DocAI

Senior Project by Riley Badnin and Justin Brunings
California Polytechnic State University, San Luis Obispo
Computer Science & Software Engineering Department
Advised by Dr. Zoë Wood
Abstract

DocAI presents a user-friendly platform for recording, transcribing, summarizing, and classifying doctor-patient consultations. The application utilizes AssemblyAI for conversational transcription, and the user interface allows users to either live-record consultations or upload an existing MP3 file. The classification process, powered by 'ml-classify-text,' organizes the consultation transcription into SOAP (Subjective, Objective, Assessment, and Plan) format – a widely used method of documentation for healthcare providers. The result of this development is a simple yet effective interface that effectively plays the role of a medical scribe. However, the application is still facing challenges of inconsistent summarization from the AssemblyAI backend. Future work involves refining summarization consistency and exploring integration with medical platforms.

Introduction

In the healthcare industry, there’s little room for missed details and inaccuracy in patient care. Following the implementation of the Health Information Technology for Economic and Clinical Health (HITECH) Act, many hospitals have drastically increased their use of electronic health records (EHRs) [4]. These often come in the form of systems that can be outdated and cumbersome with added regulatory requirements. Although EHRs have increased the quantity and consistency of medical documentation, they add a significant amount of administrative burden [2]. This burden is large – in the United States, provider notes are nearly four times longer than in other countries [6].

Nowadays, hospitals attempt to handle these issues through simple dictation software or hiring medical scribes; however, both of these methods have their downsides. Dictation software can be inaccurate, and they still require hospital employees to sort through the transcription of the notes to standardize the note format and pull out essential information. Although medical scribes are an effective approach to this problem, they are often expensive and less accurate. Aspiring medical professionals often fill scribe positions, and their knowledge is less expansive than that of a doctor. Additionally, they are paid relatively minimally and can be prone to burnout. These factors can be a catalyst for human error being introduced to EHRs, which can have detrimental effects on patient care [2].
When COVID-19 put most of the world in lockdown in early 2020, most industries had their basic workflows changed. In the medical industry, tele-scribing increased in popularity [2]. The tele-scribe added efficiency in note-taking and gathering quantifiable patient information but still lacked accuracy due to the physical separation between the scribe and the patient. Due to the newfound flexibility in post-COVID-19 workflows that came about from the divergence from traditional methods and practices, there is room for the development of new EHR technology that draws on the benefits of all existing solutions while minimizing preexisting shortcomings. Artificial intelligence can add a more nuanced approach to scribing while maintaining the accuracy that a software approach can provide, thus motivating the development of an AI-powered medical scribe – like DocAI.

Previous & Related Work

Implementing artificial intelligence into an environment as delicate as healthcare can bring up new concerns and a natural reluctance for an industry to be unsure about a new approach to an existing workflow. Thus, it is crucial to understand how receptive doctors are to an AI medical scribe and their preferences for its features.

A survey was conducted with twelve doctors gauging their perspectives on implementing an AI medical scribe. Participants were first asked about their prior experience with medical scribes and dictation software. Results showed varied experience levels, with one using a medical scribe, two employing dictation software, two utilizing both, and six having no experience with either. The doctors additionally rated their interest in an AI-powered medical scribe, with the majority expressing moderate to high interest. The survey also asked doctors about the perceived benefits of an AI scribe, which revealed that time savings and reduced administrative burden were prominent answers. Other desired features included voice recognition, real-time clinical decision support, and integration with existing EHR systems. There was also a unanimous preference for a voice-activated system with the ability to adjust results manually. Overall, the survey indicates doctors' receptiveness to AI integration in medical documentation, contingent on the presence of specific conditions and features.
Since there is a clear interest in a better solution to the current issues in medical documentation, other companies have begun work on and released similar products. A big name in the technology industry pursuing a similar idea is Microsoft, which partnered with voice-recognition company Nuance. They have recently released DAX, an “AI-powered voice-enabled solution that automatically documents patient encounters immediately after the patient visit” [3]. As a result of Microsoft and Nuance’s resources, DAX has been developed to integrate directly with hospitals’ existing EHR systems – a big time saver for doctors. Other small companies, like Augmedix [5] and Abridge [1], are developing similar products. Augmedix, Abridge, and DAX all utilize generative AI to power their applications, similar to ChatGPT. Although Microsoft and Nuance are the big names in this race toward an AI medical scribe, no company has yet achieved a system that doesn’t require humans to check its accuracy [5].

Many of these technologies, notably DAX, were recently released and are still actively under development, leaving room for DocAI to pursue a similar goal. Although DocAI offers a very similar set of features to other solutions, it takes an alternative approach. DAX, Augmedix, and Abridge are all pursuing an AI medical scribe that no longer needs a human to check its accuracy. DocAI is built on the fundamental idea that humans should not be removed from the process in order to integrate feelings, ideas, and perspectives that AI cannot offer on its own. DocAI is intended to be a true assistant to doctors – helping them do their job more efficiently and consistently while not taking away any jobs and keeping the human touch in the process – the touch that many patients value in their healthcare.
Algorithm

Overview

DocAI offers an attractive and intuitive user interface, with both live recording and MP3 uploads. DocAI can be broken down into just a handful of key steps: recording, transcription, summarization, and classification.

