Preliminary Design Investigation: IntelliScribe

Shubh M. Savani

Introduction

In a study published by the American Journal of Emergency Medicine, on average, Emergency room (ER) physicians spend 43% of their time with data entry for patients' electronic health records (EHR) – while only spending 28% of their time interacting face-to-face with patients and 12% reviewing tests/observations (Gold 2013). This study highlights a serious problem in the healthcare industry, as many studies and accounts have shown that ER physicians are spending too much time on EHR creation and management – taking valuable time away with patients. Instead of seeing more patients or spending more time reviewing patients' conditions, ER physicians are burdened with the overwhelming documentation required by EHRs. In fact, EHR documentation is one of the main contributors to burnout among ER physicians.

There are many factors that can influence data entry time for patient reports – including server/mainframe responsiveness to typing skills. In addition, the time spent differs by specialty, where internal medicine physicians spend on average 18-22 minutes on EHR reports while sports medicine/rehabilitation physicians spend on average 8-10 minutes on EHR reports per patient. The breakdown of the time spent on EHRs shows that chart review (33%), documentation (24%), and ordering (17%) are the three main culprits in the reason why physicians spend an excessive amount of time on EHR reports – with documentation being the most frustrating for physicians.

Currently, there are two common solutions that physicians are using to reduce the amount of time they spend on EHRs: in-person and virtual medical scribes.

The first solution that most physicians started to implement in their practices and hospitals is an in-person medical scribe that would sit in the office with the doctor and patient and record patient data/history. This notetaking improves patient care quality as it eliminates the need for physicians to physically document the patient's condition and history. However, there are several limitations to in-person medical scribes. One instance is that there is little regulation/training for medical scribes, restricting the number of tasks that a scribe can do – such as prescribing medicine, scheduling X-rays, and recommending additional tests. This still requires physicians to spend an adequate amount of time on filling out patients' EHRs. Furthermore, in-person medical scribes inside the medical room often make patients uncomfortable – leaving them less inclined to share vital information about their health and data. This can lead to incomplete data and inconclusive diagnosis of a patient's condition.

A solution to this privacy issue is virtual medical scribes. These medical scribes would be called in through phone and listen in on the conversation the physician is having with their patients. This perceived sense of privacy has led to more physicians adopting this solution. Nevertheless, the issue with virtual medical scribes is that there is a lack of standard in terms of training and knowledge. In addition, the cost of virtual medical scribes is high for physicians at $50,000/year per scribe. This heavy price is justifiable for large hospital systems; however, smaller hospitals and clinics cannot afford to pay for as many medical scribes – leaving them behind due to their lack of resources.

In the past couple of years, with advances in machine learning (ML) technologies, a new option has arisen that allows physicians to keep the sense of privacy that virtual medical scribes offer without having to compromise with the high cost and lack of training. These new software platforms use natural language processing (NLP) to read and understand a physician's notes and voice commands in the same way that human scribes can. With the press of a button, the software can listen in on the physician's conversations with his/her patients – gathering and analyzing the data on the patient's health history, current symptoms and illnesses, and test results. The algorithm uses the information gathered to create EHRs and patient documents – saving physicians hours from manual data entry and clerical work and giving them more time to allocate to their patients. These solutions are not only cheaper than a physical/virtual human scribe but also can work around the clock and can be improved through simple software updates.

There are several AI medical scribes in the marker currently; yet they contain several flaws in their development. The first is that these models have not been trained to understand physicians with different accents and slag terms for medical terminology. This flaw could result to an inaccurate diagnosis, harming the patient's health and the physician's reputation as a healthcare provider. To overcome these challenges AI medical scribes, need to be trained on a diverse, more inclusive dataset. In addition, another concern brought up is that it is hard for these software platforms to create the necessary forms/documents for patients with special illnesses and diseases – often requiring physicians to not always use the AI for their patients. Furthermore, AI medical scribes struggle at extracting and summarizing data from the physician-patient conversation, as the conversations are not structured – containing laymen terms, mental thoughts, and disruptions/distractions caused by outside noise.

Solution

IntelliScribe is an AI medical scribe that uses NLP and voice processing technologies to build, document, and order patient EHRs for ER physicians and uses that data generated from the patient to recommend tests, prescribe medicines, and direct patients to an internal medicine physician for special cases. The AI will be adaptable to physicians' and patients' accents by being trained on a diverse set of physician-patient conversations – making the summarization process more seamless and decreasing the word error rate in the NLP model. The user will be able to access the model through a web application. The EHR documents created will be stored in a database.

The NLP model will be implemented in the Python programming language. There will be two main platforms used to implement this model: Natural Language Toolkit (NLTK) and MetaMap. The NLTK suite of libraries are commonly used for NLP in Python and will primarily be used to clean up the raw data and identifying key words in the physician-patient conversation to collect patient information. The MetaMap application will be used for mapping the key words from the physician-patient conversation to the UMLS Metathesaurus. The common uses for

MetaMap will be text summarization, text understanding, and UMLS concept-based indexing and retrieval.

