PRIVACY-PRESERVING GENERATION OF TEXTUAL HEALTHCARE DATA

A Thesis

Submitted to the Faculty of Graduate Studies and Research

In Partial Fulfilment of the Requirements

For the Degree of

Masters of Science

In

Computer Science

University of Regina

By

Tasnia Faequa

Regina, Saskatchewan

May 2021

UNIVERSITY OF REGINA

FACULTY OF GRADUATE STUDIES AND RESEARCH

SUPERVISORY AND EXAMINING COMMITTEE

Tasnia Faequa, candidate for the degree of Master of Science in Computer Science**,** has presented a thesis titled, ***Privacy-Preserving Generation of Textual Healthcare Data,*** in an oral examination held on May 3, 2021. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner:	*Dr. Xiaoqian Jiang, University of Texas Health
Co-Supervisor:	*Dr. Yiyu Yao, Department of Computer Science
Co-Supervisor:	*Dr. Noman Mohammed, Adjunct
Committee Member:	*Dr. Daryl Hepting, Department of Computer Science
Chair of Defense:	*Dr. Na Jia, Faculty of Engineering & Applied Science

*via ZOOM Conferencing

Abstract

Technological advancements in data science have offered us affordable storage and efficient algorithms to query a large volume of data. Our health records are a significant part of this data, which is pivotal for healthcare providers and can be utilized in our well-being. The clinical note in Electronic Health Records (EHRs) is one such category that collects a patient's complete medical information during different timesteps of patient-care available in the form of free-texts. Thus, these unstructured textual notes contain events from a patient's admission to discharge, which can prove to be significant for future medical decisions. However, since these texts also contain sensitive information about the patient and the attending medical professionals, such notes cannot be shared publicly. This privacy issue has thwarted timely discoveries on this plethora of untapped information. Therefore, in this work, we intend to generate synthetic medical texts from a private or sanitized (de-identified) clinical text corpus and analyze their utility rigorously in different metrics and levels. Experimental results confirm the applicability of our generated data as it achieves more than 80% accuracy in various practical classification problems and matches (or outperforms) the original text data.

Acknowledgement

I would like to address my sincere gratitude to my supervisor Dr. Yiyu Yao and my co-supervisor Dr. Noman Mohammad who have encouraged me to follow this path and provided me with the needed means and guided throughout the graduate studies. I am also grateful towards Dr. Daryl Hepting and Dr. Xiaoqian Jiang for being a part of my thesis committee and provide valuable feedback.

Throughout my Masters, I was guided by the more experienced colleagues, Md Momin Al Aziz, Tanbir Sagar and Toufique Morshed in different aspects of the work and research in general.

My parents, who supported me with love and understanding. Without you and your sacrifices, I could never have received the success that I achieved to this date.

And finally, I would like to thank my husband. You have always encouraged and believed in me. You have helped me to focus on what has been highly rewarding and kept me on track during my Masters degree.

TABLE OF CONTENTS

LIST OF TABLES

List of Figures

LIST OF ABBREVIATIONS

BLEU Bilingual Evaluation Understudy Score

CNN Convolutional Neural Network

DP Differential Privacy

EHR Electronic Healthcare Record

GAN Generative Adversarial Network

GPT Generative Pre-trained Transformer

HIPAA Health Insurance Portability and Accountability Act

LM Language Modelling

LSTM Long Short-Term Memory

PHI Protected Health Information

PII Personally Identifiable Information

PIPEDA Personal Information Protection and Electronic Documents Act

RNN Recurrent Neural Network

SGD Stochastic Gradient Descent

VAE Variational AutoEncoder

Chapter 1

Introduction

1.1 Motivation

Health-care data are digitized during the last decade in hope that digital systems will be beneficial for understanding and improving the care system. Due to the universal consensus about the benefit of such data [ZW11], patient information and clinical procedures are routinely collected in most hospitals. It has created a new area of research, namely Health Information Technology.

A major component of health information technology is made up of Electronic Healthcare Records (EHRs) which are collected at different phases of patient care. For example, the medications and diagnosis of symptoms and diseases are a quintessential part of any EHR. However, due to the nature of the data, they tend to be large and unstructured in most cases. Most importantly, regardless of the applicability, it is not widely used in current health-care solutions.

Technological advancements in data science have also played their part by offering affordable storage and efficient algorithms to query these large volumes of data. The accumulated health data can utilize such methods which might be instrumental for healthcare providers and improve patient care. For example, the physicians can search

for similar patients or share medical history and decide the next course of action using EHRs. However, due to several issues, digital EHRs have not been applied in the contemporary healthcare system.

One of the fundamental reasons is the private nature of the data. These EHRs contain our medical history, demographics, medical conditions and other information which are sensitive and cannot be published or shared without consent. There have been several laws (i.e., HIPAA [Ann03] in the US, PIPEDA in Canada [Nis07]) in action that prohibits the healthcare organizations from sharing the data even among themselves in plaintext. To access such databases oftentimes demands consent forms, certifications, and lengthy administrative processes impeding timely scientific achievements [ASA+19]. Therefore, to tap the resources from such rich medical data, we need privacy-preserving mechanisms that protect the sensitivity of the data and allow the research community to access it alongside.

1.2 Problem Description

De-identification of EHRs: One common approach for privacy-preserving EHR publishing is *de-identification* , which detects and removes the personally Identifiable Information (PII) from EHRs before the public dissemination [AAM20]. Throughout the last two decades, a considerable amount of effort has been devoted to the de-identification problem. At first, expert human annotators tried to find sensitive information manually in the EHRs and remove them, which unfortunately turned out to be both costly and error-prone [DCR+04, DCR+05, NDL+08]. Although the subsequent automated systems involving regular expressions and machine learning algorithms were somewhat an improvement over the manual approaches, these still were very much dataset-specific. In 2016, the first neural network-based de-identification system was proposed by Dernoncourt *et al.* [DLUS16] while recently Ahmed *et al.*

[AAM20] achieved the state-of-the-art model for de-identification. The neural network-based approach is around 97% accurate and significantly better than the previous automated or semi-automated methods without any dataset dependency or manual feature extraction.

On paper, a state-of-the-art model is a reasonable approach to de-identify sensitive tokens as the reported *recall* (accuracy metric) values are over 97%. In other words, 3% PII tokens will be released erroneously, which admittedly is a high recall value for any neural network-based model. Unfortunately, the 3% inaccuracy reported bears more significance than the 97% accuracy. A 97% accuracy signifies that it cannot detect three PII instances for every hundred PII instances. These three PII instances in the hands of a worthy adversary can lead to the re-identification of the patients, consequently rendering the 97% accuracy ineffective. Therefore, a permissible error from a de-identification model should be lower than 1% [YPM20].

Ideally, we need a neural network model with 99% recall value (or more) which is seemingly difficult given the arbitrary structure of EHRs and the amount of information present in them. Though the de-identification accuracies have improved over the years, unfortunately, until now, none managed to achieve this feat. Therefore, the scientific community needs to investigate a parallel approach to EHR de-identification.

1.3 Generation of Synthetic EHRs

The inadequacy of the current approaches curbed EHRs adoption in real-world healthcare data publishing and limited the number of publicly available datasets. Rather than being utilized in medical research, most of the electronic clinical notes collected since 2009 are only stored and left unexplored; hence, approaching the problem from a new standpoint has become imperative. Irrespective of the computational de-identification techniques, there is another way to attain both privacy and pub-

lish clinical notes for potential medical research. We can generate synthetic clinical notes from a subset of de-identified data collected from the whole corpus. Notably, such de-identification can be done using human expertise or computational methods, depending on the required privacy or expenditures.

Recent advancements in machine learning allow us to generate synthetic data from real datasets, representing the original records' statistical properties. Here, the synthetic text data generated using the deep learning techniques is probabilistic and exhibits similar patterns (e.g., medications, disease association) as the input textual dataset but withholds sensitive information. Furthermore, with a privacy-preserving technique such as differential privacy, we can generate an arbitrary number of private records that can be disseminated for data analysis. The privacy risks in publishing these newly generated texts will be lower since they are generated from a mechanism with a standard privacy guarantee and do not include any PII elements. Nevertheless, the utility of the private and machine-generated synthetic data needs to be comparable to the original dataset. For example, the physical symptoms or corresponding medications in this dataset need to be similar to show similar trends as the original data.

Specifically, we will investigate Transformer-based [VSP+17] machine learning techniques to train generation and classification models. The classifiers will be used to determine the utility of the generated data along with several other deterministic utility metrics. The generative models will use differentially private techniques to guarantee the privacy of the input EHR dataset as we compare their usability alongside. In summary, we propose methods to generate EHRs, privately and use different metrics to measure and compare their utility.

1.4 Contributions of This Research

In this thesis, we are considering the clinical notes from the myriad of healthcare data. These clinical notes in Electronic Health Records (EHRs) collect a patient's complete medical information during different timesteps of patient-care available in the form of free-texts. Therefore, these unstructured textual notes contain events from a patient's admission to discharge, which can prove to be significant for future medical decisions.

However, since these texts also contain sensitive information about the patient and the attending medical professionals, such notes cannot be shared publicly. This privacy issue has thwarted timely discoveries on this plethora of untapped information. Therefore, in this work, we intend to generate synthetic medical texts using the state-of-the-art machine learning techniques. Due to the private nature of the data, we also consider a differentially private generation scheme that ultimately guarantees the generated EHRs' privacy as well.

Importantly, we are curious about the utility of such machine generated EHR corpus. Different utility metrics are selected to capture the usability of the generated data under different privacy settings. Our experimental results promote the applicability of our generated data as it achieves more than 80% accuracy in different pragmatic classification problems and matches (or outperforms) the original text data.

1.5 Organization

The rest of the thesis is organized as follows.

- Chapter 2 discusses necessary background materials utilized in different methods proposed in this thesis.

- Chapter 3 describes our proposed method to generate private synthetic clinical notes.
- Chapter 4 presents the detailed experimental results along with the discussions.
- Finally, Chapter 5 concludes the thesis discussing the potential future works.

Chapter 2

Background and Related Works

In this chapter, we discuss some of the necessary background to understand the Thesis. We discuss the structure of the medical data targeted here in Section 2.1. In Section 2.2, we discuss the relevant machine learning techniques utilized for the proposed methods. Section 2.3 outlines the background on the differential privacy as the Related works are finally discussed in Section 2.4.

2.1 Medical Text Processing

As we deal with unstructured textual medical data, we discuss the details about them in the upcoming sections:

2.1.1 Electronic Health Records (EHRs)

There has been a surge in collecting and digitizing our health records from different healthcare organizations such as hospitals, clinics, or next-door physicians. These electronic versions of healthcare records are termed Electronic Health Records (EHRs) where the attending doctors and nurses of any patient will participate in such data collection. Hence, details of every visit from any individual to these institutions are

Table 2.1: HIPAA Safe Harbor Method defined 18 PII or PHI types [Gar15]

No.	PII Type	Description
1	Names	First, Last, Hospital Names
2	Location	Any geographic divisions smaller than a state
3	Dates	Birth date, admission or discharge date
4	Contact	Home, office or cell phone numbers
5	Vehicle	Vehicle serial or license plate numbers
6	Fax	Fax information
7	Device	Device Identifiers and Serial Numbers
8	Email	Any electronic mail addresses
9	URLs	Web Universal Resource Locators
10	SSNs/SINs	Social Security or Insurance Number
11	MRNs	Medical Record Numbers
12	IP	Internet Protocol address
13	Biometric	All finger or voice-prints
14	Insurance	Health Plan Beneficiary Numbers
15	Photo	Full-face (or similar) photographic images
16	Accounts	Bank accounts, social media profile
17	Certificate	a License or certificate number
18	ID	Any unique identifying numbers

saved and maintained digitally by approved providers. Notably, 96% hospitals in the United States now collect EHRs utilizing software-based systems [TOU+18].

The primary motivation behind storing EHRs is to save the history of the patients in a digital version. This information is considered valuable for analyzing the well-being of the patient and others with similar symptoms [RA15]. Another main motivation of these digital data is to share EHRs with other providers over more than one medical care organization. EHRs are imported to several other locations such as research centers, professionals, medical imaging departments, drug stores. Therefore, all of these locations will contain the EHR data from every clinician engaged for a particular patient.

One of the indispensable components of EHRs is the clinical or patient notes. On these textual notes, the detailed information of the patient and the underlying procedures followed are found. These records also contain different lab reports, medications, and attending doctors' names. We can see an example of such a document in Figure 2.1; it includes the patient's history of illness, medication, follow-up instruc-

tions, and other information.

DISCHARGE SUMMARY
Patient Name: Mr. Noah Russell
Admission: 2012-3-1 **Discharge:** 2012-3-7
Date of Birth: 1921-03-17 **Sex:** M
History of Present Illness: Patient is a 91 yo man from Missoula, Montana with multiple medical problems including insulin dependent diabetes secondary to severe pancreatitis in 2006, remote history of Hodgkin's disease in 1996 treated with among other things, radiation therapy which has left the patient with severe osteoporosis and resulting compression fractures, history of alcohol abuse, ...
Discharge Medications:

- Lasix 20 mg p.o. q.d.
- Pantoprazole 40 mg p.o. q.d.

Discharge Diagnoses:

- Chronic obstructive pulmonary disease
- Congestive heart failure
- Insulin dependent diabetes

Follow-Up: He will follow-up with Dr. John in two weeks.

DISCHARGE SUMMARY
Patient Name: [*Name*]
Admission: [*Date*] **Discharge:** [*Date*]
Date of Birth: [*Date*] **Sex:** M
History of Present Illness: Patient is a [*90+*] yo man from [*Location*] with multiple medical problems including insulin dependent diabetes secondary to severe pancreatitis in [*Date*], remote history of Hodgkin's disease in [*Date*] treated with among other things, radiation therapy which has left the patient with severe osteoporosis and resulting compression fractures, history of alcohol abuse, ...
Discharge Medications:

- Lasix 20 mg p.o. q.d.
- Pantoprazole 40 mg p.o. q.d.

Discharge Diagnoses:

- Chronic obstructive pulmonary disease
- Congestive heart failure
- Insulin dependent diabetes

Follow-Up: He will follow-up with [*Name*] in two weeks.

Figure 2.1: Sample of original (replaced with arbitrary PII tokens) (left) and de-identified Electronic Health Records (right)

2.1.2 Personally Identifiable Information (PII)

While collecting the health information from the patients, the EHRs usually contain sensitive information about the patients, such as name, date of birth, address, date of admission, diagnosed diseases, name of caregivers, etc. These are private information

and termed as named Personally Identifiable Information (PII).