Diagram 1. Backend logic flow

AssemblyAI, alongside two custom libraries developed by MIT, makes up the vast majority of the logic behind the user interface. All were carefully compared to competing services, and each was chosen for its superiority in categories such as relevance, ease of use, implementation, and scalability. These design choices will be discussed further in the reflections portion of this paper.
Main Algorithm Details

The recording step in the backend logic flow makes use of a library developed by MIT called ‘mic-recorder-to-mp3’. This library makes the integration of audio input and mp3 file output simple and scalable. Upon loading, a MicRecorder object is created and set to a running state when the ‘record’ button is pressed. This object will stay in the running state until the ‘stop’ button is pressed, after which the contents within the object’s buffer are saved into an mp3 file.

Transcription and summarization are both handled by AssemblyAI, with a series of POST and GET HTTP requests. First, the mp3 file is uploaded with a POST request from which a URL is returned. This URL is then used in a second POST request, which transcribes and summarizes the uploaded file. This request is customizable via the data parameter passed in. Since DocAI is built for the summarization of conversations between two or more speakers, it makes use of AssemblyAI’s conversational summarization mode, along with a formatting type called ‘bullets verbose’—the transcribed audio file is summarized into bullet points, with no limit placed on the number of bullet points created. From here, a GET request collects the output of the previous POST request, and all data is stored for classification.

The final and most complex portion of DocAI’s solution is classification. This portion makes use of a library created by a developer working in research with MIT called ‘ml-classify-text’. This library allows for a model to classify and categorize text, using n-grams and cosine similarity to make predictions. The model was trained on a dataset of over 250 mock doctor/patient consultations. Each file was manually run through the program and each bullet point in the summary was correctly categorized into training data. The trained model is run on the summarized transcript returned by AssemblyAI and builds up variables to store paragraphs for each of the four SOAP categories. Finally, a second model trained only to categorize numerical data found in the ‘objective’ category of the final SOAP output (blood pressure, weight, height, etc.) is run on the original transcript to pick up any crucial data that may have been omitted from the summary.
From here, the logic behind DocAI is complete, and the finalized data is ready for display on the user interface. This entire process is visualized in Diagram 1 above. The frontend of DocAI is a React app that utilizes the NextUI component library, custom styling supported by TailwindCSS, and animations developed using the Framer Motion library. All of these tools were chosen for their high level of customizability, straightforward syntax, and ease of integration with backend tools. The combination of a precise backend workflow and clean frontend design produced a user-friendly and intuitive prototype of DocAI.

Results

After six months of planning, developing, and refining, there is a strong functional prototype of DocAI. It can take in live audio input or MP3 uploads using the MIT-developed library, “mic-recorder-to-mp3”. Transcription and summarization, handled by AssemblyAI, efficiently summarize conversations with its conversational summarization mode. The most intricate phase involves text classification using the 'ml-classify-text' library, trained on a dataset of 250 mock doctor/patient consultations, providing accurate categorization for each bullet point in the summary. The finalized data is visualized in a user-friendly and intuitive prototype of DocAI. The frontend workflow is pictured below, with an example of the backend output.

![Figure 1. Landing page for DocAI](image-url)
Simple separation of live recording and file upload allows the doctor to begin recording with a single click or go back and run the AI model on a previous meeting. Doctors need flexibility, as they are constantly on the go, so DocAI makes sure to provide options to go back and view notes again if something were to go wrong during the meeting, but a valid audio file is still saved regardless.

Recording begins only when the ‘start’ button is pressed, ensuring a sense of privacy for the patient. DocAI promises to only listen to the meeting after patient consent is obtained (and collected before entering the room). Patients value their privacy, so DocAI makes sure to respect that and remain clear about its users’ data.
DocAI can be stopped, restarted, or reviewed at any point to ensure proper functionality, scrap unwanted recorded portions, or discard the recording altogether upon the revocation of patient consent. Per the survey results discussed previously, large amounts of time were allocated to maximizing user experience and flexibility.

DocAI offers a simple presentation of the summarized and categorized SOAP notes. Each category is given its own segment and a complete summarization of the notes. In addition, each box has the option to be revised and edited. Any segments of the original summarization that did not match one of the four categories will appear in the ‘Unclassified’ section and can be directly copied and pasted into any of the four boxes. From here, there is the option to export notes or summarize a new meeting. DocAI exports the SOAP notes in text format, hoping to integrate them into various medical platforms in the future.
DocAI still has some limitations. As seen in Figure 6, the summarized output from AssemblyAI is not tailored to a medical environment. Oftentimes, speaker names are thrown out and replaced with ‘Speaker A’. This leads to a cluttered output and includes large amounts of filler words that would not be present on a typical Doctor’s SOAP notes. Despite this shortcoming, DocAI consistently records the subjective, assessment, and plan data, which is arguably more important than the objective data because those numbers are all in the system before being seen by the doctor. Integration with leading medical software platforms would solve this issue and produce a much higher-quality result.