Procedure

Since this project requires extensive time and resources, making it difficult to complete in one semester, the objective of this design project will be to create the main feature of creating the NLP algorithm that can read and analyze a simple physician-patient conversation and create an EHR document. The recommendation system will be implemented after this semester. The conversations will be structured and the patient will not have any special cases or special illnesses that require additional documentation. In addition, the model will not be trained on a diverse set of physicians and patients, as that requires a sizeable amount of search and recruiting for physicians and patients to collect data. In addition, a basic web app and database will be created for the user to interact with the AI and to store the patient data, respectively.

The design project is split into four different phases that will be completed by the end of the semester. As of now, the project will start on 01/30/23 and end on 05/17/23. These dates can be revised with an advisor.

The first phase of the project will focus on learning the NLTK framework and MetaMap application. This will be accomplished through learning from online resources from Udemy, YouTube, and the official documentation released for the frameworks. These resources will provide the programming knowledge necessary for the project.

Furthermore, the second objective of this phase is learning about EHR documents – more specifically the information physicians collect from their patients to complete HER documents. This will be accomplished by conducting customer interviews with ER physicians and experts in the field. Conducting 10-15 interviews should be enough to understand EHR documents.

Lastly, the third objective is to gather data for training, validation, and testing the NLP set. This will be accomplished by using the MedDialog dataset. This large-scale medical dialogue dataset contains 3.4 million Chinese conversations that cover 172 specialties of diseases and 260,000 English conversations between patients and doctors that cover 96 types of diseases. The first phase will start on 01/23/23 and end on 02/20/23.

The second phase of the project will focus on building, training/validation, and testing the NLP model. The model will be built using the algorithms and NLTK framework libraries learned in the previous phase. To train and optimize the hyperparameters in the model, cross-validation techniques will be used. Ideally, these techniques will optimize the balance between bias and variance error – avoiding both underfitting and overfitting the data. Once the optimal hyperparameters are found, the testing dataset will be passed into the model to get the accuracy scores. The accuracy scores will be measured by evaluating the word error rate of the EHR document generated from model to the actual EHR documents of the patient from the dataset. This phase will start on 02/21/23 and end on 03/31/23.

The third phase of the project will focus on building the user-interface for the project and backend storage system to contain the EHR documents created by the IntelliScribe AI. The basic front-end of the website will be built using HTML/CSS. The web app will take in a paragraph input from the user and passe the input text into the NLP model using either Flask or Django framework. The model will generate the EHR document using a template and store it in a relational SQL database using either MySQL, SQLite, or PostgreSQL. The phpMyAdmin application will be used to interface with the SQL database. The skills required for this phase will be learned in the Introduction to Databases course that I am taking this semester. This phase will start on 04/01/23 and end on 04/30/23.

The final phase of the project will be writing the project report based on the requirements set by ABET and NYU Tandon's ECE department. This phase will start on 05/01/23 and end on 05/10/23.

Data Collection and Results

As stated previously, the MedDialog dataset be used for the training, validation, and testing stages of the NLP model. The dataset will undergo a 70:20:10 percentage split, respectively.

The metric that is going to be used for determining the accuracy of the model is to be finalized after phase 1, when learning about the different NLP algorithms and techniques. Through some research, the potential metrics that might be used are BLEU, ROGUE, or perplexity.

My Background

Since high school, I have been interested in healthcare technologies. During my third-year of high school, I was a research intern at the Feinstein Institute for Medical Research in the Center for Bioelectronic Medicine – focused on developing microelectrodes for Vagus Nerve Stimulation in mouse models. In my first year at NYU Tandon, I joined with other NYU Tandon students to develop smart glasses for EMT workers to use to stay in contact with nearby hospitals and transmit live patient data to ER physicians/nurses prior to the patient arriving in the ER. We used this idea to win first-place in the NYU Tandon Made Telecommunication, Telehealth, and Teletherapy enabled by Communications (5G) Challenge and accepted into NYU's InnoVention 2020-2021 Cohort – after pivoting the idea to a hospital resource management system to track patient flow and equipment in the ER.

Since entering NYU Tandon, I have taken a rigorous courseload, giving me knowledge on fundamental CS concepts in my Data Structures and Algorithms and Object Oriented Programming courses. In addition, as a computer engineering major, I have taken electrical engineering courses, teaching me the components and logic behind common hardware devices. More importantly, I gained the skills for this project through my AI and ML courses. Through my Introduction to Data Science course, I learned how to use NumPy and SciPy for linear algebra and statistical modeling of large datasets. My Artificial Intelligence and Introduction to Machine Learning courses taught me important ML models and algorithms that are used in the industry, such as neural networks, logistic regression, and backtracking algorithms. Currently, I am taking Introduction to Databases, which will be useful towards the end of the project, where I will develop a database for the EHR reports. Through my courses and work experience as a hardware and software engineer, I have skills in Python, C++, Kotlin, and HTML/CSS.