patient is done in a distant room the model might have the inability to capture all the conversation for the transcription.
- Limitations in handling low-volume speech, background noise, microphone issues, or unclear speech.

Model Limitations:

- Risks related to the summarization model's ability to capture nuisance/irrelative terms in the conversation.
- If the LLM is not fine-tuned to the medical data the model might have the inability to capture few medical terminologies.

Conclusion:

This AI-supported solution helps clinicians save over 50% of the time they spend on documentation, enabling them to focus more on patient care. Automating transcription and summarization of patient interactions ensures accuracy and efficiency, while also facilitating integration with EHR systems for guaranteed results. Security, scalability (including HIPAA compliance) are the main focus in the system's design, with a strong emphasis on user-friendly interfaces for both doctors and patients. This project focuses on a comprehensive method that addresses both transcription accuracy problems and data privacy issues, preparing it for future healthcare documentation challenges.

Reference:

https://www.carepatron.com/blog/how-many-hours-are-you-spending-writing-medical-notes-manually

https://lemongrad.com/conversation-between-doctor-and-patient/