
University of Witwatersrand

School of Computer Science and Applied Mathematics

# De-identification of textual clinical-notes using Recurrent Neural Networks

[ A dissertation submitted in partial fulfillment of the requirements for the Masters of Science degree ]

*Student:*
Peter Mtungwa
(1259477)

*Supervisor:*
Prof.Turgay Çelik

August 20, 2020

# Abstract

Every visit to a clinician or healthcare practitioner's office leads to the generation of medical records for the patient in question. Such provides opportunity for increasingly usable digital medical information on complementary implementations, like biomedical studies, that could allow quicker and thus more participatory research and at the same time safeguarding healthcare services recipients confidentiality by providing adequate and effective safeguard mechanism to protect protected health information (PHI). However, due to legal constraints aiming to safeguard the confidentiality of patients, only a few of clinical investigative students can only access PHI-*free* medical notes. A general framework for providing necessary patient confidentiality together with 18 types of protected health information (PHIs) that are to be "scrubbed" from de-identified healthcare services recipients notes, is provided by the Health Insurance Portability and Accountability Act (HIPAA) in the United States before these notes and records containing PHIs can be accessed by others or by organizations.

In South Africa, the disclosure of protected health information is governed by Protection of Personal Information Act (POPI Act or POPIA). The Protection of Personal Information Act No. 4 of 2013 encourages the protection and prevent unauthorized disclosure by public and private institutions containing personally identifiable information. The South African government ratified the POPIA Act on 19 November and released Notice 37067 on 26 November 2013 in the Government Gazette. The shared de-identified ("PHI-*free*") medical information records could help in improving care, estimating medical care costs, and supporting policies on human health. There are innumerable examples that support the need to utilize de-identified records.

Clinical text de-identification offers such advantages, however, if performed manually, then such process typically becomes a tediously cumbersome process and prone to errors. Employing deep-learning methods and techniques coupled with methods for Natural Language Processing (NLP) techniques could mitigate the difficulties and make this process smoother. However, I should point out that there has only been seldom any investigation into further use of automatically de-identified medical narration. In doing this, a dependable computerized "PHI removal and sanitizing" system will consequently be of great importance in carrying out this task and vast amount of data contained in medical notes will be made accessible for further research and uses. In this research, I have studied machine learning methods to address the problem of de-identification of textual clinical notes.

I examined various options in order to develop methods that can utilize gradient-based algorithms and/or deep learning methods that will achieve accurate and robust Text de-identification of the Clinical notes. In the process, I developed the de-identification LSTM_CRFs model that could fully utilize a clinical corpus which was developed for the 2014 i2b2 challenge and the 2016 CEGS NGRID. Furthermore, I evaluated the performance using clinical notes collected and compared the performance. The results obtained in this study are further compared to five different word embeddings trained from the general English text, de-identified clinical text, and biomedical literature. The developed LSTM_CRFs model for de-identification achieved superior performance compared with other ML-based methods.

# Contents

# List of Figures

# List of Tables

# List of Appendices

# Declaration

I, Peter Joseph Mtungwa, (*Student Number* **1259477** at the University of Witwatersrand, Johannesburg, Republic of South Africa) hereby declare that the contents of this Research Proposal to be my own work.

I am aware and understand that permission must be sought, acknowledged, explicitly granted, and properly cited at any time or when using third party's work (journals, books, newspapers, etc), ideas, and material. This proposal is submitted for the degree of Masters of Science in Computer Science at the University of the Witwatersrand, Johannesburg.

Signature:

August 20th 2020

Date:

# Acknowledgements

I wish to thank Prof.Turgay Çelik for providing advice and guidance on the content of this research proposal, and the background in various concepts in this area of research. Also the University of Witwatersrand Library staff for providing great and invaluable customer support to ensure that I get access to most of the sources that I used to conduct my research.

**A very special thanks to:**

**Justina, Jonathan, Francisca, Peter, and Jackson**
**you are everything to me..!!**

**Thank you for your love and support; without which nothing would have been possible.**

# Chapter 1

# Introduction

There is increasing concern with regards to the individual privacy issues revolving around the procured personal information by the government and non-governmental entities and/or just about any employer. Many, if not all of these entities, collect, transmit, store information. Hence the concerns at individual level is on access, utilization, and permitted disclosing of private information. Most of the time, the collected information is used for commercial gain by many organizations. Security concerns have a major impact on customer readiness to participate in digital transactions.