As mentioned before, it is unlawful to share these notes publicly as they are confidential to the patient and the healthcare institution. To share these EHRs, we need to remove (sanitize) these identifying information. This process is called de-identification , where we employ the HIPAA safe harbour method [Gar15], which specified 18 categories of information, as mentioned in Table 2.1. A de-identification mechanism will annotate the EHRs with these categories and remove them before dissemination, as shown in Figure 2.1 (right). These sensitive tokens are also named Protected Health Information (PHI) according to the HIPAA criteria.

2.1.3 Word Embedding

To understand the words and their context in a document, algorithms will need a numeric representation which is often handled by a word embedding. Most machine learning tasks representing natural language will translate the words or tokens with a numeric vector embedding. For example, a unique word can be represented with a fixed-size vector, which is handled prior to the machine learning task, in the *Embedding layer*. This layer receives a sequence of tokens (sentences) as inputs from the data layer (Section 3.3.1). Then, it generates random unique numeric encodings (or vectors) for each token. Previously, NLP methods encoded each token as a single discrete number, which perceived no beneficial information about the dis/similarity among the tokens.

The weaknesses of these encoding are mostly addressed in the Vector Space Models (VSMs), where each token is represented in an endless vector space. In VSMs, semantically similar tokens are mapped to nearby locations in a fixed dimensional geometric space. In this mapping, a fixed-sized vector represents one word, as the method is called a word embedding. For example, if you subtract 'Man' from 'Brother' and add 'Woman,' it should give the embedding vector closer to the word 'Sister' (Figure 2.2).

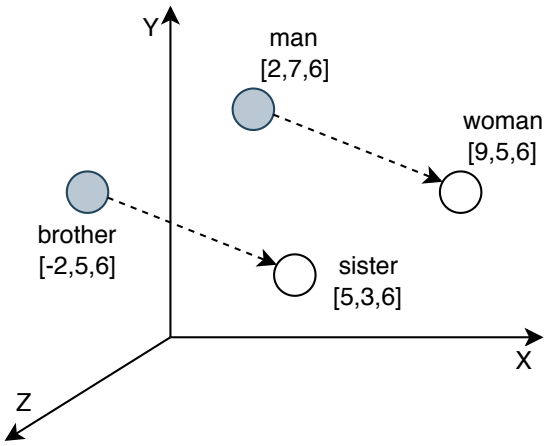

Figure 2.2: Embedding space for four words where similar words have similar distances

Here, these numeric word embeddings help answer questions like "How similar the tokens Brother and Sister is compared to Man and Women?". Nevertheless, there are two different embedding scheme used previously:

Fixed Embedding: In this embedding technique, each token will be represented by a unique numeric vector that is fixed or unchanged regardless of the usage or context. This style of embedding was useful for small corpus with a simpler computational task but did not capture the differences of the words or context in general. For example, in 'The doctor was right about ...' and 'the patient's right atrium ...', the word 'right' has different meanings. However, in the fixed embedding space, both words will have the same vector.

Dynamic Embedding: The dynamic embeddings take the context of the tokens into account and can have different vectors based on their context. For example, the word 'right' will have a different embedding vector in the dynamic embedding scheme as it can mean correctness and direction, based on its usage. We can use a machine learning technique— self-attention [VSP+17] to create such dynamic embeddings which are described next.

2.2 Machine Learning Techniques

In this section, we describe some of the machine learning techniques necessary to understand the methods proposed in this thesis:

2.2.1 Recurrent Neural Network

The Recurrent Neural Network (RNN) is a class of artificial neural networks that is applicable for processing sequential data like tokens in sentences or documents. In RNN, the network acknowledges prior outputs to be used as inputs while having hidden states. We represent the hidden states as h_t for a specific time step, t.

A RNN cell takes input token a_t (i.e., from embedding layer) and prior time step value h_{t-1} as input. It stores the information learned in time step t in hidden state h_t and goes forward with this information to the next time step. At time step t, output y_t is as follows:

$$h_t = \tanh\left(W_h \begin{pmatrix} h_{t-1} \\ a_t \end{pmatrix}\right)$$

$$y_t = W_y h_t$$

Here, W_h and W_y are the weight matrices for the RNN unit which are trained by a learning rule. The updates in the values represent minimizing the errors of the network while training. Furthermore, the weights are updated during backpropagation as we calculate the partial derivative of the error concerning the weight matrices. This partial derivative is termed as *gradient*.

However, one of the shortcomings of the aforementioned vanilla RNN is the long-term dependencies. Traditional RNN only takes information from the recent unit to perform the immediate one. As an example, 'The ducks are swimming in the ___,' it is apparent that the next word is going to be the 'water'. RNN can easily get

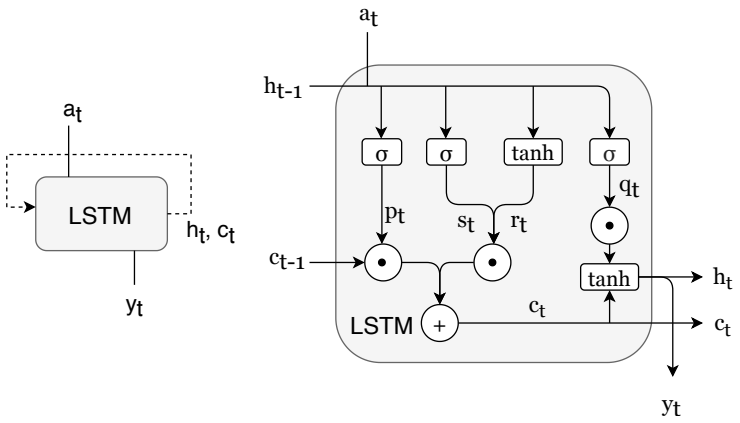

Figure 2.3: Recurrent Structure and Long Short Term Memory

this information from the recent task in these cases as the distance with swimming is pretty small. However, there are some other cases where this gap between the two tasks is greater.

Let us consider a sentence, 'I was born in Canada, and can speak both English and French.' If we try to guess the last three words, we have to need the information from 'Canada' from the previous tokens. Here the distance between the relevant information and the present task is very large and significant. The traditional RNN cannot perform with this kind of large distance, which is called 'long-term dependencies.' Long Short Term Memory (LSTM) is one type of RNN which is capable of learning these long-term dependencies.

Long Short Term Memory: Long Short Term Memory (LSTM) is a special modification of RNN to solve this dependency problem. All RNNs have a series of recurring modules of neural networks. In the traditional RNN, there is only one single NN layer in that recurrent module. In contrast, LSTM has a similar module with four NN layers. These four layers or gates perform element-wise multiplication with the sigmoid function. Every gate detects the information from the memory unit and delivers it to the next one.

Suppose a single LSTM cell that takes a_t, h_{t-1}, c_{t-1} as inputs, and generates hidden

state h_t, memory unit c_t at t step as follows:

$$
\begin{aligned}
p_t &= \sigma(W_p[h_{t-1}, x_t]) \\
s_t &= \sigma(W_s[h_{t-1}, x_t]) \\
r_t &= \tanh(W_r[h_{t-1}, x_t]) \\
q_t &= \sigma(W_q[h_{t-1}, x_t]) \\
c_t &= (p_t \odot c_{t-1}) + (s_t \odot r_t) \\
h_t &= q_t \odot \tanh(c_t)
\end{aligned}
$$

Here, p_t, s_t, r_t, q_t represent the gates used in a LSTM cell. p_1 and s_1 determine whether to write or erase information from the memory cell, respectively. On the other hand, the q_1 gate determines how much information should be written in the memory cell, and r_1 decides the amount of saved information sent as output. These decisions are regulated by the weight matrices W and learned during training.

Symbols σ and tanh refer to element-wise sigmoid and hyperbolic tangent functions, and $\odot$ is an element-wise multiplication. The sigmoid function keeps the values within a certain range forcing the gates to behave like a logic gate. This gating mechanism eliminates the matrix multiplication during backpropagation. As evident from the equation, backpropagation from c_t to c_{t-1} only requires element-wise multiplication.

2.2.2 Transformer Model

RNNs performed poorly for longer documents, which is common in EHR texts. As RNN units utilize the context from their preceding units, which are nearby, they often cannot perceive information from the far back of the document. For example, discharge medications depend on the diagnosis or the 'history of the illness' (Figure 2.1). However, RNN models might forget the context from the 'history of illness' and

utilize only the 'discharge diagnosis' for its predictions. Furthermore, RNNs handle the words in a document *sequentially*, which is comparatively slower.

In recent years, there have been several attempts on paying more attention to specific words to solve some of these problems. Neural networks can focus on the part of a subset of the information given to them. For example, the output of another RNN can be attended by a new RNN. This concentrates on different positions in all other RNNs at every phase of the way.

Attention is a method used in a neural network to solve these problems. In this technique, each word has a corresponding hidden state which is passed through the decoding process instead of just encoding the entire sentence in a hidden state for RNNs. The hidden states are then utilized to decode at each stage of the RNN. The concept behind it is that every word in a sentence might carry relevant information. Therefore, every word of the input must be taken into account, using attention for the decoding to be successful though some problems are not solved with attention-based RNN as it is not feasible to process inputs (words) in parallel. It increases the time allocated for translating the text for a large corpus of the text.

Convolutional Neural Networks (CNNs) help to solve the parallelization problem of attention-based RNN. In that technique, each input language can be processed simultaneously and does not necessarily depend on the preceding words to be translated. But CNN also has some problems that are the method have difficulties with the dependencies when translating sentences. To figure out this problem, the new model proposed with the combination of CNNs with attention named *Transformer model*.

Transformers attempt to solve the parallelization problem. They use Convolutional Neural Networks in combination with attention. The transformer model applies attention to increasing speed, and more specifically, it uses self-attention to translate from sequence to sequence.

Transformers mainly work like only paying attention to each other in one dimension, but the model uses the concept of Multihead attention for the modification of the work. The concept of Multihead attention is whenever you are translating a word, you may give specific attention to every word based on the type of question that you are asking. Mainly we use a Multi-headed Self-attention Model in our model.

2.2.3 Multi-headed Self-attention Model

The self-attention mechanism for language modelling and dynamic embedding calculation was first introduced by Vaswani *et al.* [VSP+17] as a replacement of the firmly established Recurrent Neural Network (RNN) [GK96] based models. The model introduced in [VSP+17] is named the Transformer model, which consists of an encoding component, a decoding component, and connections between them. Both the encoding and decoding component is a stack of encoders and decoders blocks, respectively. Each block in their respective stacks is identical to each other. However, the structure and functionality of an encoder block and decoder block are different from each other.

An encoder block has two sub-layers–(i) a Multi-headed Self-Attention sub-layer and (ii) a Feed-Forward Neural Network sub-layer. The inputs to the encoder flow through the self-attention layer first. This layer helps to look at every other word in a sentence when it encodes a specific word. Hence, the name self-attention. The output of the self-attention layer is fed to the feed-forward neural network. Each sub-layer has a residual connection around it and is always followed by a layer-normalization step.

The output of the final encoder bock this forwarded to the decoder component. As mentioned earlier, the decoder component is constructed with a decoder block. These decoder blocks also have two sub-layers, similar to the encoder block with one major distinction in the self-attention sub-layer. Whereas in the encoder blocks, the attention score of each input word is calculated with respect to every other input

word, the decoder blocks only look at only to the words that came before the current word for which the attention score is calculated. This slightly modified self-attention is called masked self-attention. As we have seen we used only the encoder component in the classifier models and the decoder component in the generation model.

2.3 Differential Privacy

In this thesis, we utilized a privacy technique to protect sensitive information from the patients. Differential privacy (DP) [Dwo06] is a cryptographic method that guarantees privacy over the data or any outputs generated from the data. It is a probabilistic mechanism that provides a theoretical guarantee and does not depend on the data points. Hence, it is a generalized mechanism that works for arbitrary datasets. In other words, any mechanism proved to be differentially private will hold its privacy guarantee under any query or underlying data. Formally,

Definition 1. A randomized mechanism $\mathcal{M}$ is (ϵ, δ) differentially private if for a set of neighbouring datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ where $\mathcal{D}_1$ is different from $\mathcal{D}_2$ in at most in one record ($||\mathcal{D}_1 - \mathcal{D}_2|| \leq 1$) and for all possible dataset created by $\hat{\mathcal{D}} \subseteq Range(\mathcal{M})$,

$$\mathcal{P}[\mathcal{M}(\mathcal{D}_1) \in \hat{\mathcal{D}}] \leq \exp(\epsilon)\mathcal{P}[\mathcal{M}(\mathcal{D}_2) \in \hat{\mathcal{D}}] + \delta.$$

2.3.1 Parameters of Differential Privacy, ϵ, δ

The core component in a DP mechanism is the privacy budget or loss ϵ. On a high level, from Definition 3, ϵ regulates the amount of noise allowed in the output. For example, if this budget $\epsilon = 0$ and $\delta = 0$, it will results in $\mathcal{P}[\mathcal{M}(\mathcal{D}_1)] \leq 1 * \mathcal{P}[\mathcal{M}(\mathcal{D}_2)]$. This denotes that our proposed mechanism offers the same probability of answering any query is the same for $\mathcal{D}_1$ and $\mathcal{D}_2$. It also denotes that the output or analysis using $\mathcal{M}$ does not depend on any single individual.

As a result, lower ϵ's offer better privacy while larger ϵ values do the opposite. However, this higher privacy from lower ϵ's does come at a cost in terms of accuracy or utility as it will add larger noise values to the results (details in Gaussian Mechanism 2.3.3). These noisy results inadvertently affect the utility necessitating an optimization between the desired privacy and accuracy.

The value of δ is another discussion point for any DP mechanism. If any algorithm supports $\epsilon > 0, \delta = 0$, it is called a pure differential privacy whereas $\epsilon > 0, \delta > 0$ is known as *Approximate differential privacy.* Any $\delta > 0$ represents that with probability $1-\delta$, the mechanism $\mathcal{M}$ will hold ϵ privacy guarantee and fail for δ times. Therefore, the value of δ is kept quite low as even $\delta \in \mathcal{O}(1/|\mathcal{DB}|)$ will be critical as it will reveal the records coming from a small number of participants. Informally, δ offers a bound when the randomized mechanism will fall short in terms of privacy whereas ϵ provides the upper bound on the impact of an individual's record on the final result.

2.3.2 Function f and Sensitivity Δf

In differential privacy, the targeted function to execute on the dataset $\mathcal{D}$ has immense importance. For example, real and linear functions $f : \mathcal{D} \rightarrow \mathbb{R}^d$ denotes a function f being executed on $\mathcal{D}$ which results in a set of d real numbers $\mathbb{R}$. Such linear functions, f represent the queries or operations performed on the dataset $\mathcal{D}$.

