Understanding the Value of Notes in Electronic Health Records

by

Shantanu Karnwal

B.Tech., IIITDM Kancheepuram, 2017

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Computer Science

2019

This thesis entitled:
Understanding the Value of Notes in Electronic Health Records
written by Shantanu Karnwal
has been approved for the Department of Computer Science

_____

Prof. Chenhao Tan

_____

Prof. Martha Palmer

_____

Prof. James Martin

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Karnwal, Shantanu (M.S., Computer Science)

Understanding the Value of Notes in Electronic Health Records

Thesis directed by Prof. Chenhao Tan

The digitalization of health records has enabled the collection of large-scale valuable datasets on healthcare. However, it has led to complaints about the diminishing value of medical notes, and often contributes to the growing physician burnout. Since writing good notes can potentially improve the quality of healthcare, it is important that doctors get some machine assistance with writing notes and bridge that gap in quality. Therefore, we examine the value of medical notes compared to the structured information in electronic health records through a prediction framework. We hypothesize that 1) medical notes provide additional predictive power to structured information; 2) certain parts of medical notes are more valuable than others (for example, original vs. "copy-pasted"). To evaluate our hypotheses, we use the task of in-hospital mortality prediction, using timeseries derived from structured information for the first 24 hours and first 48 hours of a patient's admission. We run an additional retrospective mortality prediction task where we use all of the data associated with the patient's admission. Our results show that although medical notes bring only marginal predictive value to structured information, using them together consistently improves the prediction. Surprisingly, we also find that the usage of more common English words in notes provide more value than the uncommon English words (which also includes Medical words). Our findings indicate that there is great room for further understanding and improving the value of medical notes.

Acknowledgements

I would like to thank my advisor, Chenhao Tan for all of his help. Not only he has emphasized on asking the right research questions, but also on how to write good quality code. He has been a very motivating figure and I have and continue to learn a lot from him. I express my gratitude towards Ziad Obermeyer and Sendhil Mullainathan for providing their expert opinion regularly throughout this work. I would also like to thank my committee members - Martha Palmer and James Martin for their critical feedback on this work. I also acknowledge the feedback and motivation from all my colleagues in our NLP/CSS group. Many thanks to my parents and close friends in India, who provided all the encouragement and support they could in spite of being half a world away. Lastly, I would thank all the faculty members whose classes I took and all the students with whom I collaborated on various projects while at CU. My two years spent here has really helped me think differently, something that I hope I will be able to reflect going forward in my life.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

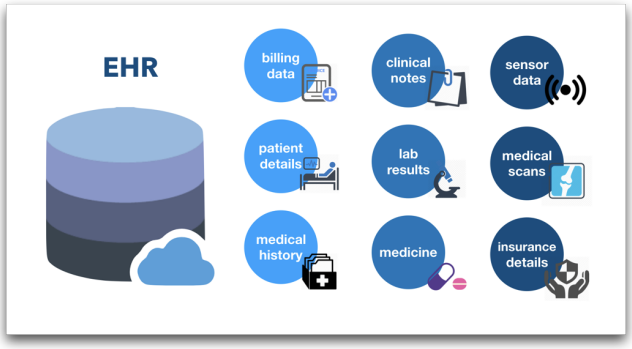
Introduction

1.1 Electronic Health Records - A Mix of Structured Information and Notes

A health record is a collection of documents that show a patient's medical history, usually associated with a hospital, clinic, or medical center. These documents record a wide variety of data about each patient, which usually depends on the kind of problem that the patient has. But broadly speaking, these documents cover two major kinds of information - structured, which are usually numerical or categorical measurements taken of an item, from either the patient or the healthcare providing equipment, and unstructured, which covers notes written by healthcare professionals, explaining in words the problems the patient faces, the solutions that the former recommend, and justification for such a recommendation.

Before the advent of computers and the Internet, health records have been maintained on paper. But in the year 2000, it was revealed in Kohn et al. [2000] that Paper-based Health Records are prone to medical errors to a severe extent. These medical errors are prevalant in maintaining structured information, whether they are incorrect drug prescription or mistaken patient identity. As a result of these errors, more people have died in the year 1997 than the people who died from vehicle accidents, cancer and AIDS. Not only does this fact make healthcare less trustworthy amongst potential patients, but this also creates frustration amongst doctors and healthcare providers as these mistakes were not deliberate. Hence, this course of events gave thought to a new approach of solving this issue, which would be to use computers to store the healthcare records. Not only does this medium bring down the possiblity of causing medical errors, but it also helps in

Figure 1.1: Typical overview of Electronic Health Records and the kind of data they collect. Image courtesy of Mohandas [2018].



maintaining better medical histories.

As Kierkegaard [2011] has described, these Electronic Health Records[1] are being increasingly used in most places around the world and is considered to have opened the gates to new kinds and far more comprehensive forms of medical data collection. Some of the kinds of data now being collected are demographics, insurance, medical notes, allergies, medication, laboratory test results, radiology images, and billing information.

According to the article by Johnson et al. [2008] structured data and notes are often merged together as one single narrative. Although this narrative seems to look like a merger of two distinct entities, it is not entirely true. There is a lot of information that is written down in notes, but is actually structured information. They can either be written in the notes as a result of doctors copying from structured information, or it can also be a result of the patient reiterating this information when they have an interaction with the doctor. An example of this would be the terms like allergies, fluid balance, vital signs, radiology reports, that are reiterated from structured information to notes. On the other hand, information like demographics, billing, diagnoses, charting, laboratory results, input and output would largely be seen as a part of structured information, whereas, information like chief complaints, social history, past medical history, family history, physical examination, assessment and plan are all the information doctors write after interaction with

[1] Electronic Health Records and EHRs have been used interchangably in this paper.

the patient, and hence are seen only as a part of notes. The Venn Diagram in Figure 1.2 gives an overview of this distinction.

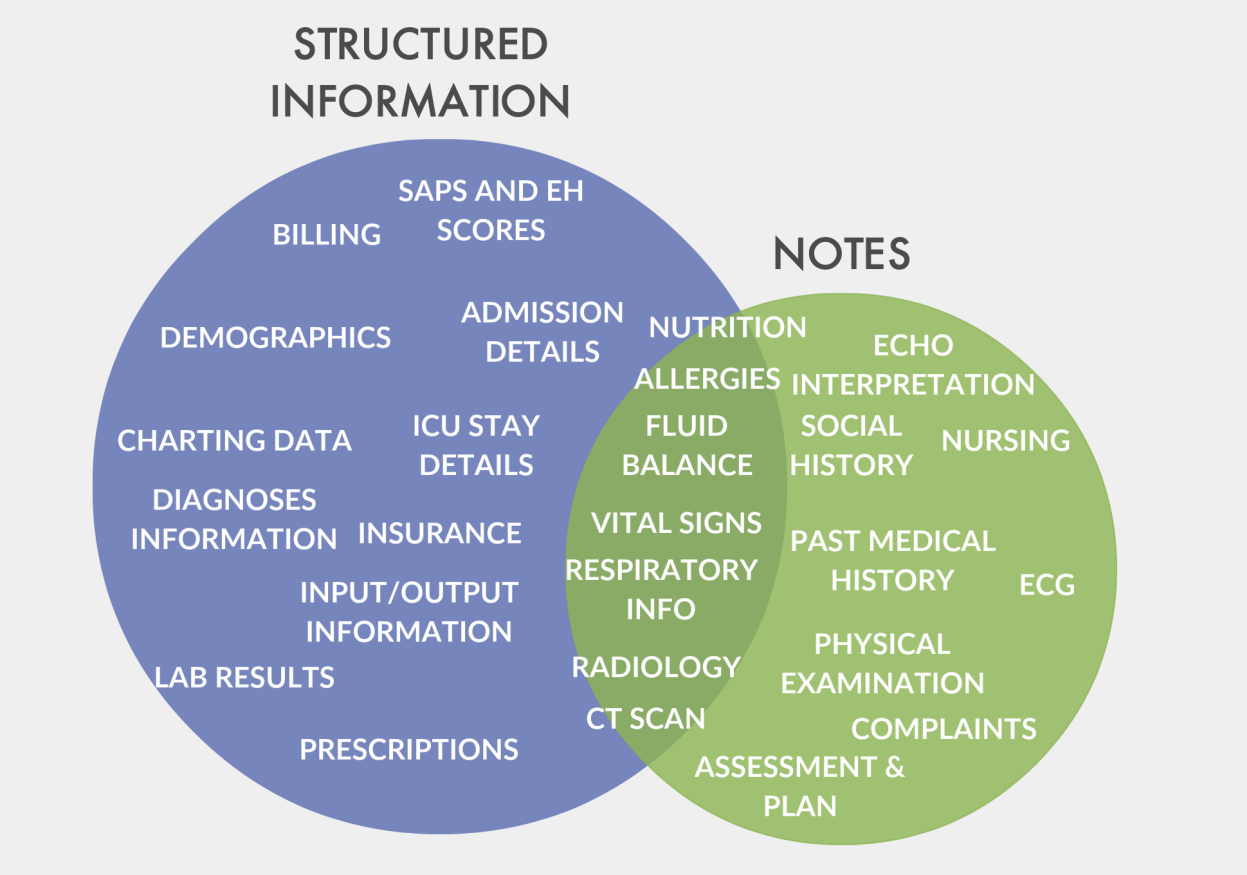
1.2 The Challenges of Writing Good Notes

As it is often said, "Struggle is Not Exclusive", meaning that everyone in this world face challenges at some point in their lives, possibly due to exposure to new environments. And so, as usage of Electronic Health Records increased, physicians and other kinds of doctors started to feel the adversity of computing technology. As this article in The New Yorker [Gawande, 2018] describes in detail, hospitals or medical center managements can sign new contracts with software developers to provide them an EHR platform, and doctors unaware with how to use these new softwares do not really have time to learn it, since new hospital admissions happen all the time. Hence it becomes harder for doctors to learn about their EHR software and attend to the patient with their full mental capacity. This not only makes it a risk for potential patients, but also can cause serious burnout amongst doctors because of increased and tireless hours of work per day. This trouble does not seem to easen up as doctors start getting comfortable with this new software, since different doctors have different ways of defining a problem which in turn can make it harder to look up something to help a particular patient that they are attending to, because these EHR systems generally are very centralized in nature and mix up information from different doctors.

But on the other hand, these systems have helped doctors to quickly check medical records of patients with a similar medical history and can recommend what they can do with the patient that they are currently attending. This has also helped patients, who now can also use the system to retreive all of their medical history and also remain cautious about the necessary steps they can take to avoid medical attention in future.

The other challenge as described by Van Vleck et al. [2007] is the problem of information overloading, i.e., EHRs have increased the scope of information collection to the extent that physicians find it difficult to identify what pieces of information is relevant for them to write good notes. It talks about how the notion of evidence-based medicine, which is the ability to make decisions

Figure 1.2: A Venn Diagram representation of Structured Information and Notes in a typical Electonic Health Record. The intersection of the two sets represents the structured information that often appears in notes.



based on available evidences, is the only area that is being paid attention to in healthcare. However, the challenge of how to help physicians identify relevant pieces of information to write good notes has been vastly ignored. It does a study on a large group of physicians to give them patient records and asks them to identify what information they find relevant. What it finds is that most physicians give a thorough attention to notes (especially admission notes), while a small subset of physicians also look at laboratory results to arrive at their conclusions. It also analyses the fact that use of EHRs have increased sloppiness of physicians to the extent that they now copy a lot of information from previous notes, rather than writing crisp, concise and original notes.

A large section of the physicians surveyed in Van Vleck et al. [2007] also expressed the desire to have trend charts, or visualizations of laboratory data available which they think would help then arrive at better and more accurate conclusions, and help them write better notes. This study also gives us the intuition that notes have a lot of potential usefulness, and in times of emergencies, a doctor would prefer to read a note rather than look at structured information. A parallel study by tenBroek et al. [2010] arrives at similar conclusions, but also adds that physicians do appreciate the electronic templates made available from note writing and can write notes a little faster as compared to paper records.

Jing et al. [2017] talks about the problem of patients wanting to receive a consitent quality of healthcare all around the world, but doctors can also be inexperienced and might be hailing from rural areas, which has shown that reports written by them is more error prone. In countries with population as large as China, there are less hands providing help and more wanting help, and that leads to doctors having not so pleasant experiences. They do not want to write complete notes on their own, and want assistance with finishing the notes and are willing to provide some keywords to write them.

The article by Adler-Milstein and Bates [2010] further reinforces the views in Jing et al. [2017] by going into the monetary tradeoffs, and talks about the fact that hospitals in rural areas don't have access to the same resources that multi-speciality hospitals in urban areas do, which makes quality healthcare weaker. The article also adds the fact about physicians not having substantial

IT assistance, if they have trouble operating the EHR system that they are working with.

1.3 How can Good Notes Improve Quality of Healthcare?

Before answering this question, it is important to know whether the shift from Paper-based to Electronic Health Records have improved note writing or not. The answer to that is yes, with the results being evident from the study by Hoang et al. [2014], where they use their self-developed QNOTE score (takes into account certain templates that should be there in physician notes and how clear and concise they should be) to display significant difference in scores when physicians used paper to write notes and now when physicians have been using EHR for 5 years. But as we highlighted the challenges in Section 1.2, there is a lot to be done so that notes can have a much better quality.

An important benefit of writing good notes is that in many ways it helps the doctors themselves. As described brilliantly by Robey [2011], good notes maintain a continuity in healthcare. If suppose a patient gets transferred from an old hospital to a more specialized hospital, it is essential that the attending physician is well aware of the patient's illness. To be well aware, they tend to rely solely on notes, especially previous physician notes, discharge summaries, and list of prior illness, because if all of these are well written, a quick skim over all these notes would be able to provide the most amount of information to the doctor, making them ready to provide their best help to the patient. Not only does this make healthcare a lot safer and reliable, it also reveals the doctor's compassion for the patient. An important property of good notes is that it should be able to narrate a whole story, so the better the story, the better the note. This art comes from more practice of writing notes, and hence it is important for experienced doctors to write good notes, which can become an inspiration and a benchmark for new doctors to follow and keep on continuing this practice.

Another big advantage that writing good notes can offer is better Patient Engagement, which as described by James [2013], is the practice of keeping the patient in-the-loop with the doctor, whether it is quick information retrieval for patients or collective decision being made by both the

patient and the doctor to solve a particular problem. Patient engagement becomes possible because the retrieval of notes is now much more simpler and both patients and doctors now have access to it. If the notes are well written, then it increases Patient Activation, i.e., the patients understand their condition, which makes them more self-aware and willing to take better care of themselves.