The increase of commercial transaction and activities necessitates sharing and exchange of information but when such access or sharing of information is unauthorized, then it turn to incidents of privacy invasion whereby an individual or entity is affected. It is important to note that privacy invasion came in many forms. However, many of these privacy invasion activities fall into the following categories:

- Customer data collection as well as analysis without the individual subject's awareness, approval and/or sanction;
- Disclosing or sending user data to others without the individual subject's knowledge and authorization, and
- Use of private information in a way not sanctioned by the individual subject.

At the consumer level, many recognize the importance and utility of such collected information in furtherance of better problem solving as it pertain to healthcare delivery or any commerce activities for that matter without compromising of the individual privacy. Hence the need to fully de-identify data becomes of utmost importance given that such vast amount of collected information could benefit the academia, the industry and the government immensely.

Several US legislation and policies, in particular, acknowledge or place significant value unto the applicability and usefulness of de-identification of data. Health Insurance Portability and Accountability Act (HIPAA) being primary. In Europe, General Data Protection Regulation (EU GDPR) 2016/679 is the European Union (EU) and European Economic Area (EEA) data protection and privacy governing policy. EU GDPR also encompasses personal data transfers outside the EU and EEA. In South Africa, every entity(both natural and legal persons) has the same privacy status as the EU GDPR under the Protection of Personal Information Act (or POPI Act), which provides certain requirements for relevant stakeholders(the so-called controllers in other jurisdictions) to lawfully process the identifying information of the subject matter. The POPI Act does not prohibit anyone from accessing your personal information and does not allow you to obtain the consent of data subjects. Whoever decides when personal information is processed and how it is processed is accountable for the conditions.

The POPI Act (POPIA) is significant, as it covers individuals against damage, such as theft and discriminatory practices arising from abuse of such data. The consequences of non-compliance to POPI Act include harm to credibility, fines, and jail and reimbursement for data subjects for losses incurred arising from data processed. The biggest risk is a fine for not safeguarding accounts after reputational damage. The US Health and Human Services(HHS) states on its website that;

> *"The increasing adoption of health information technologies in the United States accelerates their potential to facilitate beneficial studies that combine large, complex data sets from multiple sources." [32]*

Often the data collected from clinical trials  typically integrative clinical studies  can be beneficial to testable theories that can help influence science or find new information that has not initially been found or revealed whenever looking at them, if additional investigators are available, or doctors or academics in the community.

Similar to POPIA; The HIPAA Privacy Rule protects, in any manner or medium, whether electronic, on paper or orally, many "separately recognizable health data" retained or disseminated by a dedicated organization, or its company partner. POPIA defines "personal information" as means information relating to an identifiable, living, natural person, and where it is applicable, an identifiable, existing juristic person [109], including, but not limited to —

- Information relating to the race, gender, sex, pregnancy, marital status,national, ethnic or social origin, color, sexual orientation, age, physical or mental health, well-being, disability, religion, conscience, belief, culture, language and birth of the person [109],
- Information relating to the education or the medical, financial, criminal or employment history of the person [109],
- Any identifying number, symbol, e-mail address, physical address, telephone number, location information, online identifier or other particular assignment to the person [109],
- The biometric information of the person [109],
- The personal opinions, views or preferences of the person [109],
- Correspondence sent by the person that is implicitly or explicitly of a private or confidential nature or further correspondence that would reveal the contents of the original correspondence [109],
- The views or opinions of another individual about the person; and(h)the name of the person if it appears with other personal information relating to the person or if the disclosure of the name itself would reveal information about the person [109].

Such data is called protected health information (PHI) in the HIPAA Privacy Rule as shown in Figure 1.1.

### 1.0.1 HIPAA Applicability on the Providers, the Affiliates, and the PHI

Generally, the HIPAA Privacy Rule protections is applicable to information kept by healthcare covered healthcare providers and partners. According to HIPAA there are three types of healthcare covered entities and are defined as either (a) a health care clearinghouse (b) an electronically transacting provider of healthcare services dealing with certain standard administrative and financial , or (c) a health plan. On the other hand, a business associate can be defined to be an individual or entity (excluding the workforce of a healthcare covered entity) performing certain functions or activities on behalf of a healthcare covered entity that involves the use or disclosure of protected health information, or providing certain services to that entity. Under business associate agreement, a covered entity may use an affiliate to de-identify PHI on its behalf only to the permitable degrees that activity is allowed by the arrangement. [31]