Sensitivity of the targeted function f effects the privacy in any DP mechanism. It tells us about how many rows are effected for a particular f. For a count query, every record $r_i \in \mathcal{D}$ is required and checked whether it satisfies the preset conditions, $f_{count}(\mathcal{D}) = \sum_1^n cond(r_i)$ where $cond(r_i) \in \{0, 1\}$. Hence, l_1 sensitivity can be defined as,

Definition 2. Sensitivity for any real valued function $f : \mathcal{D} \rightarrow \mathbb{R}^d$ is defined as,

$$\Delta = \Delta f = \max_{\mathcal{D}_1, \mathcal{D}_2 \in \mathcal{D}} ||f(\mathcal{D}_1) - f(\mathcal{D}_2)||_1$$

The aforementioned definition shows the l_1- sensitivity of a single record in $\mathcal{D}_1$ as $||\mathcal{D}_1 - \mathcal{D}_2|| \leq 1$. In other words, a single participant can only effect f_{count} by maximum 1 (absent/present) on a count queries; hence $\Delta f_{count} = 1$. However, for maximum (/minimum) queries, the sensitivity will not be 1, since one record's presence can alter the whole result.

2.3.3 Laplace and Gaussian Mechanism

Laplace Mechanism ($\mathcal{M}_{\mathcal{L}}$) adds a random noise from the independent and identically distributed (i.i.d.) Laplace distributions according to:

$$f(\mathcal{D})' = f(\mathcal{D}) + Lap(\frac{\Delta}{\epsilon}). \tag{2.1}$$

Here, $Lap(x|\lambda)$ has a probability density $\frac{1}{2\lambda} \exp\left(\frac{-|x|}{\lambda}\right)$ where $\lambda = \Delta/\epsilon$ and centered around 0.

In Gaussian Mechanism, the laplacian noise is replaced with a Gaussian one as it attains (ϵ, δ) differential privacy. Similar to equation 2.2, for Gaussian Mechanism:

$$f(\mathcal{D})' = f(\mathcal{D}) + \mathcal{N}(\sigma^2), \tag{2.2}$$

where, $\sigma = \frac{2\Delta \log 1.25/\delta}{\epsilon^2}$ and centered around 0.

2.4 Related Works

Broadly, this work concerns two different active fields of research: 1) Medical text generation with Deep Learning algorithms and b) Privacy-Preserving techniques over text data. Therefore, in this section, we separate the different works in these individual areas and highlight the advancements made over the last few years. We encourage the readers also to check these surveys as they connect NLP problems in the medical domain, including different machine learning solutions [WRD+19, DKBB19, XCS18, STBR17].

2.4.1 Generation Techniques

Generating unstructured texts is an active area of research in deep learning and natural language processing. Specially since 2015 multiple methods are proposed for such generation, we categorize our related works according to the corresponding machine learning technique:

Variational Autoencoder

The Variational Autoencoder (VAE) [KW14, RM15] is a generative model that is based on a regularized version of the standard autoencoder combined with variational inference. VAE has developed as one of the well-known methods for unsupervised learning of complicated distributions [Doe16]. Therefore, VAE was applied for several text generation problems. Usually, a recurrent neural network-based language model generates sentences one word at a time step, instead of generating a complete sentence at a time. To solve this problem, Bowman *et al.* [BVV+16] introduced an RNN-based VAE generative model that combines distributed latent representations of the complete sentences. This method used simple deterministic decoding, especially to generate distinct and well-formed sentences.

For medical texts, Dr. Lee [Lee18] has shown promising results with the encoder-decoder model relying on LSTM cells. It utilized around 5 million de-identified emergency department records from the New York City Department of Health and Mental Hygiene. The work also extended to epidemiological validity, different semantic similarity metrics and analyzed the PII tokens' presence on the generated data. Unfortunately, due to the nature of autoencoder models, these generators often picked highly frequent words, which is avoided with Transformer or self-attention models. These autoencoder models are also slower to train, which is why we did not opt for VAE-based solutions in generating EHRs.

Generative Adversarial Network (GAN)

GAN was proposed in 2014 by Goodfellow *et al.* [GPAM$^+$14], which utilizes a generator-discriminator-based model. The generative component of a GAN generates synthetic data, and the discriminator tries to predict whether the data is real or fake. The generator learns to generate better data by trying to fool the discriminator and conceptualize the data inherently. However, regardless of this simplicity while training, GANs often behave unpredictably and prove to be challenging to understand their behaviour.

In 2017, Yahi *et al.* [YVET17] proposed an unsupervised framework to produce consecutive time-series data of EHRs exploiting the GAN technique. The framework predicts the consequence of drug exposure on laboratory test data. Similarly, Esteban *et al.* [EHR17] proposed a Recurrent GAN (RGAN) and Recurrent Conditional GAN (RCGAN) to generate multi-dimensional real-valued time series, which can prove to be useful for medical applications. RGAN uses a recurrent neural network (RNN) in the discriminative and generative model as they are trained on auxiliary information in RCGAN.

In 2016, Guan *et al.* proposed an approach named Medical Text Generative Ad-

versarial Network (mtGAN), which generates synthetic data EHRs [GLYZ18] using GANs. It uses a summarized version of the EHRs to train its `reinforcement` algorithm-based network. In this work, we utilize the full unstructured EHR texts, which are longer and contains more information than a summary.

However, all GAN models generate texts by sampling words sequentially, and sometimes the outcomes of these models are not realistic due to the lower quality of their sampled words. Recently, MaskGan is proposed by Fedus *et al.* [FGD18] in 2018, improving the quality of the sample token which comprises the synthetic data. It proposed an actor-critic conditional GAN framework that teaches the generator to generate high-quality samples and showed success in image generation. Primarily, the model is trained using the Cloze task (similar to fill in the blanks) on a large text corpus. Notably, this is the first unconditional generative model using sequence-to-sequence learning. In 2018, Jordon *et al.* [YJS19] introduced a state-of-the-art GAN framework named PATE-GAN which can be trained to generate `differentially private` [DR13] synthetic data. They revised the Private Aggregation of Teacher Ensembles (PATE) framework [PSM+18] and applied it to GANs.

Reinforcement Learning (RL)

However, GAN techniques have a limitation where the outputs of the model create obstacles for passing the gradient update from the discriminative model to the generative model. To solve this problem, some studies apply reinforcement learning (RL), where RL policy is used as the method of the discriminator guiding generator.

Yu *et al.* introduced a new GAN method named sequence generation framework (SeqGAN) [YZWY17] in 2017. SeqGAN altered the generative model to avoid differentiation and directly go to the policy gradient using the RL technique. The reward-based on RL trains the discriminative model on a complete sequence and goes back to applying a Monte Carlo search. The BLEU and G^2 test results on smaller

sequences were promising as we employed this method to generate textual EHRs, and added other utility metrics (Section 4.1.3 and 4.1.4).

However, there are some shortcomings when the required length of the generated text is arbitrarily long. To address this problem, Guo *et al.* [GLC+18] proposed a new RL framework called LeakGAN. In their studies, they allow leaking its own high-level extracted features from the discriminative model to strongly guide the generative model. The framework has two modules, namely Manager and Worker, where the manager learns the extracted features of the current generated words, and the worker learns to fulfill the next one. However, it was difficult to train on the EHR dataset, and we opted for its predecessor, SeqGAN, for benchmarking.

Recently in 2020, Ive *et al.* [IVK+20] utilized self-attention models [VSP+17] to generate artificial mental health records along with MIMIC-III EHRs. Though, our generation task is parallel to [IVK+20] while differing on the privacy concepts as we added differentially private generation. We also proposed computational utility metrics that provide different perspectives. Interestingly, there was a human evaluation task in [IVK+20] which we consider as an important future work.

2.4.2 Privacy Preserving Techniques

In terms of medical text data privacy, the most popular technique at use is de-identification. Here, we discuss some of the seminal works in de-identifying EHRs and add some differentially private generation techniques as well:

De-identification Techniques

Prior to the proliferation of machine learning techniques, de-identification is mostly done with grammatical or rule-based methods. These systems employed different patterns, regular expressions and relied on dictionary searches to detect PII tokens. Douglass *et al.* [DCR+04] developed one of the earlier schemes in 2005 where

the healthcare professionals highlighted PII tokens on a machine. Later on, Douglass *et al.* [DCR+05] extended that method to incorporate lexical dictionaries, simple heuristics and regular expressions to locate PII instances. In 2008, Nematullah *et al.* [NDL+08] contributed the state-of-the-art rule-based automated de-identification that had a recall value of 94.3% on 130 documents. Notably, these approaches did not scale for large EHR corpora, often faltering for unstructured inputs from different physicians or hospitals.

The machine learning-based automated de-identification gained momentum around 2014 after the i2b2 challenge [SKU15, US15]. It offered a human-annotated real-life EHR dataset for de-identification, containing around 500 documents. Among the ten participating teams with 22 different de-identification systems, the highest recall achieved was around 91% over the i2b2 dataset. The winning solution achieved from Yang *et al.* [YG15] utilized Conditional Random Fields (CRFs), along with regular expressions to capture the PII tokens.

The performance of CRFs [ABY+10] and other Bayesian solutions [CCG15] intrigued the machine learning community to investigate the de-identification problem. Furthermore, the improvement of the neural network techniques, especially the deep learning techniques, was surfacing around that time. In 2017, Dernoncourt *et al.* [DLUS16] utilized LSTM and CRFs in conjunction to propose a deep learning-based model to de-identify both MIMIC-III and i2b2 dataset. It is one of the earliest works that showed the impact of a sizeable dataset when using deep learning techniques such as the LSTM network. It achieved an astounding 97.835% recall on i2b2 and was the state-of-the-art until recently Ahmed *et al.* [AAM20] improved it to 98.41%. This work, published in 2020, employed a self-attention mechanism (same as our classifiers) to detect the PII tokens replacing the LSTM network.

Differential Privacy

Differential Privacy (DP) [DR13] offers standard privacy-preserving techniques when providing a theoretical guarantee over the privacy of the participants. Due to its robustness in data [NHDC20, PG18], DP algorithms have seen much success in deep learning algorithms as it helps to avoid overfitting [PAE+16]. Though differential privacy has been around 2006, we will only discuss its applications in concurrent deep learning techniques.

Around 2014, Song *et al.* [SCS13] and Shorki *et al.* [SS15] proposed differentially private algorithms to perform Stochastic Gradient Descent (SGD) [Rud16] algorithm. These paved the way for larger neural networks to incorporate DP approaches as SGD is commonly used in many different machine learning methods. In 2016, Abadi *et al.* [ACG+16] improved earlier works by proposing another method that uses stricter bounds and lowered the privacy loss. The fundamental contribution from [ACG+16] is the 'Moments Accountant' that calculates the privacy cost at each private data access and accumulates the cost progressively as the training continues. This is an alternate to the stricter sequential composition [DR13] allowing a lower privacy cost of ϵ.

Such a differentially private approach saw its application in different machine learning algorithms. McMahan *et al.* [MRTZ17] showed impressive performance while using the DP-LSTM approach as the private model performed similarly to the original one. In recent years, researchers have found better ways to manage the privacy budget ϵ for ML techniques [PMSW18]. One of them is the 'Private Aggregation of Teacher Ensembles' or PATE [PAE+16, PSM+18] which uses a student-teacher model. Here, an ensemble of 'Teacher' models transfer knowledge to a 'Student' model. However, the student model only gets differentially private access to the teachers' models as it results in stronger privacy bounds accumulated by these models.

For medical texts, Melamud and Shivade [MS19] proposed a differentially private generation technique using Language Models (LM). The major motivation here was to replace the de-identification models as they might reveal sensitive information under adversarial attacks [SSSS17]. Realistically, these attacks are not closely investigated for the recent attention-based de-identification models as research works are just surfacing [LMG$^+$20]. This work [MS19] proposed a workaround with DP methods on LM and analyzed the corresponding utility of the generated texts, particularly the privacy with different metrics. Related to this, Kerrigan *et al.* [KST20] in 2020 proposed a language model where the choice of the tokens was driven by differential privacy. We incorporate DP while training, but this can also be tested in future works.

Chapter 3

Private Synthetic Text Generation

In this chapter, we discuss the proposed techniques to generate privacy-preserving medical text data. Initially, we start by summarizing the contributions in Section 3.1. Section 3.2 describes the targeted problem in details. Finally, The methods to generate the data are described in Section 3.3.

3.1 Contributions

In this thesis, we propose a differentially private neural network-based model for generating synthetic EHRs that satisfy both the privacy and utility requirements (as mentioned above). Importantly, we generate synthetic unstructured medical texts under two different privacy models: 1) from de-identified data where the PII tokens were never present in the input data (sanitized), and 2) generate texts with a differential privacy guarantee. Here, in both cases, the resulting text dataset will not contain any identifying information on any individual. However, we emphasize heavily the usability of such machine-generated texts as it was the foremost reason behind such generated synthetic private data in the first place. The primary contributions of this article can be summarized as follows:

- We employ a self-attention based [RWC+] generative neural network to create synthetic and private unstructured medical texts from multiple original and sanitized datasets (Section 3.3.1).

- Furthermore, a differentially private mechanism is incorporated with the generation technique to guarantee quantifiable privacy over the generated synthetic EHR text data (Section 3.3.1). To the best of our knowledge, this is the first work on generating long medical texts using a differentially private mechanism.

- We rigorously analyzed the utility of the generated text corpora at three different levels (word, document, and corpus) and demonstrated that our generated synthetic EHR corpus could be utilized as a substitute for real-world datasets in applications such as medical document classifiers ((Section 3.3.2).

- The proposed approach is compared with another text generation model, SeqGAN [YZWY17]) and shows the efficacy of our method in generating long unstructured medical documents. Experimental results show that the generated texts from our method outperform the existing method and deliver satisfactory results in all utility metrics as it achieves BLEU-1 scores of a minimum 0.89 and more than 80% accuracy in different disease classifications and adversarial success (Section 4.1).