A big example of this practice is evident from the OpenNotes Project [Bell et al., 2018], which makes notes available to the patients. The reports said that the patients usually forgot what the doctors told them during their meetings, but since now they can review their notes, they would remember what precautions to take and hence keep both patient activation and engagement in play. This not only helps patients to trust their doctors, but it also makes doctors more satisfied. Patients can also provide their own necessary feedback to the doctors if they think the doctor has written something wrong in the note, and hence can become a partial co-author of the note. This practice can potentially refine the practice of note-taking and hence can mostly do away with the notion of copying and pasting notes, which developed as EHRs came into play.

1.4 Motivation - Use Machine Learning to Bridge the Gap

As we saw in Section 1.2, though notes in Electronic Health Records provide a lot of advantage compared to Paper-based records, they still pose a lot of challenges to the doctors. In Section 1.3 we saw the potential benefits that notes can offer, provided that we can mitigate these challenges. So how can we solve this problem? An excellent solution could be to not mix up information from different doctors and provide assistance to doctors individually, depending on the type of help they need. This sets up an opportunity to use Machine Learning to help, since we need both quick and quality assistance, and recent work in this area has proved to be of an excellent use.

The recent New England Journal of Medicine article [Rajkomar et al., 2019] elaborates on this theory, since there can be lack of staff, and if there is, then it would take time for another caretaker to help the doctor attend to the patient. Yet, doctors do need assistance with their work, especially in note-writing, and hence usage of machine learning becomes the fundamental groundwork. And as it turns out, the problem of information overloading is actually a blessing

for the machine learning model, since more data that is available, the better the model is able to generalize to unseen examples. Scanning through millions of patient records in a very small amount of time is something machine learning has no problem providing, something which cannot be done with assistance from another human.

A good machine learning model can learn patient health trajectories [Beaulieu-Jones et al., 2018], which is a way to form a sequential model of interation of the patient and the doctor, which can help predict future course of events in the patient's health. Machine learning can also help in predicting accurate preliminary diagnoses and final ICD codes [Xie and Xing, 2018], and can also make patients aware of conditions that they may face later after getting discharged from the hospital. It is also important for machine learning to know what the ideal health condition of a patient should be, for which the data that the model learns on may not be ideal and hence it should be provided with some carefully crafted artificial data or be given some knowledge of what causal effects are. The problem of note-entry can also be solved with machine learning methods like speech-to-text dictation, quick look up and autocompleting sentences catering to the physician, and also selection of useful information from past notes written by the same doctor and other caregivers during the current and previous patient visits.

We have seen now that machine learning offers numerous benefits to physicians and other kinds of doctors, and can potentially overcome the challenges in Section 1.2, but it can also do much more. For instance, Gabriels and Moerenhout [2018] explains how self-tracking devices like Fitbit and Apple Watches can record sensor data from patients and how this data can be used by machine learning algorithms with physicians to provide accurate interpretations of these data, although its real incorporation in the clinical practice is yet to be done and is still part of active research discussion.

Since we need to address our inital problem of helping doctors write good notes, it is important that we should know how to help doctors to write them. And **in order to learn that, we must know what information in notes is useful**. This is why we have been motivated to solve this problem. If we can find what parts of notes are useful to the doctors, we can help save a lot of the

doctor's time and provide them with the relevant information to write clean and good notes. It may be further useful to know what part of the note-writing should be delegated [Lubars and Tan, 2019] to the machine learning model and what should the physician write themselves on the basis of their analysis and judgement.

1.5 Research Contributions

We provide the following research contributions -

- We provide a new logistic regression classifier with much more extensive features than the baselines to calculate in-hospital mortality, with the help of the MIMIC-III dataset. We show that our model outperforms the baselines by a considerable amount.
- Secondly, we find that notes provide additional marginal value and predictive power to structured variables.
- Next, we find that Physician notes are not the most useful category of notes compared to other categories. In fact, Nursing/other notes is the most useful category of notes instead, likely driven by the volume.
- To our surprise, we also find that common English words in notes are more useful compared to uncommon English words in notes, which also includes most medical words.
- Moreover, we find that nouns in notes are more useful than adjectives or verbs in notes.
- Lastly, we find that notes that are more similar to notes written prior to them (what we term as "copy-pasted") are less useful than notes that are less similar, i.e., notes having more original content.

1.6 Organization

This thesis is organized as follows. Chapter 2 discusses the relevant literature related to how machine leaning has been used in Electronic Health Records. In the end, we then form concrete

hypotheses that we would like to evaluate. Chapter 3 discusses the MIMIC-III data and how we preprocess it to form timeseries from it for our Mortality Prediction task. We also show a lot of trends and properties that exist in the MIMIC-III notes, by doing a word-level analysis. Chapter 4 discusses how we go about testing our hypotheses, our final stages of pre-processing and the logistic regression classifier that we use in detail. It also evaluates all of our hypotheses with the detailed results in tables. Chapter 5 provides the conclusion and future work.

Chapter 2

Applications of Electronic Health Records in Machine Learning

2.1 Overview

In this chapter, we try to answer the question of how Machine Learning has been applied to Electronic Health Records and the major findings that were obtained from those experiments. We do this by extensively reviewing recent literature, which cover the different ways in which Health Care researchers and Machine Learning practitioners have used Machine Learning algorithms in EHRs. Our literature research looks at two major sections - various machine learning methods and applications used in EHRs, and Medical Note Generation as a future motivation. We end this section by talking about the hypothesis and findings that we expect from our research.

2.2 Machine Learning with Structured Information and Notes in EHRs

An important area in which machine learning has been used in EHRs, is for the in-hospital-mortality prediction task, as described in [Ghassemi et al., 2014]. The authors use Latent Dirichlet Allocation to convert all of the patient's notes into features, and use these features to predict a patient mortality. These are done in 3 timelines - in-hospital stay, 30 days and 1 year post patient's discharge. The features used comprised of baseline features, statistical features and features from the notes. To be precise, notes were tokenized first and later their stopwords were removed. CountVectorizer was used to convert LDA to count features amd LDA was used to convert them into groups of 50 topics, hence 50 features. These were passed onto a linear SVM model after doing a 70-30 train-test split. Various models were used in the SVM - Admission Baseline Model with

3 features, Time Varying Topic Model with 50 features, Combined Time Varying Model with 53 features, Retrospective Derived Features Model with 36 features, Retrospective Topic Model with 50 features, Retrospective Topic with Admission Model with 53 features, and Retrospective Topic with Derived Features Model with 86 features. The final conclusion from the paper comes with the fact that LDA features from notes are better than structured features, and all features combined give the best prediction of mortality.

The experiements with timeseries have been used a lot for predicting mortality of patients. The works by Harutyunyan et al. [2017], Ghassemi et al. [2014], Che et al. [2018], Johnson et al. [2017a] and Purushotham et al. [2018] have all focused on this task. Since there were so many experiments available regarding prediction of mortality, we find it right to use the same task to find the value of notes, since we will be able to make a good comparison with the existing implementations. We discuss details of some of these papers in later chapters.

There are a lot of Deep learning applications in EHRs as well. Deep learning is a subarea of machine learning which specializes in self-representation learning, i.e., given raw data and target outputs, it decides to discover patterns and learn features on its own without human help. This representation is generally arranged in multiple layers to learn good patterns from very complex datasets. Esteva et al. [2019] gives a brief overview of how these are being applied to EHRs. It talks about how Convolutional Neural Networks (CNNs) are being applied for medical imaging, especially in radiology applications, giving very good performance. CNN being used with transfer learning methods, i.e., methods to learn CNN on very large datasets like ImageNet but usage of those models on medical imaging data has yeilded very good results. In Natural Language Processing, deep learning is used in the form of Recurrent Neural Networks (RNNs), to learn features from the medical data as a time series. It can incorporate structured data like demographics and lab results, and also data from notes. Unsupervised learning methods like usage of autoencoders [Beaulieu-Jones et al., 2018], which is the practice of learning useful features from data by first reducing the dimensionality and then reconstructing unlabeled data, has also been prevalent in these applications. There has been active research on RNNs being used with very large (size of

as big as 46 billion data points) medical datasets [Rajkomar et al., 2018] to combine both the structured data and unstructured (note) data, to give some very impressive results in in-hospital mortality prediction, 30-day readmission prediction, length of stay forecasting and final discharge diagnoses prediction.

In a slightly similar work, Shickel et al. [2018] does a detailed survey on the current deep learning methods that are being used in the notes in Electronic Health Records. The first application that it is primarily used in is information extraction, whether it is a singular medical concept (like disease, procedure, treatment), temporal event (a singular medical concept with respect bounded by time - like last week, last month, last 6 months), relations between medical concepts in notes, or expanding abbreviation with the help of free text available. The other applications include representation learning of ICD-9 codes, i.e., learning a real value vector representation of an ICD-9 code[1] and mortality prediction, i.e., predicting whether the patient would die within their course of stay in the hospital (in-hospital mortality) or would die after a certain period of time after they get discharged (out-of-hospital mortality).

Another very interesting application of notes in EHR is phenotyping, an active area of research in medicine, which can possibly give us more fine-grained subdivisions in types of diseases and give us better precision in specific diagnosis. This not only would improve how we would define diseases now, but it also gives us the possiblility to discover new phenotypes. This is evident from the fact that Google's Deepmind was able to solve the protein-folding problem [Evans et al., 2018], something what medical researchers thought would be impossible to solve.

While all of these research have conveyed the fact that EHRs extensive usage in machine learning has been very proficient, the discussion doesn't just stop there. It is also true that notes alone can provide a lot of value to a patient's health record apart from structured variables, and has been very crucial for some very important prediciton tasks. An example of this is the paper by Liu et al. [2018a] which uses CNN, RNN and Bidirectional RNN on both the structured variables

[1] ICD-9 code is the International Statistical Classification of Diseases and Related code in its 9th revision and is assigned to a patient after they get discharged from the hospital or medical facility.

and word embeddings from notes on a dataset of over a million patients, and found that the notes to perform better than structured variables to predict if the patient would develop a chronic onset of disease.

The study by Luo [2017] shows how sentence level and sentence-segment level Long Short Term Memory (LSTM)recurrent neural networks can be used to classify relations between medical entities in notes, and it is found to perform better if the word embeddings are training on medical vocabulary (more in Section 2.3). Another work, by Lee et al. [2017] shows how we can model notes into embeddings and use then in a feed-forward neural network to obtain named entities, which would help in dataset deidentification, and therefore it would be easy to share the dataset with people who would be interested in this research. The Sachan et al. [2017] paper also performs a Biomedical Named Entity Recognition, but uses a labeled dataset instead of training on an unlabeled note.

Sometimes, thinking about all of these experiements, we may figure out that it is obvious that patients will not visit the doctor when they are healthy, and will only go when they are unwell and need help. Hence this can lead to bias in our trained machine learning system, since we would not know what the ideal health condition for the patient would be. In order to study that, Zheng et al. [2017] run a study on using Hidden Markov Models to form a timeseries of both the structured and note data at regular intervals and impute whatever data they are missing. They then run several LSTM variants on applications like in-hospital mortality and predicting ICD diagnoses and obtain better results than the setup when the timeseries is irregular. Another work that employs the usage of timeseries is by Lipton et al. [2016] where the authors use a regular LSTM to do a multilabel classification of diagnoses using an irregular timeseries of various sensor and laboratory results.

There are a lot of medical concepts that people not belonging to the field of healthcare are oblivious to. Hence when researchers decided to use neural networks for medical notes, which have a lot of medical terminologies in them, then the usage of standard english langauge word embeddings like Skipgram, GloVe, Word2Vec and stacked autoencoder did not perform well. Hence for this purpose, there is a lot of work on creating specialized word embeddings that cater towards medical

concepts. We discuss some of these in detail.

A very common project being talked about nowadays is the Med2Vec project [Choi et al., 2016]. Med2Vec takes two main things into consideration - the set of all ICD codes there are and a set of all patient visits to the hospital. For both of these values, Med2Vec's aim is to convert them into real valued vector representations (or embeddings). The initial visit vector is taken and after a linear combination (of weights and biases) and a ReLU activation, it is concatenated with a patient demographic vector and the previous process is repeated again to return the final visit vector. The final visit vector after a softmax returns the predicted ICD code. Comparing this technique with the normal english word embedding techniques gave a higher accuracy of diagnoses code prediction.

Another very common project for medical word embeddings is the Graph-based Attention model or GRAM [Choi et al., 2017]. Like Med2Vec, GRAM also takes the set of all ICD codes and patient visits as its main inputs. But here the initial visit vector is not created straight away, and instead the codes are arranged in the form of a directed acyclic graph, with the more general concepts at non-leaf positions and the more specific codes at leaf nodes. Depending on the type of visit, the code is taken from one of the leaf nodes along with its parent nodes and then these codes are converted into the embedding space and combined together by using an attention mechanism. After passing this through a tanh activation, it returns the final visit vector. This vector after passing through a feed-forward neural network returns the predicted ICD code, which is seen to give a higher AUC score than the baselines.

An excellent example of how embeddings can be run on unstructured data (or notes) is given by Zhu et al. [2018]. It first forms a contextual word embedding with ELMo, by using a corpus of clinical notes and wikipedia pages related to medicine. After the training is done, this trained embedding is run on different kinds of notes available in EHRs with the help of a bidirectional LSTM, and returns the precise clinical concepts related to a word or phrase in the input note. This technique achieves the best F1 score for clinical concept extraction and has a lot of uses in mortality prediction, diagnoses prediction and automated de-identification.

An inportant consideration that we should keep is that out of all the literature that we have seen, whether the machine learning model uses the structured data or unstructured data, using one does not render the other one useless. Notes are only able to provide additional value to the structured data. Hence it is important to see what we can do if we use them both together, which we expect to increase the model performance. Hence we also look at multimodal embeddings, which are the embeddings that run for different modalities of data, like image and text, structured tables and text, etc., since using these can potentially boost up our machine learning model's performance.

The paper by Kiros et al. [2014] provides a good introduction to this concept. That concept can be easily extended to note writing, especially for writing notes from radiology reports. The images can be given as inputs to the log-bilinear model defined in the paper to generate radiology report from the images. An implementation of generating radiology notes has been tried in Jing et al. [2017] but can be extended to support this notion of modality.