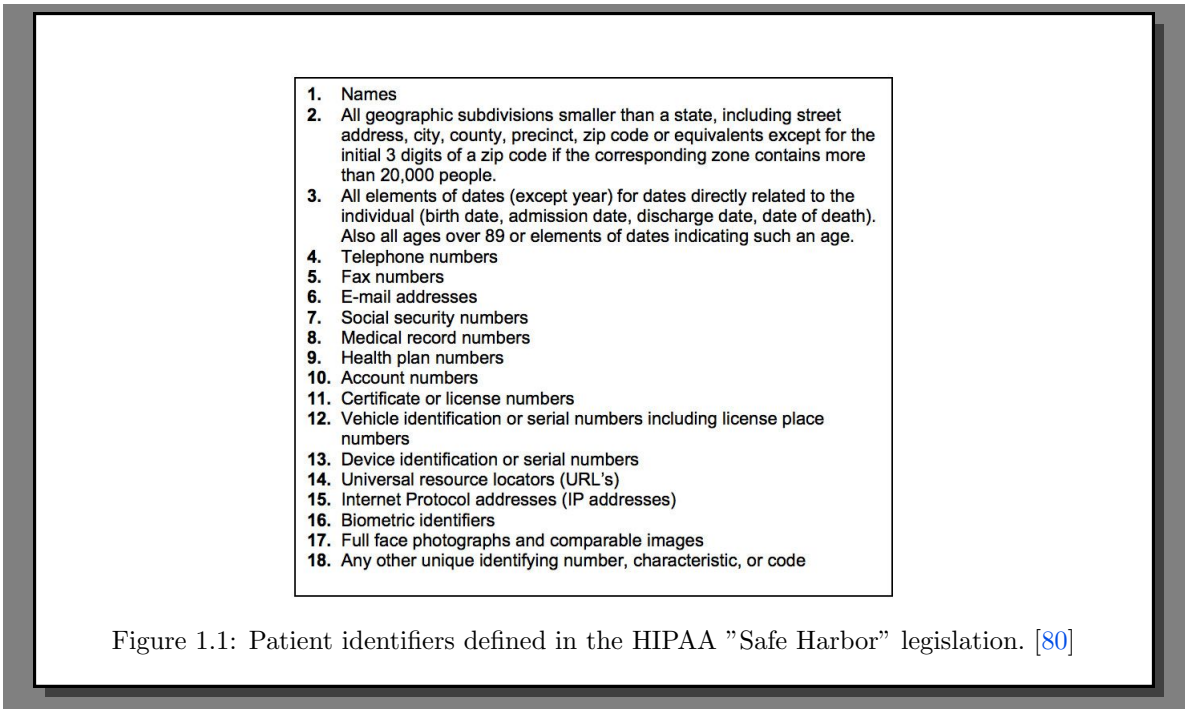



Figure 1.1: Patient identifiers defined in the HIPAA "Safe Harbor" legislation. [80]

The Rule permits only some PHI applications and disclosures or through the information subject's authorization; the Privacy Rule intend to create sufficient safeguard mechanism for personal medical data. Using the de-identification level of quality and specified requirements for implementation§164.514(a)-(b); the Privacy Rule allows applicable provider or its affiliate or partner to generate information that that is not prone to re-identification due to the fact that it acknowledges usefulness of personal 'identity-disconnected' health data that has eliminated a possibility for recipient of such data to link to individual identity §164.502(d). These provisions allow the entity to use and disclose non-identifiable information [31].

#### 1.0.1.1 §164.514: Additional requirements pertaining to PHI usage and disclosure

In §164.514(a), the standard for de-identification of protected health information is outlined, but also the subsection states that *health data not identifying a person and wherefrom there is no good reason to believe that such data could lead to establish a personal identify is not individually identifiable health information* [92].

The de-identification process can be accomplished using two separate pathways within the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule framework—Safe Harbor Method and Expert Determination Approach(shown in Figure 1.2). The Safe Harbor method requires the elimination of all 18 personal identifiers. However, in Expert Determination Approach, some personal identifiers (usually dates and demographics) are retained as long as there is the assurance of an expert that these identifiers could not be used to re-identify the patient. Note that the Privacy Rule does not define the expert's qualifications or expertise or penalize any set of methods that would result in such a determination. The implementation requirements that a covered entity required to comply under the de-identification standard are stipulated in Section 164.514(b) together with Section 164.514(c) of the Privacy Regulation.