3.2 Problem Description

The primary objective of our work is to generate synthetic medical (EHR) texts which do not contain the original PII values but represent the other statistical properties from the original EHR texts. For example, our synthetic dataset needs to be privacy-preserving as it cannot have the name, date, or contact information, which can be used to re-identify a patient if published. However, the medically relevant information

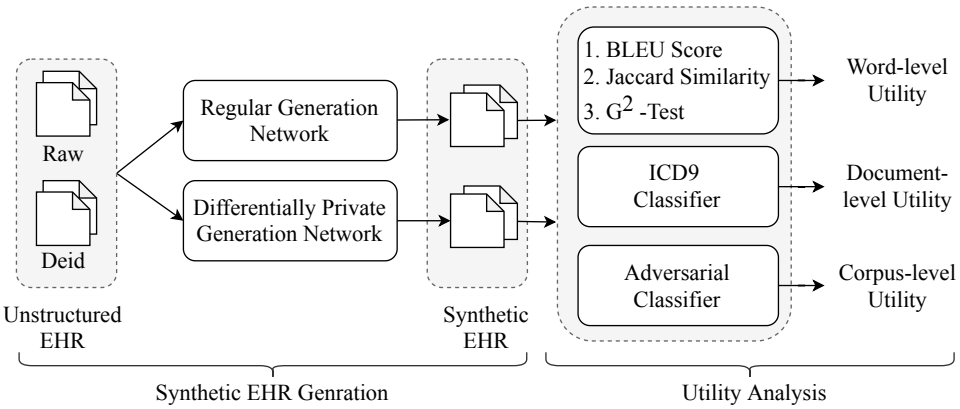

Figure 3.1: Overview of the solution mechanism separating two different tasks: a) generation, and b) utility analysis

such as the patient's underlying symptoms, disease diagnosis, or the medications at different timesteps will be coherent in these generated EHRs.

An overview of our work is depicted in Figure 3.1. We generate private synthetic EHR text data from a small subset of de-identified text corpus using deep learning techniques with a competitive utility. In other words, we have three separate tasks- (i) Generation of clinical texts from de-identified EHRs, (ii) Provide a quantifiable privacy guarantee over the generated data, and (iii) Utility analysis of these private and synthetic texts (Figure 3.1). In the following subsections, we briefly describe how we approach and evaluate these two tasks:

3.2.1 Synthetic EHR Generation

Publishing the vast resources available in the textual EHRs is the primary motivation behind generating textual EHRs. This task will take raw unstructured de-identified EHRs as inputs and build a model that can output similar EHR texts (synthetic data). Here, the newly generated text should follow the original text's statistical properties, such as the correlation of prior medical history (and medications) with the patient's present conditions. For example, in Figure 2.1 the original discharge summary (left), the patient had congestive heart failure and was prescribed

with Lasix (to treat fluid due to heart failure). As these relations are not sensitive and need to be preserved in the synthetic data, we hope to achieve similar trends in the generated data too. We further discuss this topic in Section 3.3.1.

3.2.2 Privacy-Preserving Generation

Since the contemporary state-of-the-art de-identification techniques do not perform convincingly in terms of finding the PII tokens, we incorporate a differentially private mechanism (Definition 3) on the synthetic data generation. With arguable de-identification techniques, resulting in de-identified data as an input to the synthetic model might leak private information on the generated medical texts. Therefore, we adopted an additional mechanism to generate tokens with manageable noise, regulated by a strict differential privacy guarantee. Our primary goal here is to compare both of these generated texts (Section 3.3.1 and 3.3.1) in terms of various utility metrics which we describe next.

3.2.3 Utility Analysis

We use different methods and metrics to measure the utility of the generated EHRs. We can broadly categorize these utility metrics according to the level of granularity:

- The first utility metric operates on the *word-level*, where we compare the word frequency from the original and generated EHRs. This can also be termed micro-level analysis. We employ three standard techniques (Jaccard similarity, BLEU score comparison, and G^2-test) to analyze how the generated texts are different from the original de-identified texts.

- *Document-level* or our macro utility analysis will utilize a classification task to show the utility of the generated EHRs in a real-life application. Specifically, we will use a supervised machine learning task to train two separate ICD9

code [OCP+05] classifiers using the original and generated EHR documents, respectively, and compare their performance.

- Lastly, we perform a *Corpus-level* evaluation where we employ a binary classifier to distinguish or separate the original EHRs apart from the synthetic ones. It is similar to the Turing Test, which will utilize human trials and expertise to classify the text documents (whether synthetic or original). This metric will consider the original and generated documents as two different types and classify a random document. The better the quality of the synthetic document, the closer the classifier's accuracy to 50%.

Notably, our three utility metrics use different techniques, and these techniques are described in Section 3.3.2.

3.2.4 Privacy Model

The privacy of the synthetic texts is important as it should not reveal any of the patient information in the generated data. In this work, we incorporate privacy-preserving techniques in two stages: 1) We first generate de-identified EHR data from the original textual data, and then 2) utilize a differentially private generation technique to produce a synthetic dataset. We define de-identification as an identifying information removal mechanism such that no particular individual can be linked back to the de-identified data. We follow the de-identification mechanism according to the HIPAA Safe Harbour Method [NG06], where there are 18 defined categories of Personally Identifiable Information (PII). All PIIs should be removed to de-identify any EHR. In this work, we adopt the HIPPA specified privacy model and detect, remove the PII elements from any EHR as defined in Table 2.1.

Since the original or raw EHRs contain PII tokens that can identify any individual (e.g., patient, caregivers), we need to utilize a de-identified mechanism that ensures

their privacy. Specifically, we intend to generate de-identified synthetic EHRs, which will not contain any PII tokens as inputs in the first place. Therefore, we consider the original EHRs and de-identified EHRs separately as we generate synthetic medical texts for both corpora and analyze their utility.

However, as we cannot solely rely on the de-identified techniques on such a large corpus, we also explore other methods to ensure the privacy of the generated data so that it cannot be linked back to any individual. Here, we use a differentially private mechanism, DP-SGD [PAE+16] incorporating in our generation technique by introducing a privacy budget ϵ, which operates as a tuning parameter between privacy and the quality of the generated texts (or utility). This parameter, ϵ also provides a quantifiable privacy guarantee over the generated synthetic data.

3.3 Methods

3.3.1 Synthetic Text Generation

We consider two standard dataset: i2b2 2014 de-identification dataset [SKU15] and MIMIC-III [JPS+16] while generating synthetic EHRs. These two datasets were chosen due to the availability of PII annotations which were necessary for our targeted privacy model. The i2b2 2014 dataset also has the raw (unsecured) EHR corpus, which allowed us to generate synthetic EHRs from the raw corpus. However, the MIMIC-III dataset gives access only to the human de-identified corpus. Using the MIMIC-III raw data would require manually adding custom PIIs, which would bear statistical similarity to the original raw data, which is an expensive and time-consuming manual process. Hence, we only used the de-identified MIMIC-III data in our experiments.

Input Data

Before the synthetic EHR generation, we prepare two *different* text corpora from one dataset. The first set contains the private EHR records that include all PII tokens that are sensitive and can be used to identify any individual. This dataset is referred to as original (or raw) data throughout the thesis, which is not safe to publish. Consequently, the synthetic EHRs generated from this set are not publishable as they might contain sensitive PII tokens (i.e., names), On the other set, we create a corpus with only the perfectly de-identified EHRs according to the HIPAA privacy requirements. This dataset is named as *de-identified* (or sanitize) dataset in the rest of the thesis. These PII tokens are annotated employing human expertise and categorized according to Table 2.1. Notably, as the MIMIC-III dataset was only available without PII tokens we could not produce this set of input.

The two datasets (original and de-identified) were fed into the generation model, separately. Specifically, the two different text corpus (with and without PIIs) went through a generation (and differential privacy) model and provided two parallel synthetic corpora. Here, the generation models had the same settings and model parameters were initialized from a publicly available pre-trained model [RWC$^+$]. Notably, the output from the original data may contain PII tokens whereas the other set has not seen any PII tokens during its training. The motivation here with two different sets is to check whether the utility from both sets (original and de-identified) are the same.

The documents from the EHR corpus were first broken into sequences (or sentences) and represented as a group of tokens (or words). For simplicity, we considered the maximum sequence length to be equal to the maximum number of tokens in the largest document in the corpus. These sequences are then forwarded to our generation model.

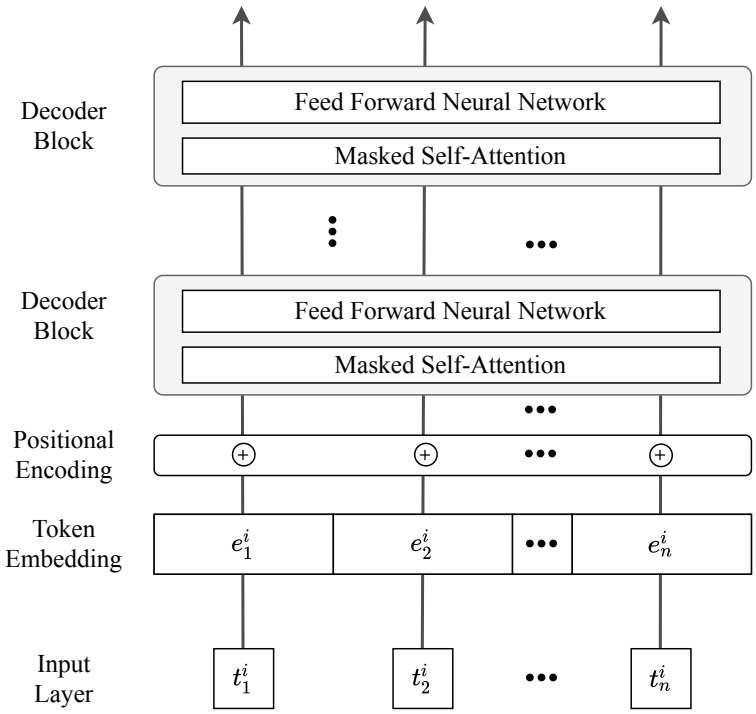

Figure 3.2: Simple Architecture of Generation Model

Generation Model

We ended up with two different EHR sets from the input layer, original and de-identified corpus, albeit the generation technique will remain the same for both corpora as we proceed to the utility analysis. The generation model receives the token sequences from the input layer as shown in Figure 3.2, where, sequence $\{t_1^i, t_2^i, ..., t_n^i\}$ represents the i^{th} document with token length n.

The underlying generation depends on Language Modeling (LM) [SC99] technique, which gained popularity after recent breakthroughs in different NLP tasks [DCLT19, RWC$^+$]. LM is an unsupervised task that is useful for training a large corpus of unstructured text. Since the EHR texts are mostly a sequence of tokens, we can model each document i as conditional probabilities where we get the probability of

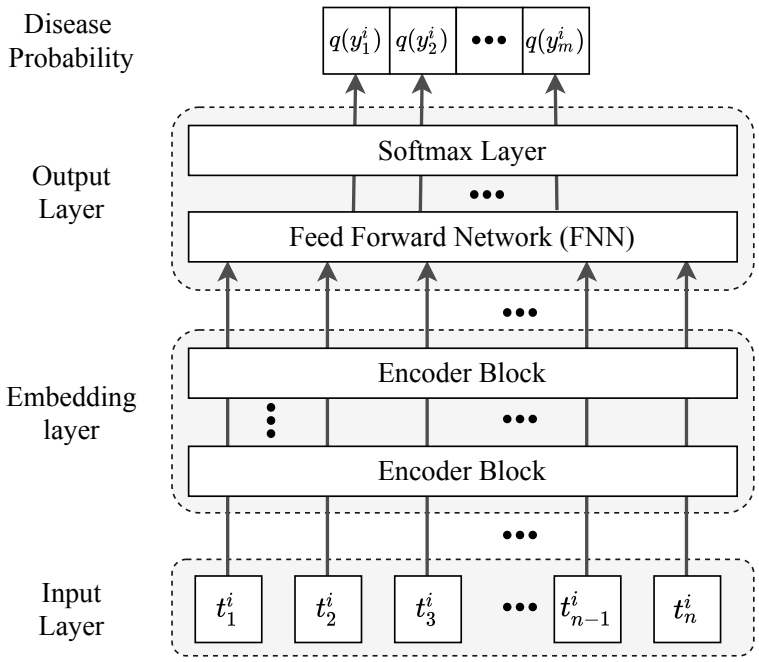

Figure 3.3: Simple Architecture of Classification Model

j^{th} token t_j as [BDVJ03]:

$$p(i) = \prod_{j=1}^{n} (p(t_j | t_1, \ldots, t_{j-1})) \tag{3.1}$$

Therefore, we can sample the next tokens from $p(t_{n-j}, \ldots, t_n | t_1, \ldots, t_{n-j-1})$ probability. Here, these probabilities are utilized in a deep learning model, which efficiently trains the sampling probability of each token in the vocabulary. Notably, for a large vocabulary, calculating the probability of each token from a massive corpus is computationally expensive. Therefore, we opted for a self-attention [VSP+17] based generation model as previous methods cannot process the tokens and their conditional probability in parallel and efficiently. The self-attention mechanism was first introduced in 2017. In their original paper, Vaswani *et al.* [VSP+17] described the Transformer model, which consists of an encoder stack and a decoder stack. Since the inception has been several variations to the Transformer model. In this work, we used an auto-regressive variation of the Transformer model. Similar to the GPT2 [RWC+]

model, this variation consists of only the decoder stack, as shown in Figure 3.2.

As mentioned earlier, the token sequences are passed to the generation model. The token embeddings for each of these tokens are looked up in the pre-trained embedding matrix we used. In the Figure 3.2, $\{e_1^i, e_2^i, \ldots, e_n^i\}$ represents the embedding for the token sequence $\{t_1^i, t_2^i, \ldots, t_n^i\}$, respectively for the i^{th} document. Before moving these embeddings to the first decoder block, a positional encoding is added to these tokens to indicate their order in the document/sequence. Each of the decoder blocks is composed of a multi-headed masked self-attention layer and a simple feed-forward network. In contrast to a self-attention layer, masked self-attention stops the model from seeing tokens that are at the right of the current position. This allows the model to take into account the last token when generating the next token. Hence, auto-regressive nature. For an input, 'Patient is a 91-year man from Missoula', we expect the model to output the next word 'Montana' (from Figure 2.1). Then, we add the current sample ('Montana') to the sentence and proceed to the next one. This auto-regression is the only similarity to the traditional generation model and the major difference with BERT [DCLT19], which uses a bi-directional approach while generating.

Now, the first decoder block passes the token embedding through the masked self-attention process to a feed-forward network. The resulting vector representation for each token is moved up through identical decoder blocks. Each decoder block, however, has its own weights, which are learned during training. At the final decoder block, the output vectors are multiplied by the embedding matrix. Each row in the embedding matrix corresponds to the embedding of a word. Therefore, the result of this multiplication is considered as the score for each word in the model's vocabulary. Even though selecting the token with the highest score would produce a reasonably competitive result, a better strategy is to sample a word from the entire list using the score as the probability of selecting that word. In our experiments, we set top_k to

40 and have the model consider the 40 words with the highest scores. This process describes one iteration, where each iteration generates a single word. The iteration is continued until the required sequence length is generated.