There are some machine learning methods that have been used for general applications in which prediction needed to be made from data coming from various sources. An example of this is the paper by Card et al. [2018], which introduces a topic model called Scholar, which takes as input the text documents, and then uses a modified version of Latent Dirichlet Allocation, after which the intermediate representation is passed onto a Variational Autoencoder, which can also take as input any word embedding of the metadata available with the document, and then predicts the final document label. The system when tested on a news articles, imdb and yahoo datasets returned the highest accuracy compared to other linear models and topic model baselines.

The multimodal embedding ideas have been used in the EHR setup as well. For example the paper by Jin et al. [2018] first uses named entity recognition on the note data to extract entities and important representations from it in the form of embeddings, then combines them with time series signals to a final feed-forward neural network which returns the patient in-hospital mortality label. This implementation result compared with the baseline implementations gives a 2% higher ROC AUC score. This implementation gives us some insight as to how we can manage to merge data from structured and unstructured sources if we want to predict in-hospital mortality.

2.3 The ideal future solution - Medical Note Generation

Currently, the goal of the thesis is not to generate medical notes. It is rather to find what parts of notes are important which would further help to brainstorm better approaches to note generation in the future. But inspite of that, it may be useful to see what good text generation approaches are out there and how we can extend our work to text generation.

Holtzman et al. [2018a] addresses the problem of using a simple RNN for language generation, which is the issue of the generated text not at the level of communication specified by Grice's Maxims of text having informativeness, truthful, relevant and unambigous. Hence the authors develop sub-models that can capture all these 4 qualities in the generated text. These sub-models are simple encoder - decoder models with an objective, which is to comply as much with the Grice's Maxims, and also a weight balancing scheme that would balance the combined output of all sub-models. These scores are then ranked and the highest ranking text is the one which the system would return in the end. This model gets very high scores compared to the vanilla RNN on human evaluation and turing tests. Another paper by Holtzman et al. [2018b] tries to solve this problem again but by using RNN but would give a higher reward to texts that are more coherent with a provided reference text and comply with the Grice's Maxims.

Another good text generation reference that can be used on medical data is the paper by Bosselut et al. [2018], whose objective is to generate text by keeping the discourse structure as close to the reference text. The closer it is the more the model would be rewarded. The generation is done using an RNN encoder-decoder network and the reward is given by a policy iteration reinforcement learning algorithm. Since the most important aspect of this method is the order of sentences, it can be applied to medical data as well where it needs to be figured out from the current patient condition of what should be the assessment and plan and therefore what steps need to be taken, for which order is very important.

Not all text generation approaches are abstractive, some are extractive as well. Extraction based methods have been mainly used in note summarization. One example of this approach is given

in Zhang et al. [2018], where the goal is to generate a concise 1-line impression of a radiology report. For this purpose they used an encoder-decoder model with attention, one for the background part of the report and one for the findings part. After encoding the text from both parts and getting the attention, both are given as input to the decoder which would generate an overall vocabulary distribution. The words or phrases for which the distribution is highest would be used to generate the impression. This approach gave significant improvements in ROUGE score over extractive baselines and pointer-generator networks.

A second example of extraction based text generation approach is given by Liu et al. [2018b], in which the authors use unlabeled note data from EHRs, and try to generate summaries from those by using neural extraction methods. The way they do it is by having the dataset of all notes for all patients, and then get a cover, i.e., the maximum similarity of that note with any previous note. After that a binary label vector is calculated, which is also known as pseudo lablels. This vector is fed onto a neural network, which is a word level bidirectional GRU, the output of which is given to a sentence level bidirectional GRU, whose output passed through a sigmoid returns 0 or 1, depending on whether the model finds that sentence important for the summary generation or not. They also show to get the highest ROUGE score compared to all of the baselines.

We also found one paper, by Liu [2018] that actually implements the task of medical note generation. Since the language models needed to write notes are supposed to be very restricted, it is important to know all the details about the patient, like lab results, medications, demographics and notes written previously to write the current note. It gives the hypothesis that notes seem to have a common template, which can be learnt easily, and a lot of the content in the notes can be predicted. It uses both the static data and the timeseries data, from both the structured and unstructured sources, to generate the notes. The note generation problem is defined as a supervised learning task which takes as input a note context, which consists of patient past data, intended category of note and a seed of 10 tokens that should be there in the output note. The patient past data (which is a part of note context) consists of gender and age, prescriptions, laboratory values with flags of abnormality (if any) and all the past notes. It then uses all of this information and uses

a transformer, and a transformer with memory compressed attention to generate the note, in which the latter gets a better score. The evaluation metrics that were used were perplexity per token - how well the token is predicted given past tokens, accuracy of next token, ROUGE-1, ROUGE-2, ROUGE-1 after removal of autopopulated templates and gender & age accuracy.

All of this work give us a very good intuition as to how we would approach this task of note generation in the future.

2.4 Hypotheses Formulation

After doing the extensive literature review as we can see from the previous sections in this chapter, we should be able to tell what we think we would get after we do our research. Hence, in accordance with medical notes, we have formulated the following hypotheses -

- **H1 -** Notes provide additional predictive power to the structured data in Electronic Health Records.
- **H2 -** Certain parts of notes are more valuable than other parts -
    - **H2.A -** Physician notes are the most useful compared to the other categories of notes.
    - **H2.B -** The common english words in notes are less useful than uncommon english words (which may also include a large fraction of medical words).
    - **H2.C -** The nouns in notes are more useful compared to both adjectives and verbs.
    - **H2.D -** The notes that are similar to previous notes (what we term as "copy-pasted") are less useful compared to notes that less similar, i.e, are original and concise.

Since we discussed in Section 2.2 of how there are so many tasks related to mortality prediction, we use the same task to test our hypotheses.

Chapter 3

Working with the MIMIC - III Database

3.1 What is MIMIC-III?

The Medical Information Mart for Intensive Care - III or MIMIC-III is a freely available medical database consisting of de-identified patient records. These patients were admitted in the Beth Israel Deaconess Medical Center between 2001 and 2012. Although the total number of tables in this database is 26, but more broadly speaking, this database consists of two major types of information - Patient information and Hospital information. Patient information consists of basic patient data, admission details, demographics, insurance, etc. Hospital information consists of events, like laboratory events, charting events, note data and also billing information. We look at both of these as we define some of the tables that we use for our implementation. We chose the MIMIC-III dataset for our research becuase it is freeely available, has de-identified patient records and a variety of different patient cases, especially in the ICU, which would help us get a more generalized machine learning model.

3.1.1 Patient Related Tables in MIMIC-III

The first table is the PATIENTS table. Out of the columns that we are interested in, PATIENTS consists of SUBJECT_ID, which is the unique identifier of an individual patient, and stays the same no matter how many times the patient is admitted to the hospital. Other than that it consists of GENDER and DOB (Date-of-Birth). In the DOB section, if the patient's actual age is equal to or more than 90 years, then the DOB is shifted back 300 years before admission,

accordingly. The PATIENTS table has in total, 46,520 records.

Table 3.1: An example of what the PATIENTS table would look like. The values are not taken from MIMIC-III and are falsified. Also, the actual table has more columns than these.

	SUBJECT_ID	GENDER	DOB
0	33	M	3021-12-31
1	34	M	3047-09-21
2	35	F	2988-07-13
3	36	M	3004-04-15
4	37	F	3012-05-25

The next table is the ADMISSIONS table. Out of the columns that we are interested in, ADMISSIONS consists of SUBJECT_ID, which acts as a foreign key to PATIENTS, HADM_ID, which is the unique hospital admission identifier associated with a patient. For the same patient getting admitted into the hospital multiple times, a new HADM_ID is generated each time. The other attributes include ADMITTIME, DISCHTIME, i.e., the admit and discharge times, and a HOSPITAL_EXPIRE_FLAG, which tells if the patient died in the hospital during that admission (1), or if the patient was discharged (0). If a patient unfortunately, dies in the hospital, then that patient's DISCHTIME is same as the DEATHTIME (another attribute in the ADMISSIONS table). The ADMISSIONS table has in total, 58,976 records.

The third table in this list is the ICUSTAYS table. Out of the columns that we are interested in, ICUSTAYS consists of SUBJECT_ID and HADM_ID, acting as foreign keys to both the PATIENTS and ADMISSIONS table, ICUSTAY_ID, which is the unique identifier assigned to the patient when they are admitted in the ICU. The same patient can be admitted multiple times in the ICU, each time getting a unique ICUSTAY_ID, while the HADM_ID would remain the same if all those ICU admissions took place within a single hospital admission. The other attribute that we are interested in is DBSOURCE, which says whether the DBSource is CareVue or MetaVision or both. This is discussed more in Section 3.4. The ICUSTAYS table has in total, 61,532 records.

The fourth table is the SAPSII table. SAPII is not among the 26 tables that we said belongs

Table 3.2: An example of what the ADMISSIONS table would look like. The values are not taken from MIMIC-III and are falsified. Also, the actual table has more columns than these.

	SUBJECT_ID	HADM_ID	ADMITTIME	DISCHTIME	HOSPITAL_EXPIRE_FLAG
0	33	1033	3021-12-31 02:30:00	3022-01-04 12:30:00	1
1	34	1034	3056-09-21 12:30:00	3056-12-31 13:30:00	0
2	34	1035	2988-07-13 13:30:00	3021-12-31 14:30:00	1
3	35	1036	3004-04-15 16:30:00	3004-05-23 19:30:00	0
4	36	1037	3012-05-25 22:30:00	3012-05-36 23:30:00	0

to MIMIC-III. Instead, it is derived using the ICUSTAYS table. It contains the SAPSII Score, which is the Severity of Illness Score [wik, 2018], given to the patient within 24 hours of the patient's ICU Admission. Therefore it has the columns, SUBJECT_ID, HADM_ID, ICUSTAY_ID and SAPSII. Like ICUSTAYS, SAPSII has 61,532 records.

3.1.2 Hospital Related Tables in MIMIC-III

The first table is the CHARTEVENTS table. With 330,712,483 records, it is the biggest table in the MIMIC-III database. CHARTEVENTS contains all the data from the patient coming in from the ICU charts, which include things like heart rate, lab values, ventilator settings, mental status, and so on. The columns which are of interest to us in this table are - SUBJECT_ID, HADM_ID, ICUSTAY_ID, ITEMID, which shows what is the concept that we are measuring, CHARTTIME is when the ITEMID was measured, VALUE and VALUEUOM which are the amount or category of the ITEMID and unit of measurement respectively.

The next table is the LABEVENTS table. This table contains all laboratory measurements and the values in this table can be duplicated in the CHARTEVENTS table. If that is the case, then we take the LABEVENTS value as the ground truth. The columns which are of interest to us in this table are SUBJECT_ID, HADM_ID, ITEMID, CHARTTIME, VALUE and VALUEUOM. Note that ICUSTAY_ID is not a part of this table since it covers laboratory measurements that are also outside the ICU. The LABEVENTS has in total, 27,874,055 records.

Table 3.3: An example of what the ICUSTAYS table would look like. The values are not taken from MIMIC-III and are falsified. Also, the actual table has more columns than these.

	SUBJECT_ID	HADM_ID	ICUSTAY_ID	DBSOURCE
0	33	1033	20033	carevue
1	34	1034	20034	metavision
2	34	1035	20035	carevue
3	35	1036	20036	carevue
4	35	1036	20037	metavision

The next two tables are INPUTEVENTS_CV and INPUTEVENTS_MV. Both of these tables cover Inputs, i.e., any fluids that have been provided to the patient either orally, through tube or through syringes (containing medications). INPUTEVENTS_CV is the record of all such input events from the CareVue information system and INPUTEVENTS_MV is the record of all such input events from the MetaVision information system. For the CareVue data, we are interested in SUBJECT_ID, HADM_ID, ICUSTAY_ID, ITEMID, CHARTTIME, VALUE and VALUEUOM. The INPUTEVENTS_CV has in total, 17,527,935 records. In the MetaVision data, there is no CHARTTIME. Instead it has a STARTTIME and an ENDTIME. We consider ENDTIME as CHARTTIME because we are interested at when the patient would completely receive the medication (or any other input) provided. Hence for the MetaVision data, we are interested in SUBJECT_ID, HADM_ID, ICUSTAY_ID, ITEMID, ENDTIME, VALUE and VALUEUOM. The INPUTEVENTS_MV has in total, 3,618,991 records.

The next table is OUTPUTEVENTS. This table covers all Outputs, i.e., anything that is excreted by a patient (for example - urine) or extracted from a patient (for example - through a drain). Like the Input tables, the columns that we are interested in are SUBJECT_ID, HADM_ID, ICUSTAY_ID, ITEMID, ENDTIME, VALUE and VALUEUOM. The OUTPUTEVENTS has in total, 4,349,218 records.

The last and the most interesting table in this section is the NOTEEVENTS table. The NOTEEVENTS table consists of all the different types of notes written by doctors. The columns

Table 3.4: An example of what the SAPSII table would look like. The values are not taken from MIMIC-III and are falsified. Also, the actual table has more columns than these.

	SUBJECT_ID	HADM_ID	ICUSTAY_ID	SAPSII
0	33	1033	20033	16
1	34	1034	20034	18
2	34	1035	20035	35
3	35	1036	20036	21
4	35	1036	20037	44

that we are interested in are SUBJECT_ID, HADM_ID, CHARTTIME, which shows when the note was entered, CATEGORY, which tells about the type of note, and TEXT, which contains the actual note.

3.2 Understanding Notes in MIMIC-III

All the notes in the MIMIC-III table are available in the NOTEEVENTS table (from Section 3.1). All the different categories of notes and their total counts in the database is given in Table 3.9.

An important thing to remember in NOTEEVENTS is that the Radiology, Echo and ECG notes can also be from outpatients, i.e., from a facility outside the hospital. Notes which are from outpatients should be removed for the working of our task, since they do not have HADM_ID. Also patients that were not admitted in the ICU also lack an HADM_ID, and hence those records should be filtered out as well. Echo notes are generally generated using a template. This leaves us with two important kinds of notes that are Physician notes and Nursing notes, since they have a very high count and it may be useful to know what information they can cover, since these are written by physicians and nurses and require touching and talking to the patient.

Physician notes generally have a 10 elements in it, which are Allergies, Last dose of antibiotics, Infusions, Other ICU medications, Vital signs, Fluid balance, Blood products, Physical examination, Respiratory report and Assessment and Plan. When we tested whether how much of

Table 3.5: An example of what the CHARTEVENTS table would look like. The values are not taken from MIMIC-III and are falsified. Also, the actual table has more columns than these.