Considering the difficulty of the problem at hand, DocAI offers very strong results and has a clear route to improvement. Meetings are consistently summarized and categorized into subjective, assessment, and plan, with crucial data rarely missed. And, with DocAI’s edit feature, doctors can quickly and easily add any lost details.

**Figure 6. Categorized notes**
Reflections

Selection of Natural Language Processing Library

Various natural language processing libraries were considered for this project. Natural Language Toolkit (NLTK) was not chosen due to its extensive tools that surpassed the project’s requirements. NLTK was slower and lacked active updates, making it less suitable for the goal of developing a potential production application. Genism, although exhibiting quick performance, fell short in providing the necessary functionality, particularly in separating different speakers – a crucial aspect of our project. Another contender, spaCy, while being a strong and free library, was set aside due to its heavily structured outputs, limiting flexibility in extracting information, which is essential for our project’s objectives. Microsoft Azure and Vowel were also considered, but neither one was developed or tailored to conversational summarization as efficiently as we needed. These considerations led to an exploration of an alternative solution better aligned with our specific needs and goals.

AssemblyAI offered seamless integration and simple usage but imposed one large limitation on the final product: inconsistent summarization. AssemblyAI would sometimes use names for speakers but use ‘Speaker A’ and ‘Speaker B’ at other times, which made text classification more challenging. In addition, AssemblyAI would rarely add important objective data to the summary like height, weight, blood pressure, and more. Because of this, DocAI uses a second classification model trained only on numerical data and pulls from the original transcript, losing the summarized flow that AssemblyAI provides.

Personal Reflection

We enjoyed our opportunity to work on this senior project – it gave us a full-stack project to develop our skills and be creative with. We realized quickly that we wouldn’t have the time to develop all aspects of the application from scratch, so we got the opportunity to explore what AI tools were available. We enjoyed learning more about the medical field and connecting with those in it, and we were excited to have the freedom to develop a project of our choice. Overall, throughout this project, we learned about many different tools and technologies, along with much more about the medical field.
Future Work

As DocAI progresses into its next stages, there are key advancements and adjustments that would drastically improve the quality of the application. Alternative summarization options to replace AssemblyAI must be explored for a solution that may be better suited to DocAI’s needs. Developing an in-house model designed explicitly for summarizing doctor-patient meetings would provide a much more tailored backend – an important enhancement to the AI medical scribe’s accuracy and contextual understanding. Additionally, the creation of a dedicated summarization model for data generated by the numerical objective categorization tool would maximize the system’s adaptability and effectiveness. DocAI should also be integrated with medical software like Epic. Efficiency could continue to be approved by shifting the text categorization model to the backend, mitigating the need for repetitive training during server restarts.

Overall, on the backend, the focus should be on completing the backend switch, fine-tuning the summarization models, and prioritizing ongoing UI updates. On the frontend, emphasis should be placed on expanding account and credential management features for enhanced privacy and security, positioning DocAI to be integrated with existing healthcare systems.
References


Code Libraries

Backend

- **AssemblyAI** - Natural language processing and summarization: [www.assemblyai.com/](http://www.assemblyai.com/)
- **ml-classify-text** - Text classification: [github.com/andreekeberg/ml-classify-text-js](https://github.com/andreekeberg/ml-classify-text-js)

Frontend

- **NextUI** - React component library: [nextui.org/](http://nextui.org/)
- **TailwindCSS** - CSS framework: [tailwindcss.com/](http://tailwindcss.com/)
- **Framer Motion** - Animations: [www.framer.com/motion/](http://www.framer.com/motion/)

Survey Questions

Have you ever used a medical scribe or dictation software for documenting patient encounters?
- Medical scribe
- Dictation software
- Both a medical scribe and dictation software
- Neither a medical scribe nor dictation software

On a scale from 1 to 5, how interested are you in the concept of an AI-powered medical scribe?

In your opinion, what are the potential benefits of using an AI medical scribe in your practice or hospital? Please select all that apply:
- Time savings
- Improved accuracy in documentation
- Better patient care
- Reduced administrative burden
- Other
Are there specific medical specialties or areas of practice where you believe an AI medical scribe would be most useful? Please explain.

What features or capabilities would you like to see in an ideal AI medical scribe?

- Voice recognition
- Real-time clinical decision support
- Automated lab result display
- Other

How would you envision the interaction between you and the AI medical scribe?

- Voice-activated
- Manual inputs
- Other

Do you have any concerns or requests regarding data security and patient privacy in the context of using an AI medical scribe?

Would you prefer the AI medical scribe to be a standalone device, a software application, or an integrated feature within your existing Electronic Health Record (EHR) system?

Please share any additional thoughts, concerns, or suggestions you have regarding the use of AI medical scribes in the field of medicine.