Furthermore, there was another condition in the generation relying on the type of EHR. We utilized the underlying disease or medical conditions as they were annotated according to the International Classification of Diseases (ICD9) specifications [OCP+05]. We shortlisted seven different diseases, and the documents from the original dataset were placed under each condition (more details in Section 3.3.2).

Differentially Private Generation

The private generation technique relies on a Differentially Private (DP) training method, which we incorporated with the GPT-2 [RWC+]. The differentially private training method we used was first introduced by Abadi *et al.* in 2016. Their original paper showed that while updating the wights during backpropagation introducing randomness ensures differential privacy to the training data. Our private generation model was kept almost the same as we described earlier in Section 3.3.1. For example, the input layer, followed by the transformer-based generation model, was the same. However, the optimization algorithm was modified as outlined by Abadi *et al.* . Now, Differential Privacy (DP) [Dwo06] is a privacy mechanism offering theoretical and quantifiable bounds on the disclosure of data which can be formally described as follows:

Definition 3. A randomized algorithm $\mathcal{A}$ is (ϵ, δ) differentially private if for a set of neighbouring datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, where $\mathcal{D}_1$ is different from $\mathcal{D}_2$ in at most in one record ($||\mathcal{D}_1 - \mathcal{D}_2|| \leq 1$) and for all possible dataset created by $\hat{\mathcal{D}} \subseteq Range(\mathcal{A})$,

$$\mathcal{P}[\mathcal{A}(\mathcal{D}_1) \in \hat{\mathcal{D}}] \leq \exp(\epsilon) \times \mathcal{P}[\mathcal{A}(\mathcal{D}_2) \in \hat{\mathcal{D}}] + \delta,$$

where, $\mathcal{P}[\mathcal{A}(\mathcal{D}_1) \in \hat{\mathcal{D}}]$ denotes the probability of computing any function from $\mathcal{D}_1$.

Informally, the definition basically offers that with a predefined (ϵ, δ), the output to any query on the original dataset $\mathcal{D}_1$ may come from $\mathcal{D}_1$ or $\mathcal{D}_2$ with a probability ratio of $exp(\epsilon)$. Here, δ further reduces the probability of producing the output from the original dataset $\mathcal{D}_1$, which was fixed at 10^{-5}.

In our optimizer, for the randomness requirement, we used the differentially private Gaussian mechanism [DR13]. Similar to [KST20, MS19], while updating, the original weights are tempered with noise values calculated using the Gaussian mechanism. In every iteration, while querying for the original data to compute the loss function, a random value is added with the input. Therefore, the difference between the prediction and the input token (or loss) was different. The updates in the network had this effect as they generated differentially private tokens. Finally, the input layer of this network was different from the original network due to the noise, providing a privacy guarantee of (ϵ, δ).

Classification Model

For the utility analysis, we will require multiple classifiers that provide accurate comparisons for our generated and original data. Analogous to the generation model, we proposed a generalized self-attention based classifier that operates similarly but classifies the documents rather than creating new ones. Furthermore, these classifiers are similar in architecture as they are only different at the very last layer based on the classification task. Notably, we utilized the self-attention based classifiers due to their simplicity of training and performance improvement over prior attempts based on recurrent neural networks or other machine learning approaches. The classifiers had the following layers:

Data Layer: The first step of the classifier is a data layer that splits every sentence from the input documents (clinical texts from EHRs) in the EHR corpus into sequences of tokens or words. Here, each word is assigned or indexed to a unique numeric value through tokenization, where a unique number represents a single token. Then these values are used to convert the sentences into a vector of numeric values. In summary, The data layer converts the entire corpus into a sequence matrix where each sequence is represented with a vector of unique numeric values assigned to each token. In Figure 3.3, $t^{(i)} = \{t_1^{(i)}, t_2^{(i)}, t_3^{(i)}, \ldots, t_n^{(i)}\}$, is the i^{th} sequence in the input matrix and n is the maximum sequence length. For simplicity, we considered the maximum sequence length is equal to the maximum number of tokens in a document in the corpus.

Embedding Layer: Instead of a traditional fixed embedding: Word2vec [MCCD13] and GloVe [PSM14], we used contextualized dynamic word embedding employing a multi-headed self-attention mechanism. As mentioned earlier, the self-attention mechanism was first introduced by Vaswani *et al.* [VSP+17] in 2017. Prominent among the variations of the original transformer model is the BERT model, proposed by Devlin *et al.* [DCLT19]. BERT [DCLT19] uses only the encoder blocks from the transformer model, which could be called transformer-encoders. We used pre-trained weights for the initialization of the transformer-encoder parameters, and during training, the settings are fine-tuned on the training set. The construction details of the encoder blocks are described in Appendix 2.1.3. This encoder block helps to calculate different vector representations for each token based on the use-cases and context. Hence, the name contextualized embedding. Here, the significant difference with the previous approach is this dynamic embedding available for each word.

Output Layer: For this layer, we used a softmax function to calculate the maximum likelihood of the labels (ICD9 codes) of the document. The softmax function

normalizes the values received from the Feed Forward Network (FNN) to probabilities. First, the matrix containing the context information received from the embedding layer is multiplied by a weight matrix and added to a bias in the FNN shown in the output layer of the Figure 3.3. Both the weight matrix and the bias is learned during training. Let $x^{(i)}[l]$ be the output feature matrix of the embedding layer where i is the document number, and l is the encoder block number. b here is the bias vector, and its dimension is $1 \times m$ where m is the number of diseases. The output matrix for the i^{th} document, $y^{(i)}$ is calculated with Equation 3.2.

$$y^{(i)} = W_f^T x^{(i)}[l] + b \tag{3.2}$$

As shown in Figure 3.3, the y matrix calculated in the FNN then forwarded to the softmax sub-layer, where the probability of each disease for the documents are calculated using Equation 3.3. Here as mentioned previously, i refers to the i^{th} document in the dataset and $q_j(y_j^{(i)})$ refers to the predicted probability of j^{th} disease for the i^{th} document.

$$q_j(y_j^{(i)}) = \frac{\exp y_j^{(i)}}{\sum_m \exp y_m^{(i)}} \tag{3.3}$$

The dimension of matrix q is also $1 \times m$, where m is the total number of diseases in the dataset. The network is trained to minimize the cross-entropy loss, $\mathcal{L}(p, q)$ where p is the original probability matrix and q predicted probability matrix for the diseases related to the documents.

For the document-level utility analysis, we used a multi-label-classifier where the ICD9 codes related to the synthetic EHRs are predicted. For the corpus-level utility, we used classifiers with two and three output classes. The details of these two methods are described in the following two subsections.

3.3.2 Utility Evaluation

Word-level Utility

Our first utility metric operates on the token or word level, where we compare the word frequency from the original and generated EHRs. In this micro-level analysis, we employ standard techniques (e.g., , Jaccard similarity, NLL-test, BLEU) to analyze how different the generated texts are. In word-level utility, we mostly analyze the word (/phrase) co-occurrence or similarity, checking whether the generated EHRs have similar token distributions as the original corpora. We use three different technique to do so:

BLEU: We use Bilingual Evaluation Understudy (BLEU) [PRWZ02] as our first evaluation metric for the word level utility. A text generation task could easily be considered as a machine translation task. Thus the use of the BLEU metric, which in the machine translation task determines the closeness of the synthetic translation to one or more reference translations, applies to the synthetic text generation task. In our work, we used BLEU to compare the n-grams (words) of the synthetic documents to the original document and then to count the number of matches. Notably, these matches are position-independent, and more matches result in higher scores (maximum 1). For example, if there are 50% matches for bi-grams ($n = 2$) in both documents, then the BLEU-2 score will be 0.5. The higher the values of the BLEU score, the better the performance.

The centerpiece of the BLEU score is a *precision* measure. This precision measure is naively calculated by simply dividing the total number of n-grams ($n = 1, 2, \ldots, n$) that are present in the reference/original document by the total number of n-grams in the generated document. This naive approach, however, produces a high precision value wrongfully, when any n-grams present in the reference document is generated multiple times. Hence, a more sophisticated approach is adopted in the 'Modified

n-gram Precision' measure, where the total count for the n-grams in the generated text is clipped by maximum count for the n-grams in the reference/original text.

Jaccard Similarity: Jaccard similarity can best be defined as the ratio of the intersection's size by the size of their union. For two text corpora with token vectors O and D (original and de-identified , respectively), Jaccard similarity can be calculated as [Pla14],

$$JaccardSimilarity(O, D) = \frac{O \cap D}{O \cup D}$$

G^2-Test: Lastly, we use G^2 as a frequentist approach for measuring corpus similarity utilizing the standard log-likelihood ratio test [Dun93]. This statistical test represents how close two text corpus is in terms of a specific token [RBF04]. Most importantly, G^2-Test allows us to compare a smaller synthetic dataset with a larger EHR corpus. Here, we consider a certain number of tokens (and n-grams) based on their frequency from both corpora and calculate the log-likelihood value.

To calculate the G^2 value, we need four variables, a, b, c and d. For a specific token (or n-gram) t_i, lets assume it appears a and b times in the original and synthetic dataset, respectively. Now, if c and d are the total number of words in both corpora, then, we retrieve the G^2 value for that particular t_i following the equations below [RG00]:

$$\begin{aligned} E1 &= c * (a + b)/(c + d), \\ E2 &= d * (a + b)/(c + d), \\ G^2 &= 2(a \ln(a/E1) + b \ln(b/E2)) \end{aligned}$$

Here, $E1, E2$ are the expected frequencies of a token in two corpora, which is calculated from a, b, c, and d. Effectively, the G^2 value provides a statistical signif-

icance [LNS+16] of a chosen token's presence or frequency in multiple corpora. For example, if the G^2 value is greater than 3.84 (critical value), then the difference between the two corpora (for that particular token) happening by chance is less than 5% (or $p < 0.05$). In other words, higher values of G^2 denote a statistical significance in the difference between each corpus; therefore, lower scores are preferred.

$$\text{EffectSize} = G^2/(c+d)\ln min(E1, E2) \tag{3.4}$$

On the other hand, lower G^2 values for every token represent the ambiguity of whether two datasets are similar or not. The *effect size* of the tokens is also equally important. In equation 3.4, the effect size is calculated using the G^2 values and it ranges between 0 and 1 (inclusive) [JBMJ06]. This value for any token represents the deviation from observed (a, b) to expected frequencies $(E1, E2)$. Notably, G^2 or the effect size is independent of the corpus size, which is important as the corpora's size is not fixed.

Notably, these three word-level utility metrics provide how close the token distributions are in two corpora. However, it can be misleading in some cases. For example, if a generative algorithm repeats some tokens frequently, it will achieve a higher utility score (i.e., Jaccard). This occurs more frequently for longer documents with a large token count. Therefore, we wanted to avoid the unreliable similarity scores with the following utility metrics.

Document-level Utility

As the purpose of a generated synthetic dataset is to mirror the original sensitive corpus, we expect them to be used in a realistic scenario as well [AD17]. Therefore, we analyze the utility of these synthetic documents using a classification task. Here, the classification task will determine whether the underlying document contains a certain disease or not.

As mentioned earlier in the generation mechanism (Section 3.3.1), we generated the documents according to the underlying disease or conditions. We used the ICD9 specified codes, a standard codebook for each different medical condition as provided from the MIMIC-III dataset [JPS+16]. The dataset contained human annotations of these ICD9 codes for each document representing each visit from the patient. For example, if the patient had Type II Diabetes and Hypertension, then 250.00 and 401.9 codes were flagged as accurate as they represent the diseases, respectively.

Hence, in our document-level utility analysis, we will build different classifiers trained on original and synthetic dataset *separately*. Afterward, they will be tested on different test sets from both corpora. For example, we will train a model with synthetic data and test it on a separate original (along with synthetic) test-set to analyze the utility of the generated data. The outcome of such a test will dictate how well the synthetic corpus will perform in real-life use-cases if we intend to publish them. Similarly, the original data trained model will be tested on synthetic datasets to test their efficacy. We selected seven diseases from the MIMIC-III dataset and built a multi-label classifier on it. The architecture of the classifier model is detailed in Section 3.3.1

Corpus-level Utility

We analyze the generated EHRs on corpus-level containing all EHR documents posing it as an adversarial classification problem [DDSV04]. In this task, we create a dataset having documents from the original and generated EHRs randomly. Then, we label each document as 0 or 1, reflecting whether they are original or synthetic. Then, we train multiple classifiers on these documents predicting whether they belong to the original or synthetic pool.

The purpose of such a classifier is to distinguish between original and synthetic records. This is termed Adversarial Classification (AC), where a machine learning

approach detects the source of the data based on its characteristics [LMS+17]. This is closely analogous to the Turing test, where a human evaluator examines the data and predicts whether it is real or fake. However, we did not opt for medical professionals or utilize any human annotations for the underlying medical texts. Hence, we replaced the human effort with an ML-based solution where a classifier differentiates between the original and synthetic data.

For brevity and better performance, we kept the classification model similar to the architecture discussed earlier in Document-level utility (in Section 3.3.1). Initially, a fixed-size sequence of tokens was taken from arbitrary documents (original/synthetic) and placed as inputs to the embedding layer. The final prediction is obtained from the output layer where we have a fully connected (FC) layer. Here, relative to the earlier approach, the only difference is the final FC layer will have binary or trinary classes, whereas it was the number of diseases for document classification. Notably, these are entirely different classifiers trained separately for both utility analyses.

For AC, we created a dataset with an equal number of EHR documents from the original and generated corpus, and we train four different models. For example, the dataset for AC contained 50% original and synthetic documents, and later, these were randomly split into 80:20 ratio for training and testing. The 20% of testing data had an equal amount of original and synthetic records. The following five settings (four models) were tested:

- **AC1:** We train and test a neural network model with 50% original and synthetic EHRs, each labeled as 0 and 1, respectively. The best adversarial classifier will be able to separate the original from synthetic ones, achieving a 100% accuracy.

- **AC2:** We randomly assign 0 to half of the original dataset and 1 to the other half of the same original dataset. Then we train a different model that has never seen the synthetic data. However, an ideal AC should be able to point the original records correctly and resulting in 50% accuracy.

- **AC3:** In this case, we repeat the procedure with the *generated dataset* and label half of them to 0 and 1 vice versa, randomly. Similarly, the highest accuracy here should be 50% since we did not consider any of the original data.

- **AC4:** In the fourth adversarial classifier, we change the binary classification to a 3-class problem where we consider two input sequences where they are randomly taken from the original and/or the synthetic corpus. Here, if both the sequences are taken from the original dataset, then we label them as 0, whereas two sequences from the synthetic corpus are labeled as 2. If one of the sequences is original, whereas the other collected from synthetic documents randomly, the label is set as 1. Thematically, this classifier will show whether it can identify the source of the EHR texts.