	SUBJECT_ID	HADM_ID	ICUSTAY_ID	ITEMID	CHARTTIME	VALUE	VALUEUOM
0	33	1033	20033	1	3021-12-31 02:30:00	10.40	L/min
1	34	1034	20034	2	3056-09-21 12:30:00	70.28	mmHg
2	34	1035	20035	3	2988-07-13 13:30:00	33.55	NaN
3	35	1036	20036	4	3004-04-15 16:30:00	12.67	mg/dl
4	35	1036	20037	5	3012-05-25 22:30:00	89.99	NaN

this template exists within MIMIC-III, we got 93.59%, which is quite a high number and explains the fact that physicians do follow a template to write notes and eveything in this template should be well defined. Some additional things like Social history, Drinking habits can explain a lot about the patient's diagnosis, but they do not occur in every physician note and only show up in admission type notes and not progress notes. It may be useful in future to copy the social history of the patient in every physician note for a better template matching.

Nursing notes generally are smaller than physician notes and cover 4 major aspects - Assessment, Action, Response and Plan. They generally lack any statistical information inside of them (unlike phyisican notes that do), and generally have crisp information conveyed in words. Like physician notes, nursing notes too have more progress notes compared to admission notes, which is good because it conveys the fact that physicians and nurses keep writing notes at regular intervals throughout the stay of the patient.

As far as physician notes are concerned, 3 main sections - Everything before Allergies, Physical Examination and Assessment and Plan are the most important sections of physician notes. We find with our analysis that for every physician note of a 5000 character length, the Before Allergies and Physicial Examination sections are only approximately 400 characters long each and Assessment and Plan section is approximately 1000 characters long. This shows that eventhough notes cover a lot of valuable information, they are not crisp and concise and the rest of the information they have can be imported from other kinds of notes and structured variables.

Table 3.6: An example of what the LABEVENTS table would look like. The values are not taken from MIMIC-III and are falsified. Also, the actual table has more columns than these.

	SUBJECT_ID	HADM_ID	ITEMID	CHARTTIME	VALUE	VALUEUOM
0	33	1033	1	3021-12-31 02:30:00	10.40	L/min
1	34	1034	2	3056-09-21 12:30:00	70.28	mmHg
2	34	1035	3	2988-07-13 13:30:00	33.55	NaN
3	35	1036	4	3004-04-15 16:30:00	12.67	mg/dl
4	35	1036	5	3012-05-25 22:30:00	89.99	NaN

We also perform a word level analysis of notes altogether. We remove all the erroneous notes and only keep all the notes except the discharge summary. We show the average number of words occuring in each category of notes, and the average number of adjectives, verbs, nouns, proper nouns, common english words and medical words[1] in each category. We then take a sample of notes from each category and try to get the fraction of these different types of words in each note and plot a distibution. All the plots are available in Figure 3.2 to Figure 3.7. We further show in Table 3.10 the entire breakdown of average counts of different types of words that exists in different types of notes.

3.3 Forming Timeseries with MIMIC-III Data

In order to predict the in-hospital mortality, we need to derive a timeseries and combine our structured variables and note data. The way we do that is as follows. First, we run our extract_subjects.py file. What this will do is read all of the PATIENTS table first and return the SUBJECT_ID, GENDER and DOB. After that it will read the ADMISSIONS table, while filtering out two things - organ donors, i.e., the patients who died already but were readmitted to donate their organs (where ADMITTIME is greater than the DISCHTIME), and the patients who do not have CHARTEVENTS data, because we would not have structured variables to define such cases. It then reads the ICUSTAYS table and gets the DBSOURCE and groups the table to return how

[1] Medical words obtained from - https://github.com/glutanimate/wordlist-medicalterms-en

Figure 3.1: A distribution of the fraction of English words present in MIMIC-III notes. The notes are sampled by 1500 (upper limit) per category.

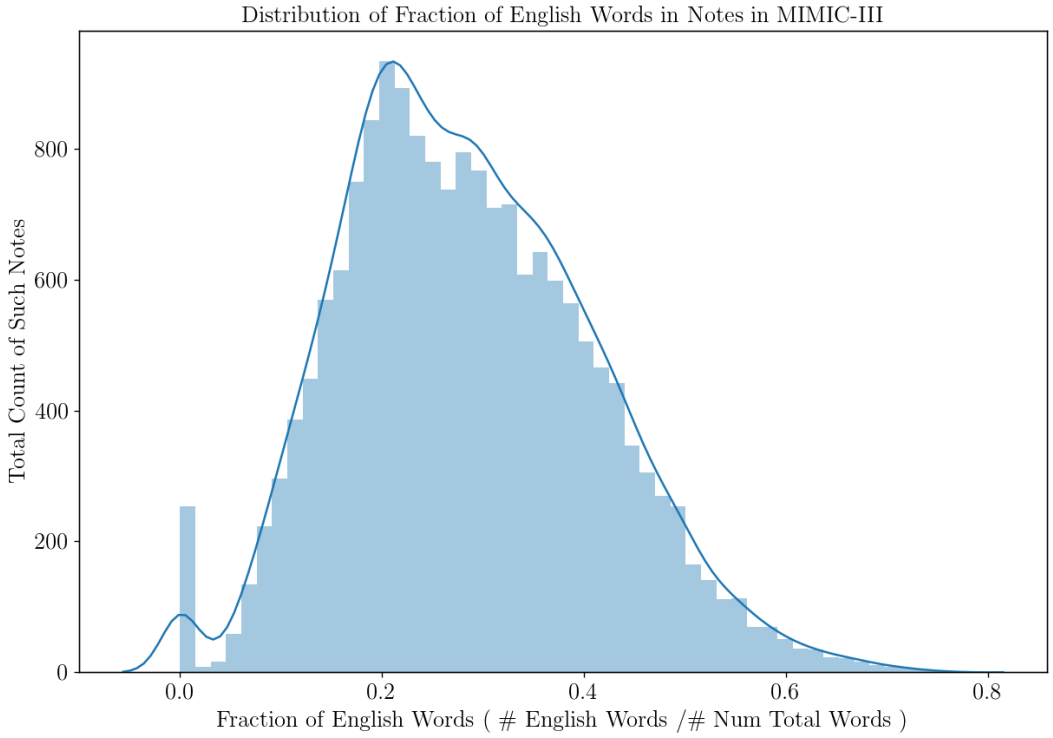


Figure 3.2: A distribution of the fraction of Medical words present in MIMIC-III notes. The notes are sampled by 1500 (upper limit) per category.

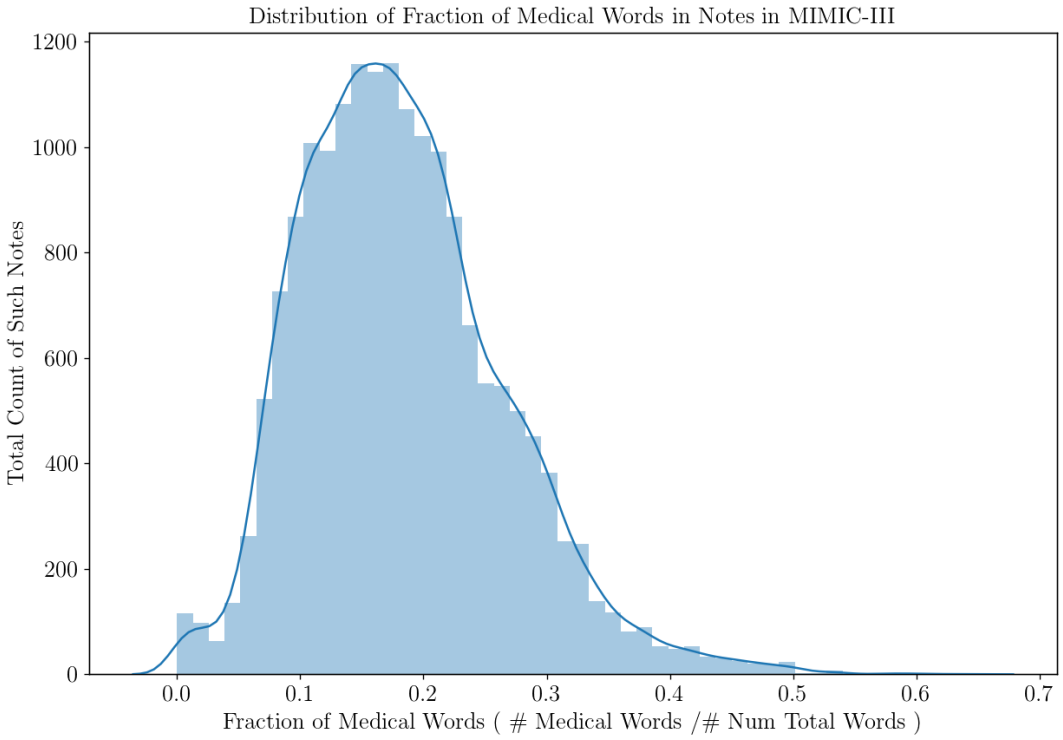


Figure 3.3: A distribution of the fraction of Nouns present in MIMIC-III notes. The notes are sampled by 1500 (upper limit) per category.

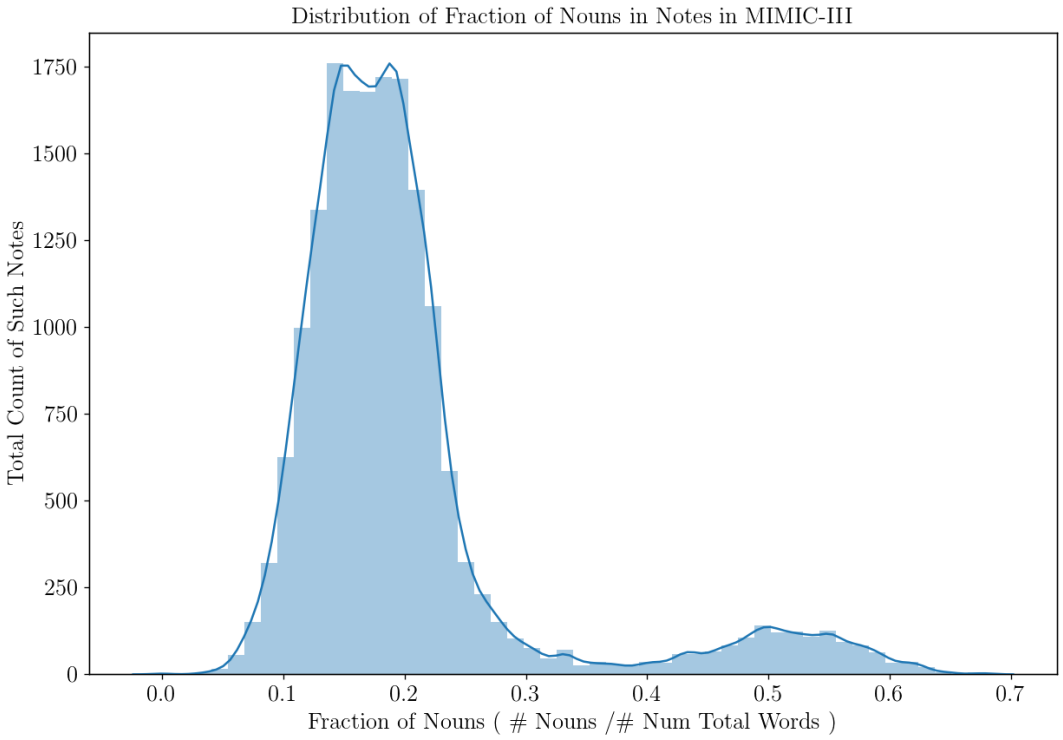


Figure 3.4: A distribution of the fraction of Proper Nouns present in MIMIC-III notes. The notes are sampled by 1500 (upper limit) per category.

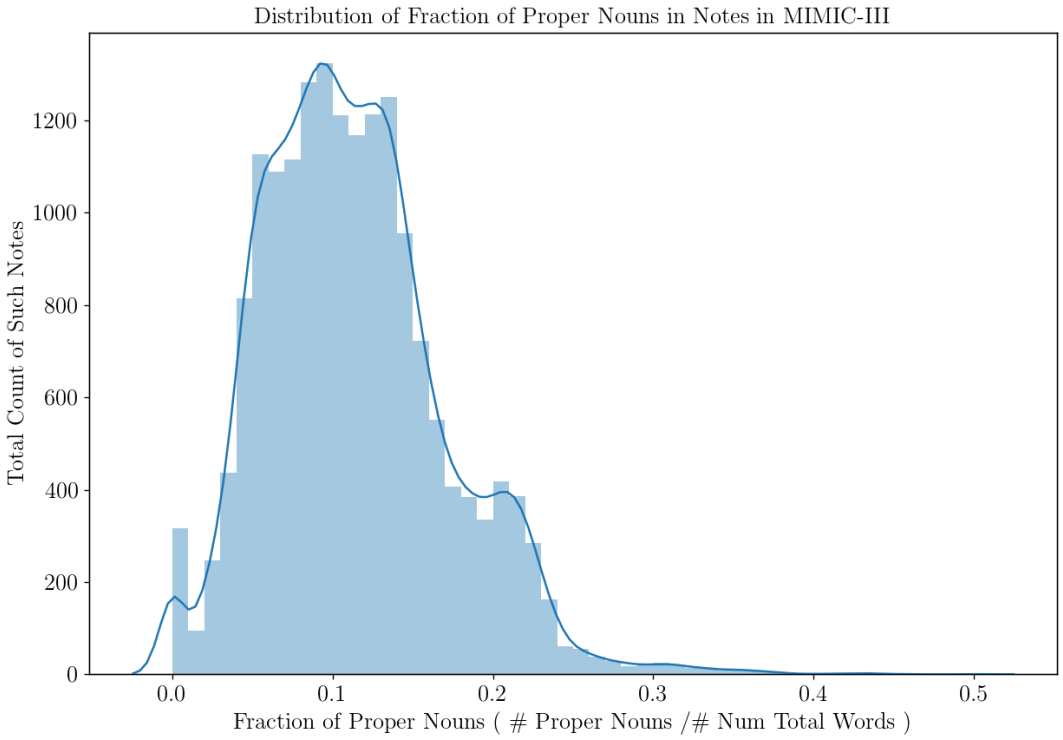


Figure 3.5: A distribution of the fraction of Adjectives present in MIMIC-III notes. The notes are sampled by 1500 (upper limit) per category.

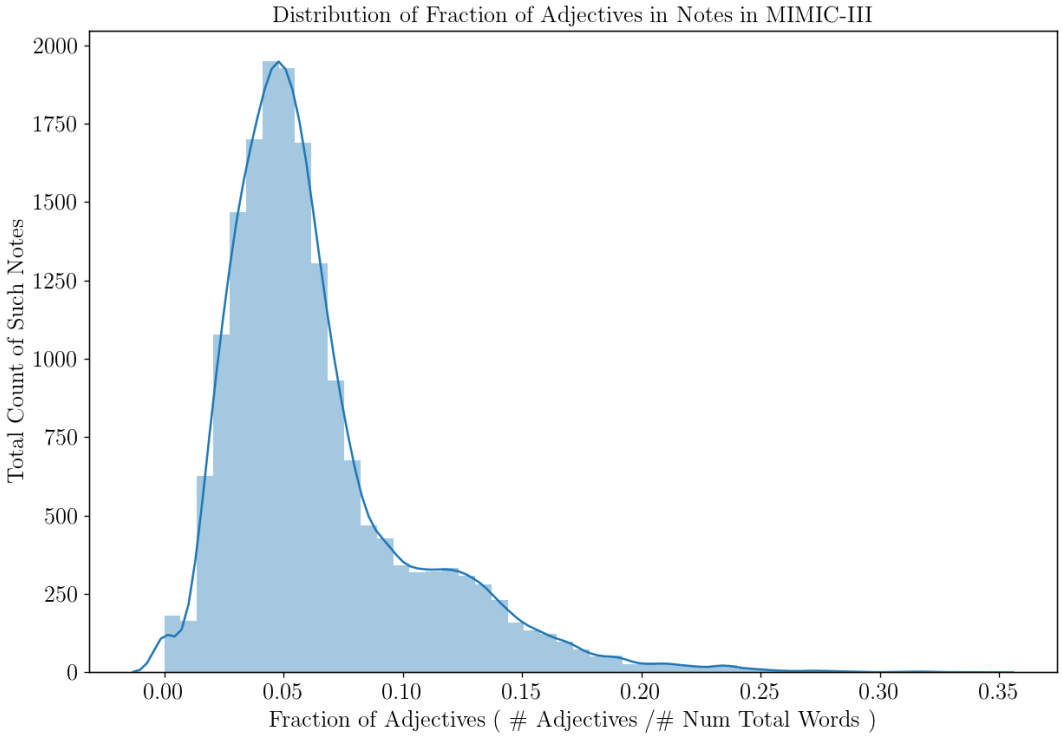


Figure 3.6: A distribution of the fraction of Verbs present in MIMIC-III notes. The notes are sampled by 1500 (upper limit) per category.

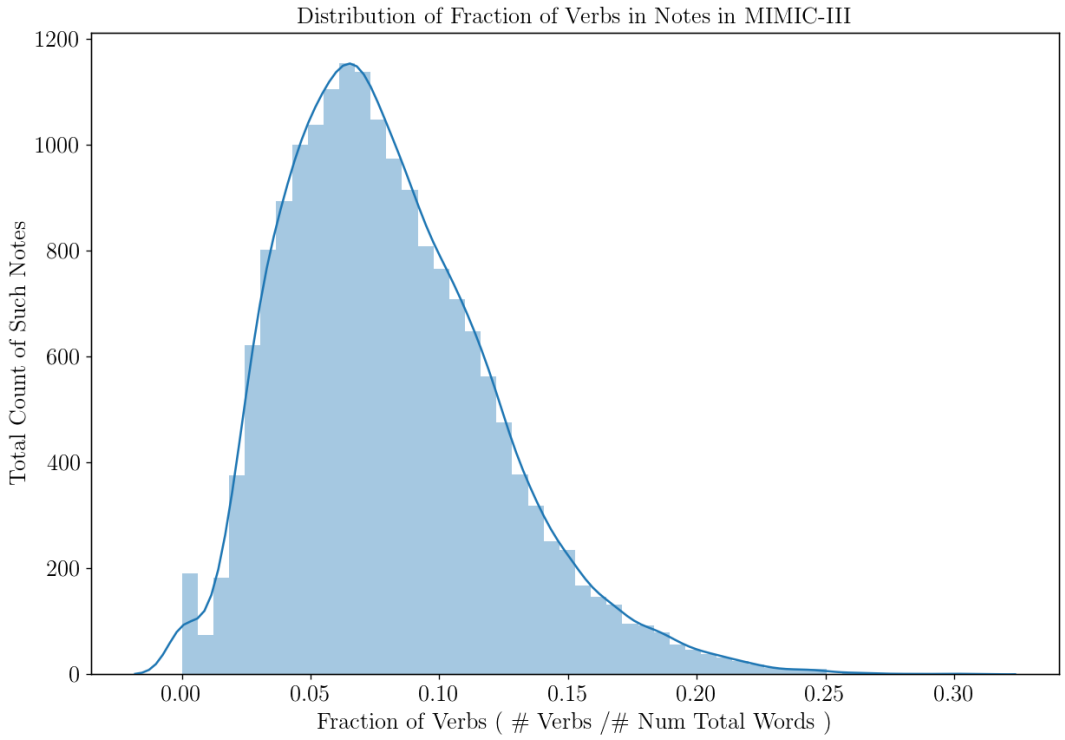


Table 3.7: An example of what the INPUTEVENTS_CV, INPUTEVENTS_MV and OUTPUTEVENTS tables would look like. In INPUTEVENTS_MV, we treat the ENDTIME as CHARTTIME. The values are not taken from MIMIC-III and are falsified. Also, the actual table has more columns than these.

	SUBJECT_ID	HADM_ID	ICUSTAY_ID	ITEMID	CHARTTIME	VALUE	VALUEUOM
0	33	1033	20033	1	3021-12-31 02:30:00	10.40	L/min
1	34	1034	20034	2	3056-09-21 12:30:00	70.28	mmHg
2	34	1035	20035	3	2988-07-13 13:30:00	33.55	NaN
3	35	1036	20036	4	3004-04-15 16:30:00	12.67	mg/dl
4	35	1036	20037	5	3012-05-25 22:30:00	89.99	NaN

many times the patient was admitted in ICU instead. This makes the ICUSTAYS table almost the same size as ADMISSIONS table (not equal because some patients admitted in the hospital may not be admitted in the ICU). After merging all these 3 tables together, we filter out those patients whose age is less than 18 years. We calculate Age from subtraction of ADMITTIME and DOB. We create seperate folders for each subject and then write an admissions.csv file for each subject denoting how many times that patient was admitted to the hospital. The next step in this file is to read all the events table, and remove all the records without an HADM_ID. In case of NOTEEVENTS, we also remove the Discharge summaries (since they mention the final outcome) and all notes that are flagged as error. Generally, ECG and Echo notes are the two types of notes that have a missing CHARTTIME, for which we run additional processing, which includes parsing the note, to get CHARTTIME. Finally, we tokenize all notes. After this is done we divide the events dataframe into smaller dataframes grouped by SUBJECT_ID, and is written as events.csv in the same folder we saved all the admissions.

The extract_subjects.py program leaves us with a total of 38,409 unique patients, 49,172 admissions and a total of 208,572,237 events. Before running the next file, we had to think about what features should our timeseries have. Ideally, the machine learning model would perform best if we have values for all our timeseries. But it is not the case here, so we can minimize the possibility. Therefore, we query all the events and select only those ITEMIDs that occur for a total of more

Figure 3.7: Overview of how we created the in-hospital mortality benchmark dataset in a timeseries format.

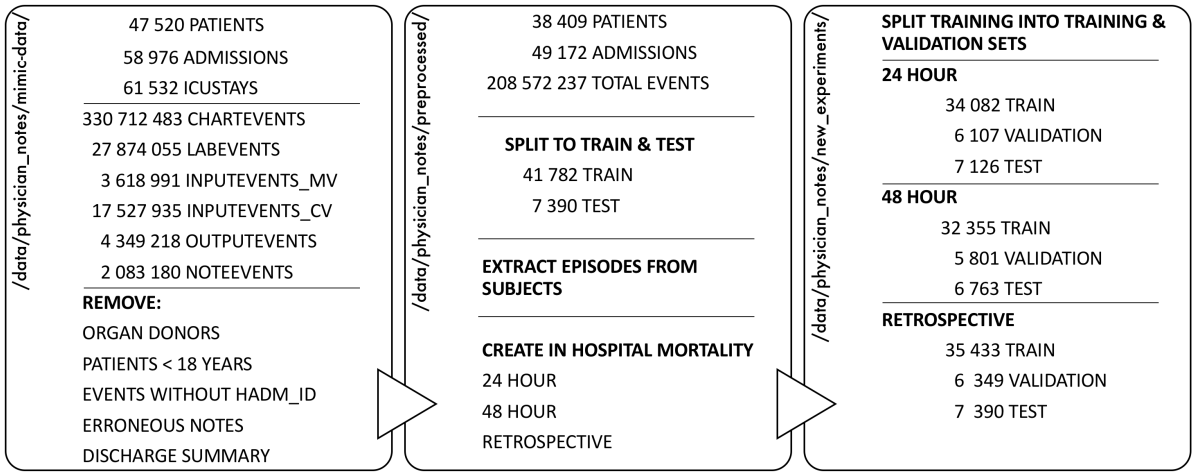


Table 3.8: An example of what the NOTEEVENTS tables would look like. The values are not taken from MIMIC-III and are falsified. Also, the actual table has more columns than these.

	SUBJECT_ID	HADM_ID	CHARTTIME	CATEGORY	TEXT
0	33	1033	3021-12-31 02:30:00	Nursing/other	Sinus bradycardia. Left ante...
1	34	1034	3056-09-21 12:30:00	Nursing	Sinus bradycardia. Left ante...
2	34	1035	2988-07-13 13:30:00	Echo	Sinus rhythm. Left ante...
3	35	1036	3004-04-15 16:30:00	Physician	Sinus rhythm. Left ante...
4	35	1036	3012-05-25 22:30:00	Nursing/other	Sinus bradycardia. Left ante...

than 100,000 times. We make a sepeate list of these variables which we use for our second program. After execution, we end up with 767 ITEMIDs out of a total of 12,487 ITEMIDs.

We now run our second file extract_episodes_from_subjects.py, which would divide each events.csv for each subject on the basis of HADM_ID and then pivot the table on CHARTTIME and selected ITEMID from the list to form the timeseries. In the end, we replace the CHARTTIME with elapsed time in Hours, which we calculate as the subtraction of CHARTTIME and ADMITTIME, and sort the table rows by hours. For a seperate admission, a seperate episode1_timeseries.csv is created which would have the timeseries for that admission and episode1_outcomes.csv is created which would have the length of stay and the mortality flag there.

After this is done, we split all of these subjects into train and test folders, the split specification of which we determine already seperately and use it for all our tasks.

Finally, we create our benchmark dataset with the create_in_hospital_mortality.py which will only keep the timestamps according to the period length parameter we provide as input. We make 3 benchmark datasets, one for 24 hour, one for 48 hour and one retrospective in which we include everything. If a patient dies before the period length, we do not include it in the benchmark dataset (not applicable for retrospective experiment).

3.4 CareVue & MetaVision

We leave the end of the chapter to specify the distinction between CareVue and MetaVision. CareVue was developed by Philips and was used as the information system from 2001 to 2008. The drawback of the CareVue dataset is that it unorganized, and there needs to be a lot of preprocessing beforehand, especially in INPUTEVENTS, to include it in the MIMIC-III database. When we analysed the fraction of patients having physician notes in CareVue system, we got only 3.57%. MetaVision was developed by iMDSoft and was used as the information system from 2008 to 2012. The data from MetaVision was much more organized and the fraction of patients having physician notes was a 43.07%. Because we think that the lack of data, and lack of organized data by CareVue is a result of a legacy issue, we include all of its values for our experiments, and hence do not make any distinction between data from CareVue and from MetaVision. This also allows us to have more generalized approach towards our task.

Table 3.9: Count of Categories in NOTEEVENTS table.

CATEGORY	TOTAL COUNT
Nursing/other	822,497
Radiology	522,279
Nursing	223,556
ECG	209,051
Physician	141,624
Discharge summary	59,652
Echo	45,794
Respiratory	31,739
Nutrition	9,418
General	8,301
Rehab Services	5,431
Social Work	2,670
Case Management	967
Pharmacy	103
Consult	98

Table 3.10: Word Level Analysis of different categories of notes. All the values in the table represents the average count of a particular type of word in the notes. We removed the discharge summaries and erroneous notes from consideration. All values have been rounded off to the nearest integer.

CATEGORY	Words	English	Medical	Nouns	Proper Nouns	Adjectives	Verbs
Nursing/other	202	61	34	38	25	11	19
Radiology	498	96	71	237	28	27	20
Nursing	410	134	84	79	41	23	37
ECG	46	16	14	10	2	6	4
Physician	1582	285	196	215	231	63	79
Echo	493	131	138	103	46	60	40
Respiratory	249	52	48	46	51	12	16
Nutrition	717	114	73	83	95	20	31
General	364	105	64	58	43	20	30
Rehab Services	718	216	125	126	92	40	58
Social Work	530	231	78	79	43	29	63
Case Management	285	87	34	39	36	10	22
Pharmacy	553	205	106	98	43	29	56
Consult	1328	300	192	190	183	66	79

Chapter 4

Testing our Hypotheses - The Task of Mortality Prediction

4.1 Prediction Setup - Evaluation Metrics

To test all of our hypotheses, we use two main evaluation metrics -

- **ROC AUC Score -** An AUC ROC [Fawcett, 2006] curve is a kind of performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. Therefore, if the AUC is higher, that means the model is good at distinguishing between patients who would die and who would not die in-hospital in the given period length.

  If AUC = 1, then the model is said to have an ideal measure of seperability. If AUC=0.7, then that means that there is a partial overlap in the ROC curves of the positive and negative classes and the model has 70% chance that it would be able to correctly classify between the positive and negative class. The worst situation is when AUC = 0.5, which means the ROC curves for positive and negative classes completely overlap each other.

  We use this metric because we are interested in getting the probability of the predicted label as 1, and not get the actual predicted label itself, so that we can get an ROC characteristic and we can quickly calculate the AUC to see how the model performs.

- **Mortality @ K -** The other metric we want to use is the Mortality @ K, or more commonly

known as Precision @ K [Manning et al., 2008]. Mortality @ K suggests that out of the first K values returned as the predicted class, what fraction of the K values actually belong to the predicted class. Since for every episode data, we calculate the probability that the patient would die, i.e., y = 1. When we get all of these probability predictions, we sort them in descending order and then according to the value of K, we check how many of those values actually have y = 1. For the purpose of evaluation here, we use K to be 10, 50, 100, 500 and 1000.