- **AdversarialSuccess (AdvSuc):** We also report the percentage of generated documents that were successful in deceiving the classifier model ($AC1$) and were tagged as the original. Here, we send all documents from the synthetic dataset through the AC1 classifier and report the percentage of the documents that were misclassified as the original. Therefore, the highest value will be 100%, which is desired from the ideal data generator which can fool the adversarial classifier.

The documents which are labeled as 1 in AC2 and AC3 should originally be 0; however, we deliberately mislabeled them to understand the generalizability of these trained adversarial models. We did not consider the incurring losses from these AC models and merge them to the generative networks as this would change the generation technique by biasing towards fooling the ACs rather than producing generalized texts. Thus, none of the utility metrics mentioned here are integrated with the generation network to improve utility performance.

Chapter 4

Results and Discussions

In this chapter, we discuss the datasets along with the experimental setup. We also analyze the utility and privacy of the generated EHRs (experimental results) in Section 4.2.

4.1 Results

We describe the utility of the generated EHRs at three different levels: Word, Document and Corpus. The experimental setup is described next which details the dataset and machine learning parameters:

4.1.1 Experimental Setup

EHR Corpus

We used two datasets that contained PII tokens along with the unstructured medical texts from different patients: i2b2 2014 [SKU15] and MIMIC-III [JPS+16]. The i2b2 dataset was constructed in 2014 as a de-identification benchmark dataset as it categorized the EHRs into the train, validation, and test sets and contained human-annotated PII tokens. Specifically, the i2b2 dataset had the sensitive PII tokens

marked. On the contrary, the MIMIC-III is a large corpus that is available in a de-identified format (without PII), and the PII tokens are not available. These tokens were captured using computational methods [NDL+08], which the data publishers have followed the HIPAA standards [JPS+16].

We split the i2b2 dataset into two different corpora: a) `i2b2-original`, and b) `i2b2-deid`. The first corpus contained all PII tokens as the generated synthetic dataset based on this corpus should also contain similar PII information. As the original i2b2 dataset (resembling real-life medical text) cannot be published publicly, we consider this version of the dataset for utility analysis and comparison. In i2b2-deid, we remove all PII tokens (de-identification) according to the dataset's human-annotated labels as they are considered sensitive information. This version is safe to disseminate, and the generated text from this corpus should fall under the same privacy guarantee as well. Thus, the utility difference from the synthetic texts from i2b2-original and i2b2-died, respectively, will demonstrate the utility loss of de-identification . Notably, in these de-identification procedures, some of the regular tokens and PIIs are also sanitized, which is why the utility comparison is essential. We show that the utility from the generated original and de-identified datasets are similar while the de-identified datasets overcome the privacy constraints.

The i2b2 dataset, however, was much smaller than the MIMIC-III as we looked at 9,817 MIMIC-III discharge summaries (from 2,083,180) whereas i2b2 only had 521 documents. However, the MIMIC-III dataset did not provide the PII labels as it was published originally in a de-identified format. Thus, we only consider one corpus of MIMIC-III. Furthermore, The MIMIC-III dataset contained ICD-9 disease code annotations as we selected seven diseases according to their frequency. These disease classes are hypertension, congestive heart failure, atrial fibrillation, kidney failure, type II diabetes, respiratory failure, and urinary tract infection. These diseases were used as class labels for a supervised task employed in our document-level utility

analysis, as described in Section 3.3.2.

All utility experiments considered the synthetic text data from i2b2-original, i2b2-deid, and MIMIC-III datasets. For the classifiers, each dataset was split 80:20 randomly, where 20% was set as a test set, which was never utilized during the training of the classifiers. We demonstrate one of the synthetic text data in Figure 4.1 generated from the MIMIC-III dataset. As expected, the unique identifiers are also kept de-identified in congruence with the original dataset as MIMIC-III did not contain the PII tokens. Interestingly, our generator did preserve the gender of the patient in the generated texts, but the date of birth, age, and admission date do not add up.

We also generated two separate differentially private corpora from i2b2 and MIMIC-III with ϵ of 23 and 8, respectively. These values were selected as the generated tokens, which were too noisy (or random) for lower epsilon values (more private). The i2b2 dataset had the PII tokens, whereas the MIMIC-III was de-identified . We choose higher ϵ for the i2b2 dataset as it had only 514 records compared to the 9k records of MIMIC-III. Therefore, according to privacy budget analysis outlined by [YJS19], `i2b2-DP` generation required more privacy budget for smaller dataset.

Neural Network Hyper-parameters

Since we utilized different classifiers for the three utility analyses, we kept the neural networks' hyperparameters the same for all settings. For example, the number of iterations was defined by the convergence of the training loss as if the loss was not updated in the last five steps, then the training phase was concluded. We also set the dropout parameter in the final FC layer to be 50% to avoid overfitting.

Furthermore, all training and testing ratios were set as 80:20; given the 20%, of the test set were never revealed to the model while training. However, due to the GPU memory constraints and the inherited classifier design from BERT [DCLT19], we confined the models from inputs of more than 512 tokens on both documents

GENERATED DISCHARGE SUMMARY

Admission Date: [*2180-8-14*] **Service:** MEDICINE

Date of Birth: [*2114-3-16*] **Sex:** M

History of Present Illness: The patient is a 41-year-old man with a past medical history significant for chronic renal insufficiency, known type 1 diabetes, a prior history of stroke, and a recent admission for fungemia secondary to a urinary tract infection, who came in with acute kidney injury and hypertension to the Hospital. He was transferred to the [**Hospital1 18**] on [**2108-6-2**] after he developed acute onset of renal failure with a creatinine of 1.6 on admission. He was transferred to the [**Hospital1 18**] after he developed worsening renal failure with a Cr of 3.0 on admission. On [**2108-6-3**] he was transferred to the [**Hospital1 18**] after developing hypotension with an SBP in the 60s.

Physical Exam: *Vitals:* BP:100/55 P:110 R: 18 O2: 98% on 100% non-rebreather *HEENT:* Sclera anicteric, MMM, oropharynx clear...

Figure 4.1: Sample of a generated EHR text data from MIMIC-III dataset

and adversarial classifiers. However, we did experiment with lower sequence length (i.e., 256, 400), which resulted in poor and inconclusive outputs.

4.1.2 Word-level Utility Result

Jaccard Similarity

At first, we show the utility results in word-level as it is the most granular analysis on the text corpus operated on the token level. Firstly, we use the Jaccard similarity test (equation 3.3.2), which tests how many tokens from the generated corpus match the original one. In other words, we consider the tokens from both corpora as two sets and retrieve the ratio of the intersection over the union (maximum 100%).

From Table 4.1, the Jaccard test shows that our generative method offers higher

Table 4.1: Jaccard Similarity and G2 test on the different synthetic datasets along with benchmark from SeqGAN [YZWY17]

Dataset	SeqGAN [YZWY17]		Our Method	
	Jaccard ↑	G^2-Test ↓	Jaccard ↑	G^2-Test ↓
i2b2-original	69.67	0.83	68.15	1.83
i2b2-deid	75.58	1.3	76.77	1.95
i2b2-DP	-	-	75.04	3.45
MIMIC-deid	82.39	3.81	84.4	2.81
MIMIC-DP	-	-	95.48	5.64

Jaccard similarity on i2b2-deid in comparison to the original i2b2. It is reasonably due to the sensitive PII tokens available in the original corpus, which were not generated accurately in the synthetic ones. For example, if the original corpus contained a patient with the name 'John' getting admitted in 31^{st} December, it is unlikely that the generator outputs the same name and date while producing the synthetic data.

In the generated de-identified corpus, there was no such issue as the name or dates were de-identified already. Furthermore, as the MIMIC dataset was already de-identified , it yielded a better similarity score. The differential private i2b2-DP and MIMIC-DP also showed impressive Jaccard scores as MIMIC-DP had 95.48% similarity. Notably, our Jaccard scores were competitive with the generated text from SeqGAN [YZWY17] as well. SeqGAN also did not have any privacy-preserving component, which is why we do not report the corresponding private results. Details on the SeqGAN's method are discussed in Section 2.4.

G^2-Test

The G^2 statistical test was conducted under the null hypothesis that for an arbitrary token, there is no statistical significance between the token's association with the original or synthetic corpus. In other words, we cannot determine with a high probability that the specific token was taken either from the original or the generated corpus. For this experiment, we considered all tokens from the generated corpus and

calculated the G^2 values and corresponding effect size (according to equation 3.4). In short, smaller G^2 values will denote the uncertainty in a token's membership which represents that the generated data is closer to the original one.

In Table 4.1, we show the average of all token's G^2 values, which is less than the critical value 3.84. Therefore, we *cannot* reject the null hypothesis for our method in all three corpora (95 percentile). Furthermore, the MIMIC synthetic dataset had a slightly higher G^2 value due to its corpus size, although the effect size of each token, in this case, was much smaller compared to i2b2.

Furthermore, analogous to Jaccard scores, our results are similar to the earlier approach [YZWY17] in all settings as well. However, contradictory to these two utility measures, the BLEU scores will show a significant difference between these methods.

The G^2 scores for i2b2-DP and MIMIC-DP were 3.45 and 5.64, respectively. However, i2b2-DP's score was less than the critical value for 95 percentile (p¡0.05), MIMIC-DP's score was not. Therefore, there is a significance between the word frequencies of the original MIMIC-III and MIMIC-DP. However, this difference is smaller than the critical value for the 99 percentile (p¡0.01), which is 6.63. We did not experiment with higher ϵ values that might further reduce the G^2 scores.

BLEU Score

In Table 4.2, the BLEU scores ($B-n$) of our proposed method is presented. We considered the $n = \{1, 2, 3\}$-gram values for the BLEU metric as we outperformed the existing SeqGAN [YZWY17] for i2b2 dataset on all three n values. Furthermore, as $n = 3$ considers three tokens appearing together, whereas $n = 1$ only takes the occurrence of a token, it is easier to achieve higher B-1 scores compared to B-3. Our experimental results are consistent with this finding, as all B-3 scores are smaller than the B-1s in all settings or corpus.

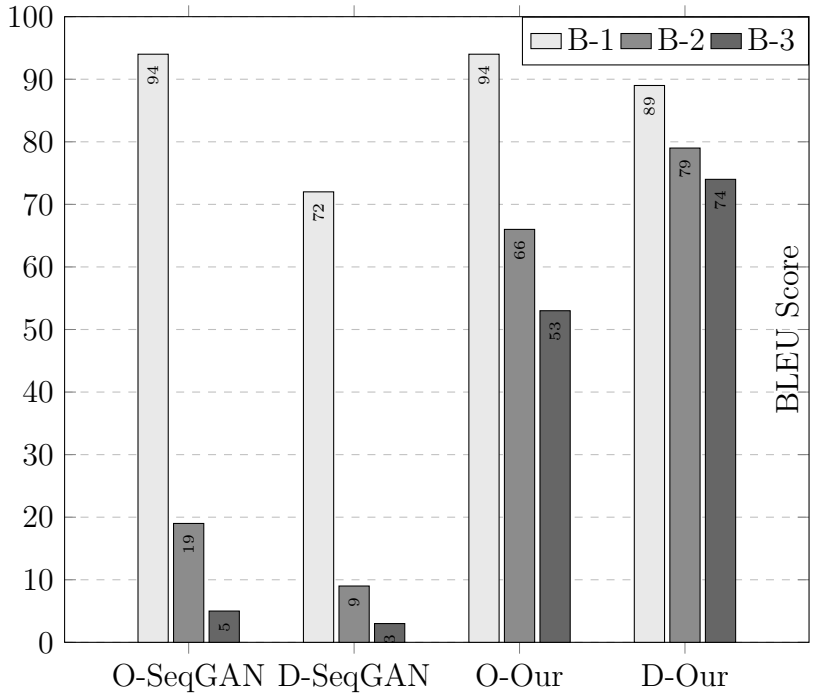

Figure 4.2: BLEU score (B-1, B-2, B-3) comparison on i2b2 dataset with our and SeqGAN model (from Table 4.2)

For example, the B-1 scores of SeqGAN and our approaches are head to head, whereas the difference in the B-2 and B-3 scores is clear. It denotes that SeqGAN [YZWY17] can generate single tokens from the input corpus but cannot output the consecutive bi-grams or tri-grams (phrases) correctly. As a result, the B-2/3 scores are significantly low. However, we see promising BLEU scores for $n = 2, 3$ in our transformer-based generative method. In summary, these BLEU scores show the utility difference of our approach compared to earlier work in a different dataset.

4.1.3 Document-level Utility Result

The document-level utility was measured using the ICD9 codes available in the MIMIC-III dataset. Since we generated the synthetic medical notes pre-conditioned of the seven different diseases, we employ a classifier to check whether the trained models perform similarly on each dataset. For example, we will train a model on the

Table 4.2: BLEU-$\{1, 2, 3\}$ scores on i2b2 and MIMIC-III dataset comparing SeqGAN [YZWY17] with our approach

Dataset	SeqGAN [YZWY17]			Our Method		
	B-1	B-2	B-3	B-1	B-2	B-3
i2b2-original	0.94	0.19	0.05	0.94	0.66	0.53
i2b2-deid	0.72	0.09	0.02	0.89	0.79	0.74
MIMIC-deid	0.98	0.34	0.06	0.98	0.78	0.48
i2b2-DP	-	-	-	0.91	0.73	0.67
MIMIC-DP	-	-	-	0.99	0.88	0.73

Table 4.3: Disease classification accuracy on different MIMIC-III dataset and varying number of diseases

Train / Test	Original			Synthetic/DP			Mix		
# of Disease	3	5	7	3	5	7	3	5	7
Original	*97.2*	83.6	82.4	84.5	75.9	*69.9*	77.6	74.7	72.8
Synthetic	80.4	75.4	68.1	98.1	91.2	*82.3*	89.8	82.1	74.9
DP	91.2	89.5	86.6	98.5	96.1	95.6	89.1	84.1	78.5
Mix	70.8	65.4	61.9	70.1	65.9	61.3	88.1	71.9	70.7

original or synthetic dataset and test on the synthetic or original corpus to test the interoperability of the classifiers. Here, we selected (randomly) 7k documents where each disease had 1k documents and was split into 80:20 train, test-set.

We also vary the number of diseases among $\{3, 5, 7\}$ to check the scalability of the classifier's accuracy in terms of class labels. Furthermore, we also created a mixed dataset, containing 50% original, and 50% generated medical texts and repeated the procedure. The train and test ratio was set as 80:20, as mentioned earlier in Section 4.1.1, and a sequence length of 512 was considered due to our GPU's memory limitation. It is noteworthy that the i2b2 corpus did not present the ICD-9 codes. Thus we only opted for the MIMIC-III dataset for this utility analysis.