4.2 Notes provide additional predictive power to structured data

As defined in Chapter 3, we pick our structured variables on the basis of their overall count and pick only the 767 which have a count of more than 100,000. Therefore, we decide to put out structured variables to the test and compare them with two baselines.

The first one is from Ghassemi et al. [2014] which runs an admission based model consisting of 3 features - Patient Age, Gender and the admission SAPS-II score. We modify it to have 6 features - Age, Gender, First, Last, Maximum, and Minimum SAPS-II scores - to get a more stronger baseline. These features from the data, split in the exact same specification as described in Section 3.3, is run with a Linear SVM.

The second baseline is from Harutyunyan et al. [2017] which runs a simple logistic regression model after obtaining six different statistical features on 7 different subsequences of the timeseries, in total returning 17*6*7 = 714 features per time series. Then they impute all the missing values and use feature scaling before giving the timeseries as input to the logistic regression classifier. Since our approach is very similar barring the feature extraction part, we do not discuss the details of this implementation here and would rather explain our implementation.

We also follow on the Harutyunyan et al. [2017] method but use different sets of features for our implementation. Since a lot of features that we selected are categorical in nature, we first list all of those features manually in a JSON dictionary to convert them into integers, before we start with the implementation. If there is an exception in the mapping or a number is not well defined,

we just input 0 as the value of that particular feature. Then we use 7 different sub-sequences in this timeseries. These sub-sequences are -

(1) Entire timeseries

(2) First 10% of the timeseries

(3) First 25% of the timeseries

(4) First 50% of the timeseries

(5) Last 10% of the timeseries

(6) Last 25% of the timeseries

(7) Last 50% of the timeseries

We then use 6 different statistical features from these sub-sequences - maximum, minimum, mean, standard deviation, skew and number of measurements. So as a result, we get a total of 767*7*6 = 32,214 features. Then we impute these features and after that use a standard feature scaling. We then use a logistic regression classifier with the class_weights argument[1] as 'balanced', since the mortality proportion is unequal.

When we are using notes, we make some slight modifications. We concatenate all notes from a single timeseries and then use the Bag-of-Words method to generate the count vector. We concatenate this count vector with the output of the final feature scaler (after feature extraction and imputation) with the help of feature union. We then finally use logistic regression again to get the final predicted probability of the mortality label. If we decide to use notes alone, we can skip the concatenation part after using bag-of-words and directly use the logistic regression classifier.

In this thesis, we run the experiments of 24 hour, 48 hour and retrospective mortality with all different sets of features -

- Structured features alone

[1] scikit-learn.org

- Note features alone
- Both Structured and Note features

So, there are two things that we want to find from the main hypothesis. First, is whether our structured variables perform better than those in the baseline, and second, do notes provide additional predictive power to the structured variables.

For the 24 hour mortality prediction task, the Ghassemi et al. [2014] and Harutyunyan et al. [2017] ROC AUC scores are the same upto the third decimal. As far as our structured variables are concerned, we see a major improvement of the ROC AUC score by more than 0.7 units. The Ghassemi et al. [2014] paper[2] gets a higher Mortality @ K socre than the Harutyunyan et al. [2017] paper for all values of K and does better than our approach for K = 10, but then sharply declines while our scores improves, indicating the robustness of our model compared to the earlier baseline. When we bring in notes, however, they perform almost as good as the baseline structured variables and have a much better Mortality @ K score, but do not perform as good as our structured variables. When we combine the notes and structured variables together, we see a marginal improvement in the ROC AUC score and the Mortality @ K is also improved.

For the 48 hour mortality prediction task, the Ghassemi et al. [2014] approach performs worse than what it did previously for the 24 hour mortality prediction task. The ROC AUC score for the Harutyunyan et al. [2017] paper improves by 0.3 units, while the Mortality @ K score deteriorates further from the 24 hour task. Our structured variables remain more robust, although the ROC AUC score improves by only 0.1 unit, still the Mortality @ K score also increases considerably. When we use notes, They perform better than the Ghassemi et al. [2014] paper but worse than the Harutyunyan et al. [2017] paper. But it is able to beat both the approaches as far as Mortality @ K is concerned. When we combine the notes and structured variables, we see a similar result as we got for the 24 hour prediction, with both the ROC AUC scores and Mortality @ K showing marginal improvements.

[2] Note that this does not have a timeseries aspect, but we keep those instances that are also there in the other two model inputs, to ensure fairness.

For the retrospective mortality prediction task, the Ghassemi et al. [2014] approach performs almost the same as it did for the earlier two setups (since there is no notion of time in this setup). The ROC AUC score of the Harutyunyan et al. [2017] shows considerable amount of improvement in the ROC AUC score, getting 0.96 units. The mortality @ K score also improves considerably well. Our structured variables perform even better than this approach and get an ROC AUC score of 0.98 with a slightly better Mortality @ K, with changes only reflecting when K = 1000. When we use notes, though we get an ROC AUC score of 0.93, we get a much better Mortality @ K, getting 1 for both K = 10 and K = 50. Combining the notes and structured variables show only marginal improvements, in both ROC AUC score and Mortality @ K.

Therefore, after looking at these prediction outcomes, we can say that our structured variables are more robust and perform better than the baseline approaches for all the three prediction setups. On the other hand, we also see that notes do provide additional predictive power to the structured variables, though that improvement is very marginal.

Table 4.1: Results of overall 24 Hour mortality prediction.

	Ghassemi et al. (2014)	Harutyunyan et al. (2018)	Structured Variables	Notes	Structured Variables + Notes
ROC AUC Score	0.767	0.767	0.834	0.765	0.838
Mortality @ K = 10	0.800	0.500	0.600	0.900	0.700
Mortality @ K = 50	0.620	0.580	0.860	0.540	0.800
Mortality @ K = 100	0.620	0.550	0.760	0.430	0.790
Mortality @ K = 500	0.404	0.408	0.474	0.360	0.484
Mortality @ K = 1000	0.293	0.302	0.384	0.302	0.388

4.3 Certain parts of notes are more valuable than other parts

In chapter 2, we gave this hypothesis in order to find what are valuable inside of notes and what are not. To test this, we use different subsets of notes and see what subsets perform better with our in-hospital mortality prediction task.

Table 4.2: Results of overall 48 Hour mortality prediction.

	Ghassemi et al. (2014)	Harutyunyan et al. (2018)	Structured Variables	Notes	Structured Variables + Notes
ROC AUC Score	0.754	0.797	0.846	0.781	0.848
Mortality @ K = 10	0.900	0.300	0.800	0.600	0.900
Mortality @ K = 50	0.540	0.540	0.820	0.440	0.860
Mortality @ K = 100	0.560	0.530	0.720	0.410	0.750
Mortality @ K = 500	0.364	0.384	0.462	0.348	0.472
Mortality @ K = 1000	0.259	0.307	0.348	0.280	0.350

A. Physician notes are more useful compared to the other categories of notes. To test this, we do not concatenate all the notes together. Instead we concatenate notes from each category together. So for each category, we use a seperate logistic regression classifier. we again use 14 different logistic regression classifiers which would have the same structured variables (or not if we decide to use notes alone), but different Bag-of-words features.

As we can see from the experiment results, all different categories of notes when combined with structured variables are used for prediction, there is only a marginal difference in the ROC AUC scores and Mortality @ K scores for all different categories of notes. However, once we remove structured variables and perform the prediciton task on categories of notes alone, we find that Nursing/other notes perform the best when it comes to both the ROC AUC score and the Mortality @ K score. The Radiology notes come at a close second when we talk about ROC AUC scores alone, but they do not get a comparable Mortality @ K score. The Physician notes also do not show a comparable result in both the ROC AUC and Mortality @ K scores for any of the time periods in the task. Therefore, we can say that Physician notes are not the most useful or valuable category of notes.

B. The common english words in notes are less useful compared to the uncommon english words (which also includes most medical words). To test this, we first concatenate all the notes together and then filter all the words out which belong to the google top 10,000 english

Table 4.3: Results of overall Retrospective mortality prediction.

	Ghassemi et al. (2014)	Harutyunyan et al. (2018)	Structured Variables	Notes	Structured Variables + Notes
ROC AUC Score	0.778	0.968	0.980	0.930	0.981
Mortality @ K = 10	0.800	0.900	0.900	1.000	1.000
Mortality @ K = 50	0.800	0.960	0.960	1.000	0.980
Mortality @ K = 100	0.640	0.970	0.980	0.940	0.990
Mortality @ K = 500	0.468	0.978	0.978	0.804	0.980
Mortality @ K = 1000	0.348	0.736	0.768	0.601	0.768

word list[3] . That note would act as the notes with common english words and the words that did not belong to the top 10,000 english words would act note with the uncommon english words. We think that most of the medical domain words are included as a part of notes with uncommon english words. We then use two seperate logistic regression classifiers to test which type of note evaluates better.

As we can see from the experiment results, the common english words in notes perform marginally better in terms of the ROC AUC score than the uncommon english words (which may also include a lot of medical domain words) in notes if we also include the structured variables. If we remove structured variables, the common english words in notes seem to perform much better than the uncommon english words in notes in terms of ROC AUC Scores alone. For both of these subsets, Mortality @ K scores are most of the time same or within a comparable range. Therefore, we can say, to our surprise, that common english words in notes perform better at predicting mortality than the uncommon english words in notes.

C. The nouns in notes are more useful compared to both adjectives and verbs. To test this hypothesis, we use spacy[4] to get the part-of-speech tags of every word in the concatenated note and then filter out 4 seperate notes, one with only nouns, one with only proper nouns, one with adjectives and one with verbs. We then use the same logistic regression model to test which

[3] https://github.com/first20hours/google-10000-english
[4] spacy.io

kind of note would evaluate better.

As we can see from the experiment results, Nouns and Proper Nouns perform almost comparable to each other, while performing better than Adjectives and Verbs as far as the ROC AUC scores are concerned. When we include the structured variables, all the scores are almost comparable but when we remove structured variables, Nouns seem to perform the best when it comes to both the ROC AUC score and Mortality @ K score. Proper Nouns perform almost as good as Nouns when we look at the ROC AUC scores, but do not compare when we look at Mortality @ K. Adjectives perform a little in the intermediary for both the ROC AUC scores and Mortality @ K scores. The Verbs perform the poorest of the four, getting the lowest ROC AUC score for all different period lengths. Therefore, we can say that Nouns perform the best and are the most useful in notes compared to Adjectives and Verbs.

D. The notes that are similar to previous notes (what we term as "copy-pasted") are less useful compared to notes that less similar, i.e, are original and concise. To test this hypothesis, we do a sentence level tokenization of every note. Then for every sentence in the note, we calculate the cosine similarity of that sentence with all the sentences in all the notes written before it and pick the maximum similarity as the similarity of the sentence. After this is done, we calculate the similarity of the note by averaging the similarities of all the sentences in the notes. We consider the similarity of the first note in the timeseries to be 0. Then for all the notes, once we get the similarity, we sort them and find the first and third quantiles. All notes having similarity greater than first quantile would be considered as "copy-pasted" notes, and the notes having similarity less than the third quantile would be considered as original notes. We then train seperate logistic regression classifiers for these two kinds of notes and report the results.

As we can see from the experiment results, notes that are not "copy-pasted", or are less similar to the previous notes, perform better in terms of both the ROC AUC score as well as the Mortality @ K score. Therefore, we can say that notes that are more similar to previous notes (or say, are "copy-pasted") have less value/predictive power than the notes that are less similar, i.e., are original and concise.

Table 4.4: Results of 24 Hour Mortality Prediction - Subset of Category with Structured Variables.

	All Notes	Nursing / Other	Physician	Nursing	Radiology	Echo
ROC AUC Score	0.838	0.825	0.832	0.835	0.831	0.833
Mortality @ K = 10	0.700	0.700	0.700	0.700	0.600	0.600
Mortality @ K = 50	0.800	0.840	0.820	0.820	0.840	0.840
Mortality @ K = 100	0.790	0.770	0.770	0.770	0.820	0.780
Mortality @ K = 500	0.484	0.466	0.466	0.476	0.466	0.468
Mortality @ K = 1000	0.388	0.366	0.380	0.376	0.373	0.377

Table 4.5: Results of 24 Hour Mortality Prediction - Subset of Category without Structured Variables.

	All Notes	Nursing / Other	Physician	Nursing	Radiology	Echo
ROC AUC Score	0.765	0.677	0.550	0.567	0.674	0.545
Mortality @ K = 10	0.900	0.900	0.400	0.700	0.100	0.300
Mortality @ K = 50	0.540	0.740	0.400	0.580	0.280	0.320
Mortality @ K = 100	0.430	0.620	0.380	0.430	0.310	0.250
Mortality @ K = 500	0.360	0.376	0.146	0.188	0.276	0.156
Mortality @ K = 1000	0.302	0.242	0.134	0.146	0.247	0.128

Table 4.6: Results of 48 Hour Mortality Prediction - Subset of Category with Structured Variables.

	All Notes	Nursing / Other	Physician	Nursing	Radiology	Echo
ROC AUC Score	0.848	0.831	0.844	0.846	0.841	0.845
Mortality @ K = 10	0.900	0.900	0.900	0.900	0.900	0.900
Mortality @ K = 50	0.860	0.820	0.780	0.800	0.800	0.820
Mortality @ K = 100	0.750	0.750	0.720	0.730	0.720	0.730
Mortality @ K = 500	0.472	0.436	0.464	0.468	0.458	0.454
Mortality @ K = 1000	0.350	0.341	0.349	0.349	0.344	0.349

Table 4.7: Results of 48 Hour Mortality Prediction - Subset of Category without Structured Variables.

	All Notes	Nursing / Other	Physician	Nursing	Radiology	Echo
ROC AUC Score	0.781	0.697	0.552	0.572	0.657	0.557
Mortality @ K = 10	0.600	0.600	0.400	0.700	0.000	0.100
Mortality @ K = 50	0.440	0.800	0.400	0.500	0.260	0.160
Mortality @ K = 100	0.410	0.630	0.290	0.360	0.240	0.210
Mortality @ K = 500	0.348	0.344	0.132	0.162	0.260	0.152
Mortality @ K = 1000	0.280	0.224	0.116	0.134	0.174	0.122

Table 4.8: Results of Retrospective Mortality Prediction - Subset of Category with Structured Variables.

	All Notes	Nursing / Other	Physician	Nursing	Radiology	Echo
ROC AUC Score	0.981	0.981	0.981	0.981	0.981	0.981
Mortality @ K = 10	1.000	1.000	1.000	1.000	1.000	1.000
Mortality @ K = 50	0.980	0.980	0.980	0.980	0.980	0.980
Mortality @ K = 100	0.990	0.990	0.990	0.990	0.990	0.990
Mortality @ K = 500	0.980	0.792	0.980	0.980	0.980	0.976
Mortality @ K = 1000	0.768	0.768	0.768	0.767	0.768	0.765

Table 4.9: Results of Retrospective Mortality Prediction - Subset of Category without Structured Variables.

	All Notes	Nursing / Other	Physician	Nursing	Radiology	Echo
ROC AUC Score	0.930	0.844	0.611	0.627	0.769	0.600
Mortality @ K = 10	1.000	1.000	0.900	1.000	0.600	0.300
Mortality @ K = 50	1.000	1.000	0.880	0.860	0.600	0.300
Mortality @ K = 100	0.940	1.000	0.720	0.840	0.600	0.340
Mortality @ K = 500	0.804	0.778	0.270	0.292	0.434	0.250
Mortality @ K = 1000	0.601	0.465	0.202	0.212	0.277	0.202

Table 4.10: Results of 24 Hour Mortality Prediction - Subset of Language with Structured Variables.

	All Notes	Common English	Uncommon English (including Medical)
ROC AUC Score	0.838	0.838	0.836
Mortality @ K = 10	0.700	0.700	0.700
Mortality @ K = 50	0.800	0.840	0.840
Mortality @ K = 100	0.790	0.760	0.780
Mortality @ K = 500	0.484	0.480	0.472
Mortality @ K = 1000	0.388	0.386	0.386

Table 4.11: Results of 24 Hour Mortality Prediction - Subset of Language without Structured Variables.

	All Notes	Common English	Uncommon English (including Medical)
ROC AUC Score	0.765	0.740	0.717
Mortality @ K = 10	0.900	0.400	0.400
Mortality @ K = 50	0.540	0.400	0.260
Mortality @ K = 100	0.430	0.380	0.220
Mortality @ K = 500	0.360	0.310	0.202
Mortality @ K = 1000	0.302	0.266	0.235

Table 4.12: Results of 48 Hour Mortality Prediction - Subset of Language with Structured Variables.

	All Notes	Common English	Uncommon English (including Medical)
ROC AUC Score	0.848	0.847	0.847
Mortality @ K = 10	0.900	0.900	0.900
Mortality @ K = 50	0.860	0.820	0.820
Mortality @ K = 100	0.750	0.750	0.730
Mortality @ K = 500	0.472	0.470	0.464
Mortality @ K = 1000	0.350	0.355	0.350

Table 4.13: Results of 48 Hour Mortality Prediction - Subset of Language without Structured Variables.

	All Notes	Common English	Uncommon English (including Medical)
ROC AUC Score	0.781	0.766	0.715
Mortality @ K = 10	0.600	0.400	0.300
Mortality @ K = 50	0.440	0.320	0.240
Mortality @ K = 100	0.410	0.360	0.220
Mortality @ K = 500	0.348	0.308	0.190
Mortality @ K = 1000	0.280	0.258	0.183

Table 4.14: Results of Retrospective Mortality Prediction - Subset of Language with Structured Variables.

	All Notes	Common English	Uncommon English (including Medical)
ROC AUC Score	0.981	0.982	0.981
Mortality @ K = 10	1.000	1.000	1.000
Mortality @ K = 50	0.980	0.980	0.980
Mortality @ K = 100	0.990	0.990	0.990
Mortality @ K = 500	0.980	0.980	0.980
Mortality @ K = 1000	0.768	0.768	0.768

Table 4.15: Results of Retrospective Mortality Prediction - Subset of Language without Structured Variables.

	All Notes	Common English	Uncommon English (including Medical)
ROC AUC Score	0.930	0.900	0.858
Mortality @ K = 10	1.000	1.000	1.000
Mortality @ K = 50	1.000	0.820	0.900
Mortality @ K = 100	0.940	0.760	0.780
Mortality @ K = 500	0.804	0.678	0.450
Mortality @ K = 1000	0.601	0.525	0.396

Table 4.16: Results of 24 Hour Mortality Prediction - Subset of Part-of-Speech Tag with Structured Variables.

	All Notes	Noun	Proper Noun	Adjective	Verb
ROC AUC Score	0.838	0.836	0.837	0.836	0.836
Mortality @ K = 10	0.700	0.700	0.700	0.700	0.700
Mortality @ K = 50	0.800	0.840	0.840	0.840	0.860
Mortality @ K = 100	0.790	0.790	0.780	0.810	0.790
Mortality @ K = 500	0.484	0.466	0.472	0.468	0.472
Mortality @ K = 1000	0.388	0.381	0.385	0.382	0.378

Table 4.17: Results of 24 Hour Mortality Prediction - Subset of Part-of-Speech Tag without Structured Variables.

	All Notes	Noun	Proper Noun	Adjective	Verb
ROC AUC Score	0.765	0.703	0.705	0.676	0.645
Mortality @ K = 10	0.900	0.700	0.200	0.100	0.400
Mortality @ K = 50	0.540	0.380	0.320	0.180	0.320
Mortality @ K = 100	0.430	0.310	0.310	0.210	0.310
Mortality @ K = 500	0.360	0.170	0.240	0.228	0.182
Mortality @ K = 1000	0.302	0.203	0.214	0.209	0.178

Table 4.18: Results of 48 Hour Mortality Prediction - Subset of Part-of-Speech Tag with Structured Variables.

	All Notes	Noun	Proper Noun	Adjective	Verb
ROC AUC Score	0.848	0.847	0.849	0.848	0.848
Mortality @ K = 10	0.900	0.900	0.900	0.900	0.900
Mortality @ K = 50	0.860	0.820	0.820	0.820	0.820
Mortality @ K = 100	0.750	0.740	0.760	0.740	0.740
Mortality @ K = 500	0.472	0.468	0.462	0.466	0.460
Mortality @ K = 1000	0.350	0.350	0.350	0.349	0.348

Table 4.19: Results of 48 Hour Mortality Prediction - Subset of Part-of-Speech Tag without Structured Variables.

	All Notes	Noun	Proper Noun	Adjective	Verb
ROC AUC Score	0.781	0.735	0.724	0.704	0.677
Mortality @ K = 10	0.600	0.700	0.600	0.400	0.200
Mortality @ K = 50	0.440	0.300	0.380	0.360	0.280
Mortality @ K = 100	0.410	0.250	0.300	0.290	0.250
Mortality @ K = 500	0.348	0.186	0.222	0.236	0.200
Mortality @ K = 1000	0.280	0.191	0.203	0.195	0.185

Table 4.20: Results of Retrospective Mortality Prediction - Subset of Part-of-Speech Tag with Structured Variables.

	All Notes	Noun	Proper Noun	Adjective	Verb
ROC AUC Score	0.981	0.981	0.981	0.981	0.981
Mortality @ K = 10	1.000	1.000	1.000	1.000	1.000
Mortality @ K = 50	0.980	0.980	0.980	0.980	0.980
Mortality @ K = 100	0.990	0.990	0.990	0.990	0.990
Mortality @ K = 500	0.980	0.980	0.980	0.978	0.978
Mortality @ K = 1000	0.768	0.768	0.767	0.767	0.767

Table 4.21: Results of Retrospective Mortality Prediction - Subset of Part-of-Speech Tag without Structured Variables.

	All Notes	Noun	Proper Noun	Adjective	Verb
ROC AUC Score	0.930	0.868	0.861	0.780	0.822
Mortality @ K = 10	1.000	1.000	0.800	0.700	0.900
Mortality @ K = 50	1.000	0.940	0.760	0.480	0.820
Mortality @ K = 100	0.940	0.840	0.680	0.470	0.790
Mortality @ K = 500	0.804	0.452	0.546	0.386	0.556
Mortality @ K = 1000	0.601	0.416	0.431	0.315	0.431

Table 4.22: Results of 24 Hour Mortality Prediction - Subset of "Copy Pasting" with Structured Variables.

	All Notes	"Copy-Pasted" (similar to previous notes)	Original (not similar to previous notes)
ROC AUC Score	0.838	0.835	0.837
Mortality @ K = 10	0.700	0.700	0.700
Mortality @ K = 50	0.800	0.820	0.840
Mortality @ K = 100	0.790	0.780	0.790
Mortality @ K = 500	0.484	0.484	0.476
Mortality @ K = 1000	0.388	0.384	0.391

Table 4.23: Results of 24 Hour Mortality Prediction - Subset of "Copy Pasting" without Structured Variables.

	All Notes	"Copy-Pasted" (similar to previous notes)	Original (not similar to previous notes)
ROC AUC Score	0.765	0.748	0.762
Mortality @ K = 10	0.900	0.900	0.900
Mortality @ K = 50	0.540	0.580	0.520
Mortality @ K = 100	0.430	0.510	0.460
Mortality @ K = 500	0.360	0.378	0.374
Mortality @ K = 1000	0.302	0.292	0.304

Table 4.24: Results of 48 Hour Mortality Prediction - Subset of "Copy Pasting" with Structured Variables.

	All Notes	"Copy-Pasted" (similar to previous notes)	Original (not similar to previous notes)
ROC AUC Score	0.848	0.844	0.847
Mortality @ K = 10	0.900	0.900	0.900
Mortality @ K = 50	0.860	0.840	0.820
Mortality @ K = 100	0.750	0.740	0.730
Mortality @ K = 500	0.472	0.468	0.466
Mortality @ K = 1000	0.350	0.351	0.349

Table 4.25: Results of 48 Hour Mortality Prediction - Subset of "Copy Pasting" without Structured Variables.

	All Notes	"Copy-Pasted" (similar to previous notes)	Original (not similar to previous notes)
ROC AUC Score	0.781	0.765	0.782
Mortality @ K = 10	0.600	0.700	0.600
Mortality @ K = 50	0.440	0.480	0.520
Mortality @ K = 100	0.410	0.460	0.460
Mortality @ K = 500	0.348	0.354	0.354
Mortality @ K = 1000	0.280	0.260	0.282

Table 4.26: Results of Retrospective Mortality Prediction - Subset of "Copy Pasting" with Structured Variables.

	All Notes	"Copy-Pasted" (similar to previous notes)	Original (not similar to previous notes)
ROC AUC Score	0.981	0.981	0.981
Mortality @ K = 10	1.000	1.000	1.000
Mortality @ K = 50	0.980	0.980	0.980
Mortality @ K = 100	0.990	0.990	0.990
Mortality @ K = 500	0.980	0.980	0.980
Mortality @ K = 1000	0.768	0.767	0.768

Table 4.27: Results of Retrospective Mortality Prediction - Subset of "Copy Pasting" without Structured Variables.

	All Notes	"Copy-Pasted" (similar to previous notes)	Original (not similar to previous notes)
ROC AUC Score	0.930	0.886	0.927
Mortality @ K = 10	1.000	1.000	1.000
Mortality @ K = 50	1.000	0.900	1.000
Mortality @ K = 100	0.940	0.890	0.930
Mortality @ K = 500	0.804	0.742	0.790
Mortality @ K = 1000	0.601	0.511	0.597

Chapter 5

Conclusion and Future Work

5.1 Conclusion

As we saw from this work, we started with the motivation to help doctors write better notes in Electronic Health Records, because of the various challenges they face, like burnout of doctors due to information overloading. We also saw the benefits that good notes provide, like maintaining continuity in healthcare, enabling patient engagement, and patient activation. We then saw how machine learning has been widely used in electronic health records, since doctors cannot write better notes with human assistance, and they do need machine assistance for it. Because of the various machine learning applications available, we saw that we can use it to assist doctors to write better notes, and so we need to learn how to help doctors to do that. So it is important to know what parts of the notes are important and which ones are not, because if we can find what parts of notes are useful to the doctors, we can help save a lot of the doctor's time and provide them with the relevant information to write clean and good notes. Therefore, we were motivated from this thought process to study and understand the value of notes in EHRs.

According to recent literature, a lot of the machine learning focus on EHRs is the task of mortality prediction. To get baselines easily and compare our approach with existing approaches, we decide to use this task of in-hospital mortality prediction and see if notes provide value to effective prediction for this task. Concretely, we use the MIMIC-III dataset, and understand its Patient related and Hospital related tables, after which we use all the patients and events tables, filter certain records, to form timeseries out of this data. Once we have gotten that, we use the

timeseries to generate 3 benchmark datasets - one for predicting mortality after 24 hours of a patient's admission, one for predicting mortality after 48 hours of a patient's admission, and one retrospective mortality prediction, where we take all of the data associated with a patient's stay, and then try to predict whether the patient died or not.

We pick up two baseline approaches, one using a Linear SVM with 6 features and a Logistic Regression with 712 features. Since all of these features were from structured variables, we decided to compare our method's structured variables (with 32,214 features) with the previous two approaches. Additionaly, we compare the performace of a seperate bag-of-words classifier obtained from the notes, with the structured variables and also see if the combining the structured variables and the bag-of-words features from notes improve the mortality prediction. We also try the experiements on various subsets of notes, and see if some parts of notes are better at predicting mortality than other parts. We report all the predcition scores with an ROC AUC score, and a Mortality @ K score (which is just a Prediction @ K score).

After obtaining all the results, we find that structured variables outperform both the baseline approaches by a big margin, in terms of both the ROC AUC score and the Mortality @ K score, for all the three prediction tasks. We then test our primary hypothesis, and learn that notes add a marginal predictive power to the structured variables, for all the three task setups. But in general, notes do get a good Mortality @ K score and perform almost as good as the structured baselines. When we test our secondary hypothesis, and find that Nursing/other notes are the most useful in all categories of notes, likely driven by their volume, but unlike our hypothesis that Physician notes would be the most useful. We then see that the common english words in notes are more useful than the uncommon english words for all three prediction tasks, with the changes reflecting mostly in the ROC AUC scores, and a comparable difference in the Mortality @ K scores. We futher come to find that nouns in notes are more useful than adjectives or verbs, for all of the prediction tasks. Finally, we also see that notes that are less similar (what we term as "original") to previous notes for a patient's health record are more useful compared to notes that are more similar (what we term as "copy-pasted").

Therefore, notes are indeed valuable and there is much more scope for understanding in detail as to what extent, notes can add additonal value to structured information in Electronic Health Records. Moreover, the identification of valuable parts of notes can help in looking for methods to obtain those valubale parts quickly so as to help doctors write good notes and overcome the challenges.