In Table 4.3, we show the results as the columns represent the training dataset, whereas the rows have the testing ones. The Mix test dataset was completely different than the Original or Synthetic ones and was randomly chosen. We can observe that the accuracy actually decreases with the number of diseases for all cases. For example,

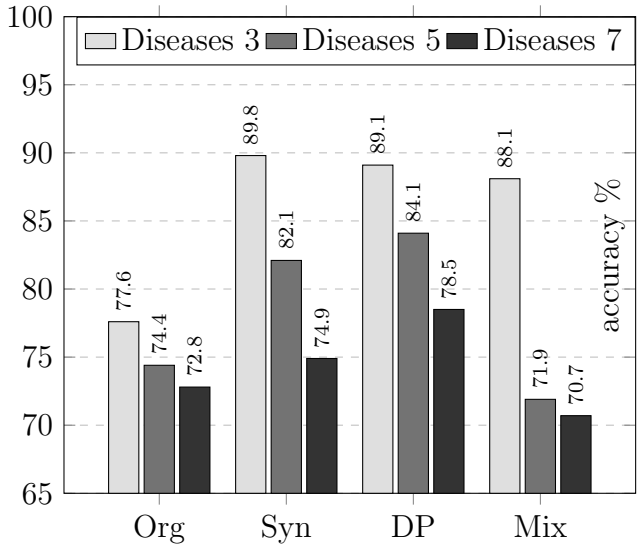

Figure 4.3: Disease classification accuracy where the classifier was trained on mixed dataset (from Table 4.3)

for three disease classification, a model trained and tested on Original MIMIC-III corpus results in 97.2% accuracy, whereas it decreases to 82.4% for seven diseases.

On the other hand, the model trained on one dataset under-performs while being tested on the other one. Here, a model trained on the synthetic dataset gained 69.9% accuracy for all seven disease classes, whereas it achieved 82.3% on its own test set. However, the drop in accuracy is not significant, which underscores the strength of the generation approach. For a lower number of classes (three diseases), the performance drop is minor.

From Figure 4.3, it is clear that the Mix dataset training favours the Synthetic test-set as it outperforms the original one. Also, it holds better accuracy for the Mix test-set as well. The generated data from the Differentially Private (DP) generator performed better in analogous test sets, whereas the accuracy dropped in Original and Mixed test sets. However, these privately generated sets were comparable with their non-private counterparts.

In summary, we see a general trend of higher accuracy for the same dataset used for training and testing. However, the experiment's motivation was to check how

Table 4.4: Corpus-level utility test using adversarial classifier on i2b2 and MIMIC-III dataset

Dataset	Type	AdvSuc	AC1	AC2	AC3	AC4	AvgAC
i2b2	with PII	83.7	37.3	44.1	30.4	51.3	40.8
	deid	86.9	43.2	42.1	51.9	45.8	45.7
MIMIC	deid	92.3	54.6	51.6	52.4	52.5	52.8
MIMIC-DP	deid	93.4	52.3	52	50.3	51	51.4

the classifiers perform on other datasets that it has a limited idea about (knowledge gained while generation). It seems that the accuracy results on each document are comparable, and our synthetic corpus can be utilized in place of real-world datasets to build such classifiers.

4.1.4 Corpus-level Utility Result

In Table 4.4, we show the corpus-level utility of the synthetic data employing the adversarial classifiers (ACs). There four different ACs, as described in Section 3.3.2, each emphasizing different properties of the data. In summary, we want to analyze whether an ML classifier can identify the synthetic records from the original ones.

As the maximum accuracy from the ideal AC1 should be 100% denoting that it can correctly distinguish the fake ones, the AC1 on our synthetic corpus resulted in less than 60% accuracy for all datasets. Also, AC2 and AC3 are lopsided towards a specific corpus (original or synthetic) and only get information from that corpus while training; hence, they can only achieve a maximum of 50%. Nevertheless, AC2 and AC3 accuracy values are around 50% for all cases, which is interesting. For example, it leads to the fact that AC2 or AC3 for the MIMIC-III dataset can identify original or synthetic records while only trained with one dataset. However, if presented with both data (AC1), it performs poorly.

Nevertheless, these AC1, 2, 3 are binary classifiers resulting in 50% accuracy, which resembles the coin toss probability over a large sample size (1000 documents).

Therefore, we test AC4 with 3 class outputs, which is more complicated than these classifiers (AC1-3) where the accuracy values should reflect the ability of the AC to understand the relation between original and synthetic data. Notably, the worst-case accuracy, in this case, is 33.33%; albeit, an ideal AC4 will have 100% accuracy as it can correctly identify whether the two sequences as inputs are incoming from the same corpus or the opposite. Also, if they are sourced from the same dataset, then it should be able to forecast which one was it.

We see that the AC4 values are above the minimum 33.3% in all cases, signifying that the classifier can distinguish some documents. However, the values are not significant enough to conclude that the AC4 successfully separated the synthetic and original records. Nevertheless, AC4's performance was better than the other three classifiers.

The Adversarial Success (AdvSuc) was tested only on the AC1 as sequences from all synthetic records were tested. For example, in this synthetic corpus from MIMIC-III, we used 7k generated records (1k per disease for equality), which were tested, and 92.3% (6461 documents) were predicted as original. The differentially private synthetic sets also performed satisfactorily as the adversarial success was 93.4%.

4.2 Discussion

4.2.1 Utility of Synthetic Data

In Section 4.1, the utility of the novel synthetic data was analyzed from three viewpoints. Firstly, the three word-level analysis demonstrated how similar the generated corpus is compared to the original ones. Interestingly, among the three tests, Jaccard (Section 4.1.2) and G^2-test (Section 4.1.2) do not reveal the difference between our approach and preceding SeqGAN [YZWY17]. It seems the Jaccard indices are almost similar (in one case larger for i2b2-original), so are the G^2 values.

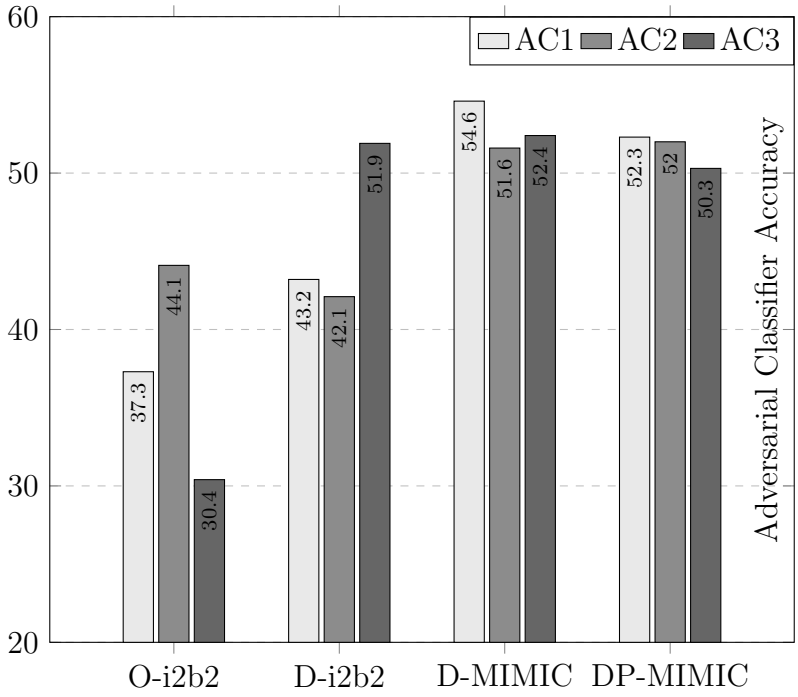

Figure 4.4: Adversarial Classifiers on i2b2 and MIMIC-III datasets with different generation mechanism (from Table 4.4)

However, these Jaccard and G^2-tests used only one unique word from the entire corpus. For example, if the generator outputs some specific or more frequent words (i.e., fever, admission, date) *repeatedly*, it might achieve a higher similarity score. This would misdirect the efficacy of our method. Therefore, these two tests are often inconclusive while considering single tokens. Nevertheless, the BLEU score for $n = 2, 3$, where two or three consecutive tokens are considered, demonstrates the difference with SeqGAN as our scores are significantly higher.

For disease (Section 4.1.3) and adversarial classifier (Section 4.1.4), the results demonstrate similar utility signatures from synthetic and original data. As shown in Table 4.3, the downtrend of classification accuracy from seven diseases to three was observed for both datasets. We expected that the model trained on mixed data would attain the highest for both original and synthetic corpus, albeit which was not observed. Interestingly, the synthetic data model performs satisfactorily on the original corpus while matching the accuracies on the Mixed test set. A similar trend

is also present in the model trained with the original data showing the interoperability of the data. We investigated this as a case of overfitting and rigorously tested with unknown random test datasets, standard regularizers, and a high dropout of 50%.

Noticeably, we observed a downward trend in terms of classification accuracy while increasing the number of diseases in all document-level experiments. For example, for the original train and test MIMIC-III dataset [JPS+16], the accuracy lowered from 97.2% to 82.4%. We agree that such classifiers need to be more precise for a larger quantity of disease classes but we employed a simple classifier as our focus in this work was not to develop a state of the art document classifier rather analyze the utility of the synthetic EHRs A more specialized classifier that groups multiple diseases (based on ICD codes) for classification may be more accurate. Nevertheless, improving accuracy for a large number of diseases can be interesting future work.

Another observation from the adversarial classifier in Section 4.1.4 is the percentage of adversarial success. For example, the synthetic data from MIMIC-III showed an impressive 92.3% success, which is higher than the i2b2 dataset. This is due to the size of the input corpus as i2b2 had only 241 documents, whereas it was around 10k for MIMIC-III. Therefore, a larger corpus produced much generalized synthetic data, which deceived the AC in attaining a better overall performance.

Prior to benchmarking with SeqGAN [YZWY17], several other methods (i.e., LeakGAN [GLC+18], JSDGAN [LMS+17] etc.) were put to test; but they fell short for our use-case, generating long medical texts. SeqGAN performed reasonably and, more importantly, was faster to train and generate a sizeable corpus. The rest of the methods were slower as they relied on slower training algorithms and may not be suitable for a dataset like MIMIC-III (around 10k records with at least 512 tokens each). Also, it is important to note that GPT-2 [RWC+] and its successor GPT-3 [BMR+20] are the state-of-the-art methods in text generation. We inherit GPT-2 model, adding the privacy notions and expect it to outperform these other generation

techniques.

4.2.2 Privacy of the Synthetic Data

In this work, we did not use any automated PII de-identifiers [DLUS16, Gar15]. We chose to avoid such tools partially due to their accuracy and the ultimate privacy guarantee of the synthetic data. As the accuracy of these de-identification techniques does not adequately identify all PIIs from long medical texts, we cannot guarantee that the generated texts from such de-identification will not contain any private information. Inherently, the synthetic data from such de-identified corpus will not be safe to publish.

We also utilized a differentially private data generator that can take arbitrary text input and produce medical texts with an ϵ privacy guarantee. Therefore, this technique can be utilized on private raw datasets without any de-identification . However, the output corpus may still contain identifying information about a certain group or individuals as the DP generation does not guarantee to completely remove all identifying information or prevent potential privacy or linkage attacks on the synthetic dataset.

An optimal solution would be to de-identification the synthetic corpus, using the state-of-the-art automated tool. However, to analyze the accuracy of such de-identification on the synthetic dataset will require the PII labels for each token. Such annotation is only possible by human efforts as expert annotators can separate the PII tokens (in the synthetic dataset), which can be sanitized before publishing. However, in the future, with an acceptable de-identification tool, we can sanitize the synthetic data generated from the original data, which should provide the same privacy as guaranteed by the de-identifier tool.

In contrast, we use the PII annotations retrieved from the datasets and sanitize accordingly (for non-DP generation). Notably, this mechanism is safe as both datasets

as i2b2 and MIMIC-III were published with the PII annotations or de-identified , respectively. Therefore, it is safe to assume that the generated data will hold the same privacy guarantee as the de-identified data, which should be the maximum in our case. Furthermore, we can create an arbitrarily large number of records from a small subset, which is representative of that de-identified corpus.

One interesting question to answer would be: What does the differential privacy step solve for an imperfect de-identification method? For example, if we use the current state-of-the-art de-identification models providing 98% recall, does the extra DP step solves the privacy issue? This is one of our major motivations behind using DP while training as it can offer quantifiable privacy on the data, regardless of the input. Even with this higher percentage, the de-identification models sill may leave traces of PII tokens that may come out from the generator. The use of DP during generation will avoid this by creating a measurable uncertainty on the participant's presence in the corpus. This uncertainty will create an extra layer of privacy, which will minimize the privacy leakage caused by the imperfect de-identification model.

4.2.3 Limitations

In this work, the utility of the texts is tested with various ML applications or standard NLP techniques. However, determining the privacy of these generated data needs human expertise. For example, we can employ healthcare or privacy professionals to check whether there is any personal information in the generated corpus. Only then can we be sure about the privacy of the documents. Unfortunately, we did not target such a user study in this thesis.

Health-care professionals can also decide whether the data is synthetic or real, which resembles our adversarial classifier. Such human endeavour can reveal the potential flaws and possibly make the generator more realistic. Furthermore, this can be coupled with the de-identification task using human efforts as the PIIs can be

identified alongside the source of the texts. However, we think such user studies for EHRs, combined with the aforementioned one, can be fruitful and are an important future direction.

Moreover, we acknowledge that the utility of these texts is always use-case-specific as it cannot be measured with a wide variety of testing. For example, there will always be a possibility for some other applications where these generated data falls short. Also, researchers usually prefer a dataset without any noise happening from a DP mechanism or keep the noise to be minimal due to the inherent implications in utility. We did not experiment or explore this privacy-utility relation as it was exhaustive due to the several ML training procedures (i.e., , generation, utility classifiers) that were necessary to get the complete picture.

The de-identified inputs to the generator impose another limitation as the generated data will only contain the statistical properties and traits of this input subset. Therefore, a small de-identified corpus as input might not represent the whole dataset. As the accuracy of the current de-identification models reports their accuracy over 97%, it may seem reasonable to use these models to de-identify larger corpus and then use these corpora to generate more sophisticated synthetic EHRs. However, the 'accuracy' of a de-identification model entails how accurately the model can predict whether a word/token in the corpus is a PII or not. This metric has its benefits in the overall performance evaluation of a de-identifier. However, in a de-identification task, the model's recall value is often more useful to evaluate the performance. The recall value reveals the percentage of PII instances the model will let go as *not sensitive* (False Negatives). Current state-of-the-art de-identification models to date have recall value around 98%. This value may appear sufficient to apply in real-world applications; however, a closer analysis of the implication of 2% error at the recall value reveals the inadequacy of these de-identification models. For a corpus with millions of PII tokens, a 2% error at the recall value will let go of thousands of sensitive tokens.