5.2 Future Work

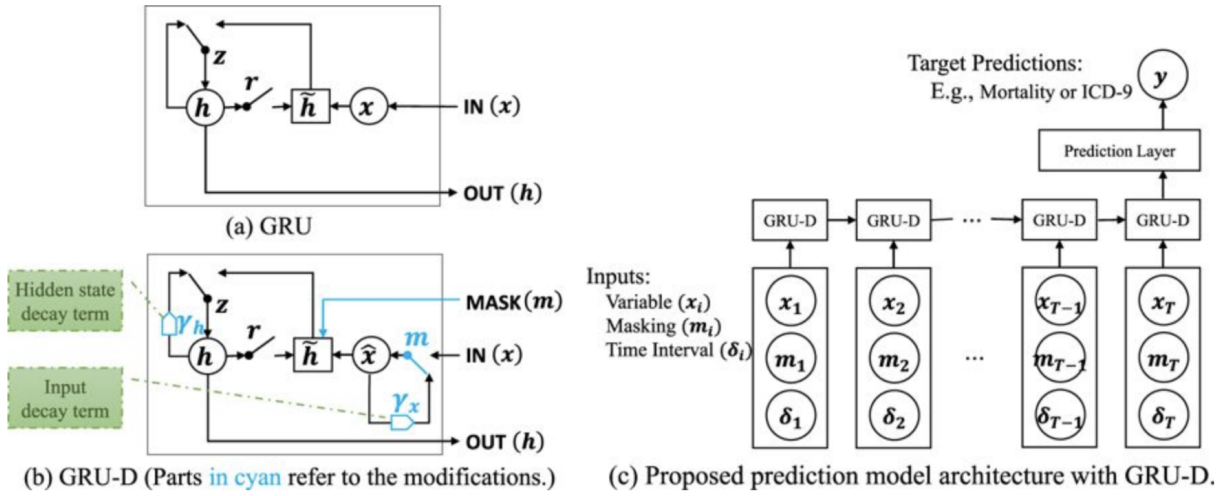
Currently we are using the Bag-of-words approach to concatenate that vector to the structured variables vector. In that process, we lose all the timeseries information associated with the note. To overcome that, we would like to use the model by Che et al. [2018] to try to keep the timeseries information preserved for not only structured variables, but also for notes. As the paper states, missing information is actually very informative, since it relies on two main points -

- If a value has been missing from a long time in the timeseries, the probability that its value is close to the default value is very high.
- If a value has been missing from a long time in the timeseries, that particular variable loses its influence on the final prediction.

Since both of these two points indicate some sort of a decay of features, the authors introduce the model called GRU-D, where D stands for Decay. The GRU-D takes 3 things as input, a value of timeseries, a masking for the timeseries which is 1 if the value is present and 0 if it is not present, and a timedelta matrix, which tells us for how long a particular feature value is missing from the timeseries.

To use the GRU-D for note data, we can use two ways - we can either use of the pre-trained embeddings from one of the methods described in Section 2.3, or train our own embedding, which currently sounds more reasonable, given the restrictions of the dataset. We are still working on this method and will report the results of this experiement in the future.

Figure 5.1: Overview of the GRU-D Model. (a)Normal GRU, (b) Proposed GRU-D, (c) Illustration of GRU-D on our time series data with missing values. Image courtesy of [Che et al., 2018].



Furthermore, we would like to differentiate our experiments for CareVue and MetaVision patients seperately and see if we do better with MetaVision patients, since they have a lot more fraction of notes, especially Physician notes. We would also like to try all the methods that we introduced on a bigger dataset than MIMIC-III, and since that dataset would have more notes, it would be interesting to see how our model behaves and see using bigger datasets reinforce the fact about the usefulness of notes.

Bibliography

Saps ii, Sep 2018. URL https://en.wikipedia.org/wiki/SAPS_II.

Julia Adler-Milstein and David W. Bates. Paperless healthcare: Progress and challenges of an it-enabled healthcare system. Business Horizons, 53(2):119 – 130, 2010. ISSN 0007-6813. doi: https://doi.org/10.1016/j.bushor.2009.10.004. URL http://www.sciencedirect.com/science/article/pii/S0007681309001529. Special Issue on Healthcare and the Life Sciences in Transition.

Brett K Beaulieu-Jones, Patryk Orzechowski, and Jason H Moore. Mapping patient trajectories using longitudinal extraction and deep learning in the mimic-iii critical care database. In Pac Symp Biocomput, volume 23, pages 123–32. World Scientific, 2018.

Signal K Bell, Tom Delblanco, and Jan Walker. Opennotes: How the power of knowing can change health care, Nov 2018. URL https://catalyst.nejm.org/opennotes-knowing-change-health-care/.

Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, Jianfeng Gao, Po-Sen Huang, and Yejin Choi. Discourse-aware neural rewards for coherent text generation. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 173–184, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1016. URL https://www.aclweb.org/anthology/N18-1016.

Dallas Card, Chenhao Tan, and Noah A. Smith. Neural models for documents with metadata. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2031–2040, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P18-1189.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. Scientific reports, 8(1):6085, 2018.

Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. Multi-layer representation learning for medical concepts. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1495–1504. ACM, 2016.

Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. Gram: graph-based attention model for healthcare representation learning. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 787–795. ACM, 2017.

Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. Nature Medicine, 25(1):24–29, 1 2019. ISSN 1078-8956. doi: 10.1038/s41591-018-0316-z. URL https://doi.org/10.1038/s41591-018-0316-z.

Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Sandy Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, David Jones, David Silver, Koray Kavukcuoglu, Demis Hassabis, and Andrew Senior. De novo structure prediction with deep-learning based scoring, 12 2018.

Tom Fawcett. An introduction to roc analysis. Pattern recognition letters, 27(8):861–874, 2006.

Katleen Gabriels and Tania Moerenhout. Exploring entertainment medicine and professionalization of self-care: Interview study among doctors on the potential effects of digital self-tracking. J Med Internet Res, 20(1):e10, Jan 2018. ISSN 1438-8871. doi: 10.2196/jmir.8040. URL http://www.jmir.org/2018/1/e10/.

Atul Gawande. Why doctors hate their computers, Nov 2018. URL https://www.newyorker.com/magazine/2018/11/12/why-doctors-hate-their-computers.

Marzyeh Ghassemi, Tristan Naumann, Finale Doshi-Velez, Nicole Brimmer, Rohit Joshi, Anna Rumshisky, and Peter Szolovits. Unfolding physiological state: Mortality modelling in intensive care units. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 75–84. ACM, 2014.

Derek A Haas, John D Halamka, and Michael Suk. 3 ways to make electronic health records less time-consuming for physicians, Jan 2019. URL https://hbr.org/2019/01/3-ways-to-make-electronic-health-records-less-time-consuming-for-physicians.

Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. arXiv preprint arXiv:1703.07771, 2017.

Albert Hoang, Charles Magee, Dorothy A Becher, Laura L Sessums, Louis N Pangaro, Patrick G O'Malley, Harry B Burke, Fang Liu, Paul Fontelo, Adam Saperstein, Christopher W Bunt, Jessica Servey, Mark Stephens, Thomas Miller, Barbara A Cooper, Julie M Chen, Nancy S Baxi, Renee M Mallory, Shawn Kane, II Capaldi, Vincent F, Zizette R Makary, David A Djuric, Joshua A Hodge, and Ronald W Gimbel. Electronic health records improve clinical note quality. Journal of the American Medical Informatics Association, 22(1):199–205, 10 2014. ISSN 1067-5027. doi: 10.1136/amiajnl-2014-002726. URL https://doi.org/10.1136/amiajnl-2014-002726.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, and Yejin Choi. Learning to write by learning the objective, 2018a. URL https://openreview.net/forum?id=r1lfpfZAb.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. Learning to write with cooperative discriminators. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1638–1649,

Melbourne, Australia, July 2018b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P18-1152.

Mohit Iyyer, Varun Manjunatha, Anupam Guha, Yogarshi Vyas, Jordan Boyd-Graber, Hal Daume, and Larry S Davis. The amazing mysteries of the gutter: Drawing inferences between panels in comic book narratives. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7186–7195, 2017.

Julia James. Patient engagement, Feb 2013. URL DOI:10.1377/hpb20130214.898775.

Mengqi Jin, Mohammad Taha Bahadori, Aaron Colak, Parminder Bhatia, Busra Celikkaya, Ram Bhakta, Selvan Senthivel, Mohammed Khalilia, Daniel Navarro, Borui Zhang, et al. Improving hospital mortality prediction with medical named entities and multimodal learning. arXiv preprint arXiv:1811.12276, 2018.

Baoyu Jing, Pengtao Xie, and Eric Xing. On the automatic generation of medical imaging reports. arXiv preprint arXiv:1711.08195, 2017.

Alistair Edward William Johnson, Tom J. Pollard, and Roger G. Mark. Reproducibility in critical care: a mortality prediction case study. In MLHC, 2017a.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. Scientific data, 3:160035, 2016.

Alistair EW Johnson, David J Stone, Leo A Celi, and Tom J Pollard. The mimic code repository: enabling reproducibility in critical care research. Journal of the American Medical Informatics Association, 25(1):32–39, 2017b.

Stephen B Johnson, Suzanne Bakken, Daniel Dine, Sookyung Hyun, Eneida Mendonça, Frances Morrison, Tiffani Bright, Tielman Van Vleck, Jesse Wrenn, and Peter Stetson. An electronic health record based on structured narrative. Journal of the American Medical Informatics Association, 15(1):54–64, 2008.

Patrick Kierkegaard. Electronic health record: Wiring europe's healthcare. Computer Law & Security Review, 27(5):503 – 515, 2011. ISSN 0267-3649. doi: https://doi.org/10.1016/j.clsr.2011.07.013. URL http://www.sciencedirect.com/science/article/pii/S0267364911001257.

Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel. Multimodal neural language models. In Eric P. Xing and Tony Jebara, editors, Proceedings of the 31st International Conference on Machine Learning, volume 32 of Proceedings of Machine Learning Research, pages 595–603, Bejing, China, 22–24 Jun 2014. PMLR. URL http://proceedings.mlr.press/v32/kiros14.html.

Linda T Kohn, Janet Corrigan, Molla S Donaldson, et al. To err is human: building a safer health system, volume 6. National academy press Washington, DC, 2000.

Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. Transfer learning for named-entity recognition with neural networks. arXiv preprint arXiv:1705.06273, 2017.

Zachary Chase Lipton, David C. Kale, Charles Elkan, and Randall C. Wetzel. Learning to diagnose with lstm recurrent neural networks. CoRR, abs/1511.03677, 2016.

Jingshu Liu, Zachariah Zhang, and Narges Razavian. Deep ehr: Chronic disease prediction using medical notes. arXiv preprint arXiv:1808.04928, 2018a.

Peter J Liu. Learning to write notes in electronic health records. arXiv preprint arXiv:1808.02622, 2018.

Xiangan Liu, Keyang Xu, Pengtao Xie, and Eric Xing. Unsupervised pseudo-labeling for extractive summarization on electronic health records. arXiv preprint arXiv:1811.08040, 2018b.

Brian Lubars and Chenhao Tan. Ask not what ai can do, but what ai should do: Towards a framework of task delegability. 2019.

Yuan Luo. Recurrent neural networks for classifying relations in clinical notes. Journal of biomedical informatics, 72:85–95, 2017.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval, volume 1. Cambridge University Press Cambridge, 2008.

Goku Mohandas. Deep learning with electronic health record (EHR) systems, Sep 2018. URL `https://goku.me/blog/EHR`.

Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. Benchmarking deep learning models on large healthcare datasets. Journal of Biomedical Informatics, 83:112 – 134, 2018. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2018.04.007. URL `http://www.sciencedirect.com/science/article/pii/S1532046418300716`.

Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Michaela Hardt, Peter J Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, et al. Scalable and accurate deep learning with electronic health records. NPJ Digital Medicine, 1(1):18, 2018.

Alvin Rajkomar, Jeffrey Dean, and Issac Kohane. Machine learning in medicine — NEJM, Apr 2019. URL `https://www.nejm.org/doi/full/10.1056/NEJMra1814259?query=featured_home`.

Thomas Robey. The art of writing patient record notes. AMA Journal of Ethics, 13(7):482–484, 2011.

Devendra Singh Sachan, Pengtao Xie, Mrinmaya Sachan, and Eric P Xing. Effective use of bidirectional language modeling for transfer learning in biomedical named entity recognition. arXiv preprint arXiv:1711.07908, 2017.

Benjamin Shickel, Patrick James Tighe, Azra Bihorac, and Parisa Rashidi. Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. IEEE journal of biomedical and health informatics, 22(5):1589–1604, 2018.

Christine Sinsky, Lacey Colligan, Ling Li, Mirela Prgomet, Sam Reynolds, Lindsey Goeders, Johanna Westbrook, Michael Tutty, and George Blike. Allocation of physician time in ambulatory practice: a time and motion study in 4 specialties. Annals of internal medicine, 165(11):753–760, 2016.

Aharon E tenBroek, Grant S Fletcher, Mardi C Labuguen, and Thomas H Payne. Transition from paper to electronic inpatient physician notes. Journal of the American Medical Informatics Association, 17(1):108–111, 01 2010. ISSN 1067-5027. doi: 10.1197/jamia.M3173. URL `https://doi.org/10.1197/jamia.M3173`.

Tielman T Van Vleck, Daniel M Stein, Peter D Stetson, and Stephen B Johnson. Assessing data relevance for automated generation of a clinical summary. In AMIA annual symposium proceedings, volume 2007, page 761. American Medical Informatics Association, 2007.

Pengtao Xie and Eric Xing. A neural architecture for automated icd coding. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1066–1076, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P18-1098.

Yuhao Zhang, Daisy Yi Ding, Tianpei Qian, Christopher D Manning, and Curtis P Langlotz. Learning to summarize radiology findings. arXiv preprint arXiv:1809.04698, 2018.

Kaiping Zheng, Jinyang Gao, Kee Yuan Ngiam, Beng Chin Ooi, and Wei Luen James Yip. Resolving the bias in electronic medical records. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, pages 2171–2180, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4887-4. doi: 10.1145/3097983.3098149. URL http://doi.acm.org/10.1145/3097983.3098149.

Henghui Zhu, Ioannis Ch Paschalidis, and Amir Tahmasebi. Clinical concept extraction with contextual word embedding. arXiv preprint arXiv:1810.10566, 2018.