Consequently, if we use a larger corpus de-identified by these models, the synthetic EHRs generated will inadvertently include real sensitive information about the participants of the said corpus. Moreover, when using a differentially private generation, it is not possible to delineate a quantitative analysis on potential risk or leakage of the sensitive information.

Chapter 5

Conclusion

5.1 Summary

In this thesis, we propose a method to publish clinical textual data, moving away from the traditional de-identification systems. Here, we propose a privacy-preserving synthetic text generator that addresses the privacy concerns of sharing medical text data. To the best of our knowledge, this is one of the first works that proposed differentially private medical text generation techniques and analyzed their applications and usability.

Initially, we discussed the motivation for such a generative approach to attain privacy of the textual data. To generate the medical texts, we relied primarily on the self-attention models which were faster to train. We altered the training mechanism to incorporate differential privacy and experimented with different EHR corpus. We also targeted the utility of such synthetic data as they are scrutinized over different metrics. The generated data is compared with the original datasets and benchmarked against other techniques such as SeqGAN [YZWY17] on three different settings. The experimental results demonstrate an almost indistinguishable performance between the original and synthetic data (private and non-private) supporting the effectiveness

of these generated free-texts. For example, these synthetic datasets achieve more than 80% accuracy in real-life disease classification problems and match the performance of original text data. Furthermore, our synthetic dataset surpassed the state-of-the-art generation technique on every utility metric.

However, in this work, we did not consider (or propose) a de-identification technique for EHRs, which is also an important research area in medical texts. We argue that our proposed method can potentially allow public dissemination of texts as they are probabilistic. In the future, any state-of-the-art de-identification tool can be easily integrated into the pipeline, making the synthetic data de-identified as well.

5.2 Future Work

Oftentimes, it proves hard to gather meaningful information from the long EHR texts that are recorded throughout patient care. A summary from these healthcare data can be a great alternative that can be generated alongside our proposed method. Here, the summarized texts can contain the key concepts such as the vitals, diagnostics or medications. Further, question-answering or similar patient search can be performed on these textual EHR datasets which can be investigated as well.

We only tested our synthetic dataset's utility for a specific task (disease classification). In future, these artificial datasets can be analyzed for more demanding and real-life tasks that are common in the healthcare domain as the medical utility of these data should be analyzed more carefully. For example, we can use medical professionals to distinguish between real and synthetic data which can reveal the efficacy of the generation process. Furthermore, these synthetic datasets need to be used on several computational tasks that can provide us with more comparison points with their original counterpart.

Lastly, the privacy component of these generated texts needs to be analyzed.

We have used a differential private generation mechanism that allows us to measure the privacy loss, from a theoretical standpoint. However, further linkage studies can potentially reveal whether the data is truly safe if it can be linked to separate datasets. Also, the outputs from the state-of-the-art de-identification methods [AAM20, DLUS16] relying on machine learning techniques can give us more insight into the sensitive PII elements present on the newly generated texts. Overall, the artificial medical text seems to be a promising parallel direction to the traditional de-identification oriented data publishing method.

REFERENCES

[AAM20] Tanbir Ahmed, Md Momin Al Aziz, and Noman Mohammed. De-identification of electronic health record using neural network. *Nature Scientific Reports*, 10(1):1–11, 2020.

[ABY+10] John Aberdeen, Samuel Bayer, Reyyan Yeniterzi, Ben Wellner, Cheryl Clark, David Hanauer, Bradley Malin, and Lynette Hirschman. The MITRE identification scrubber toolkit: Design, training, and assessment. *International Journal of Medical Informatics*, 79(12):849–859, 2010.

[ACG+16] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[AD17] H. Alemzadeh and M. Devarakonda. An NLP-based cognitive system for disease status identification in electronic health records. In *Proceedings of the 2017 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, pages 89–92, 2017.

[Ann03] George Annas. HIPAA regulations-A new era of medical record privacy? *The New England Journal of Medicine*, 348(15):1486–1490, 2003.

[ASA+19] Md Momin Al Aziz, Md Nazmus Sadat, Dima Alhadidi, Shuang Wang, Xiaoqian Jiang, Cheryl L Brown, and Noman Mohammed. Privacy-preserving techniques of genomic data: A survey. *Briefings in Bioinformatics*, 20(3):887–895, 2019.

[BDVJ03] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(2):1137–1155, 2003.

[BMR+20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *https://arxiv.org/abs/2005.14165*, 2020.

[BVV+16] Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, 2016.

[CCG15] Tao Chen, Richard M Cullen, and Marshall Godwin. Hidden Markov model using Dirichlet process for de-identification. *Journal of Biomedical Informatics*, 58:S60–S66, 2015.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language un-

derstanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186, 2019.

[DCR+04] M. Douglass, G. D. Clifford, A. Reisner, G. B. Moody, and Mark RG. Computer-assisted de-identification of free text in the MIMIC II database. In *Proceedings of the 2004 IEEE Computers in Cardiology*, pages 341–344, 2004.

[DCR+05] M. M. Douglass, G. D. Cliffford, A. Reisner, W. J. Long, G. B. Moody, and R. G. Mark. De-identification algorithm for free-text nursing notes. In *Proceedings of the 2005 IEEE Computers in Cardiology*, pages 331–334, 2005.

[DDSV04] Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 99–108, 2004.

[DKBB19] Caitlin Dreisbach, Theresa A Koleck, Philip E Bourne, and Suzanne Bakken. A systematic review of natural language processing and text mining of symptoms from electronic patient-authored text data. *International Journal of Medical Informatics*, 125:37–46, 2019.

[DLUS16] Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606, 2016.

[Doe16] Carl Doersch. Tutorial on variational autoencoders. *https://arxiv.org/abs/1606.05908*, 2016.

[DR13] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Theoretical Computer Science*, 9(3-4):1–277, 2013.

[Dun93] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[Dwo06] Cynthia Dwork. Differential privacy. In *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II*, pages 1–12, 2006.

[EHR17] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional GANs. *https://arxiv.org/abs/1706.02633*, 2017.

[FGD18] William Fedus, Ian Goodfellow, and Andrew M. Dai. Maskgan: Better text generation via filling in the _____. *https://arxiv.org/abs/1801.07736*, 2018.

[Gar15] Simson Garfinkel. De-identification of personal information. `https://doi.org/10.6028/NIST.IR.8053`, 2015. Accessed: April 11, 2021.

[GK96] Christoph Goller and Andreas Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of 1996 International Conference on Neural Networks*, volume 1, pages 347–352, 1996.

[GLC+18] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. Long text generation via adversarial training with leaked information. In *Proceeding of the 32nd AAAI Conference on Artificial Intelligence*, pages 1–14, 2018.

[GLYZ18] J. Guan, R. Li, S. Yu, and X. Zhang. Generation of synthetic electronic medical record text. In *Proceedings of the 11th IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 374–380, 2018.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 2014 Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[IVK+20] Julia Ive, Natalia Viani, Joyce Kam, Lucia Yin, Somain Verma, Stephen Puntis, Rudolf N Cardinal, Angus Roberts, Robert Stewart, and Sumithra Velupillai. Generation and evaluation of artificial mental health records for Natural Language Processing. *NPJ Digital Medicine*, 3(1):1–9, 2020.

[JBMJ06] Janis E Johnston, Kenneth J Berry, and Paul W Mielke Jr. Measures of effect size for CHI-squared and likelihood-ratio goodness-of-fit tests. *Perceptual and Motor Skills*, 103(2):412–414, 2006.

[JPS+16] Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, 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(1):1–9, May 2016.

[KST20] Gavin Kerrigan, Dylan Slack, and Jens Tuyls. Differentially private language models benefit from public pre-training. *https://arxiv.org/abs/2009.05886*, 2020.

[KW14] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *https://arxiv.org/abs/1312.6114*, 2014.

[Lee18] Scott H Lee. Natural language generation for electronic health records. *NPJ Digital Medicine*, 1(1):1–7, 2018.

[LMG+20] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. BERT-attack: Adversarial attack against BERT using BERT. *https://arxiv.org/abs/2004.09984*, 2020.

[LMS+17] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *https://arxiv.org/abs/1701.06547*, 2017.

[LNS+16] Jefrey Lijffijt, Terttu Nevalainen, Tanja Säily, Panagiotis Papapetrou, Kai Puolamäki, and Heikki Mannila. Significance testing of word frequencies in corpora. *Literary and Linguistic Computing*, 31(2):374–397, 2016.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *https://arxiv.org/abs/1301.3781*, 2013.

[MRTZ17] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *https://arxiv.org/abs/1710.06963*, 2017.

[MS19] Oren Melamud and Chaitanya Shivade. Towards automatic generation of shareable synthetic clinical notes using neural language models. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 35–45, 2019.

[NDL+08] Ishna Neamatullah, Margaret M Douglass, H Lehman Liwei, Andrew Reisner, Mauricio Villarroel, William J Long, Peter Szolovits, George B

Moody, Roger G Mark, and Gari D Clifford. Automated de-identification of free-text medical records. *BMC Medical Informatics and Decision Making*, 8(1):1–17, 2008.

[NG06] Rachel Nosowsky and Thomas J Giordano. The Health Insurance Portability and Accountability Act of 1996 (HIPAA) privacy rule: Implications for clinical research. *Annual Review of Medicine*, 57:575–590, 2006.

[NHDC20] Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. Toward robustness and privacy in federated learning: Experimenting with local and central differential privacy. *https://arxiv.org/abs/2009.03561*, 2020.

[Nis07] Josh Nisker. PIPEDA: A constitutional analysis. *The Canadian Bar Review*, 85(2):317–343, 2007.

[OCP+05] Kimberly J O'malley, Karon F Cook, Matt D Price, Kimberly Raiford Wildes, John F Hurdle, and Carol M Ashton. Measuring diagnoses: ICD code accuracy. *Health Services Research*, 40(5):1620–1639, 2005.

[PAE+16] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *https://arxiv.org/abs/1610.05755*, 2016.

[PG18] Nicolas Papernot and Ian Goodfellow. Privacy and machine learning: Two unexpected allies? `http://www.cleverhans.io/privacy/2018/04/29/privacy-and-machine-learning.html`, 2018. Accessed: April 21, 2021.

[Pla14] C. Plattel. *Distributed and incremental clustering using shared nearest neighbours*. PhD thesis, Utrecht University, 2014.

[PMSW18] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P Wellman. Sok: Security and privacy in machine learning. In *Proceedings of the 3rd IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 399–414, 2018.

[PRWZ02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on Association for Computational Linguistics*, pages 311–318, 2002.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GLOVE: Global vectors for word representation. In *Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

[PSM+18] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *https://arxiv.org/abs/1802.08908*, 2018.

[RA15] Chandan K Reddy and Charu C Aggarwal. *Healthcare data analytics*. Chapman and Hall/CRC, 2015.

[RBF04] Paul Rayson, Damon Berridge, and Brian Francis. Extending the Cochran rule for the comparison of word frequencies between corpora. In *Proceedings of the 7th International Conference on Statistical Analysis of Textual Data*, pages 926–936, 2004.

[RG00] Paul Rayson and Roger Garside. Comparing corpora using frequency profiling. In *Proceedings of the 2000 Workshop on Comparing Corpora*, volume 9, pages 1–6, 2000.

[RM15] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *https://arxiv.org/abs/1505.05770*, 2015.

[Rud16] Sebastian Ruder. An overview of gradient descent optimization algorithms. *https://arxiv.org/abs/1609.04747*, 2016.

[RWC+] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. `https://github.com/openai/gpt-2`. Accessed: April 11, 2021.

[SC99] Fei Song and W Bruce Croft. A general language model for information retrieval. In *Proceedings of the 8th International Conference on Information and Knowledge Management*, pages 316–321, 1999.

[SCS13] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248, 2013.

[SKU15] Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/UTHealth shared task track 1. *Journal of Biomedical Informatics*, 58:S11–S19, 2015.

[SS15] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1310–1321, 2015.

[SSSS17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Pro-*

ceedings of the 2017 IEEE Symposium on Security and Privacy (SP), pages 3–18, 2017.

[STBR17] 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, 2017.

[TOU+18] F Toscano, E O'Donnell, MA Unruh, D Golinelli, G Carullo, G Messina, and LP Casalino. Electronic health records implementation: Can the European Union learn from the United States? *European Journal of Public Health*, 28:213–401, 2018.

[US15] Özlem Uzuner and Amber Stubbs. Practical applications for natural language processing in clinical research. *Journal of Biomedical Informatics*, 58(5):S1–S5, 2015.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 2017 Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008, 2017.

[WRD+19] Stephen Wu, Kirk Roberts, Surabhi Datta, Jingcheng Du, Zongcheng Ji, Yuqi Si, Sarvesh Soni, Qiong Wang, Qiang Wei, Yang Xiang, Bo Zhao, and Hua Xu. Deep learning in clinical natural language processing: A methodical review. *Journal of the American Medical Informatics Association*, 27(3):457–470, 2019.

[XCS18] Cao Xiao, Edward Choi, and Jimeng Sun. Opportunities and challenges in developing deep learning models using electronic health records data:

A systematic review. *Journal of the American Medical Informatics Association*, 25(10):1419–1428, 2018.

[YG15] Hui Yang and Jonathan M Garibaldi. Automatic detection of protected health information from clinic narratives. *Journal of Biomedical Informatics*, 58:S30–S38, 2015.

[YJS19] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *Proceedings of the 2019 International Conference on Learning Representations*, pages 1–21, 2019.

[YPM20] Vithya Yogarajan, Bernhard Pfahringer, and Michael Mayo. A review of automatic end-to-end de-identification: Is high accuracy the only metric? *Applied Artificial Intelligence*, 34(3):251–269, 2020.

[YVET17] Alexandre Yahi, Rami Vanguri, Noémie Elhadad, and Nicholas P Tatonetti. Generative adversarial networks for electronic health records: A framework for exploring and evaluating methods for predicting drug-induced laboratory test trajectories. *https://arxiv.org/abs/1712.00164*, 2017.

[YZWY17] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. SeqGAN: Sequence generative adversarial nets with policy gradient. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 2852–2858, 2017.

[ZW11] Jiajie Zhang and Muhammad Walji. TURF: Toward a unified framework of EHR usability. *Journal of Biomedical Informatics*, 44(6):1056–1067, 2011.