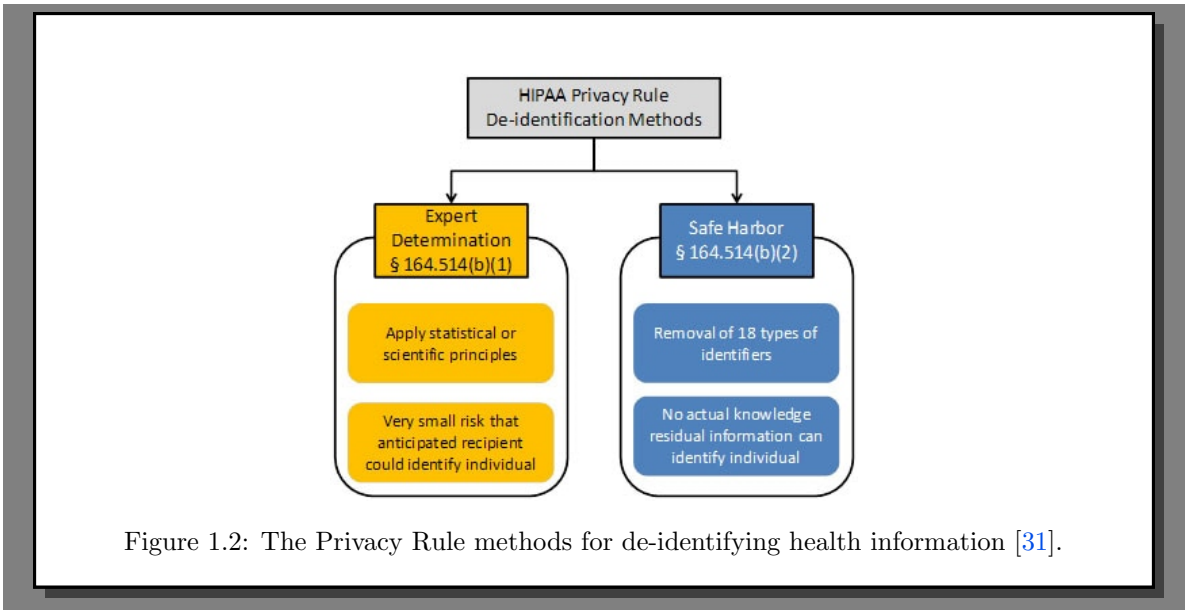



Figure 1.2: The Privacy Rule methods for de-identifying health information [31].

### 1.0.2 Fulfillment guidelines for HIPAA Expert Determination Method

In §164.514(b), for de-identification purposes, the Expert Determination method is defined accordingly as:

1. Individual whose adequate experience and knowledge together with generally accepted statistical and technical fundamental propositions and procedures to render information not personally re-identifiable. (i) Such fundamental propositions and procedures, when applied, determines minimized risk for a prospective recipient identifying the information subject using the data alone or in conjunction with other readily accessible data; and (ii) Document the analytical methods and results justifying such determination [31].

2. The utility of pattern recognition and computational linguistic procedures to recognize and write in the text words and other alphanumeric tokens denoting PII (e.g. names, addresses, telephone and social security numbers)comes handy in Computational de-identification. Grammatical tagging or word-category disambiguation is the process used in corpus linguistics by marking a word in a corpus as corresponding to a particular part of the speech tag (POS tagging or PoS tagging or POST) based on both its definition and context i.e., its relationship with adjacent and related words in a phrase, phrase or paragraph.

This process is similar to what school-age children are taught the identification of words as nouns, verbs, adjectives, adverbs, etc. Once the process of recognizing words and other alphanumeric tokens denoting PII in the clinical text has happened and redact has occurred we can then say that the patient privacy has been protected and clinical knowledge is preserved.

### 1.0.3 Challenges and limitations in achieving high accuracy

There are many challenges in achieving high accuracy in the de-identification of clinical notes. Many researches have documented and indicated challenges of attaining high recognition rates. The problem of low recognition rates, and thus low accuracies could be the result of many unforeseen circumstances. Many studies select hyperparameters from the development set [64, 70], usage of a smaller dataset (13862 vs 14987 sentences) was done by Reimers and Gurevych [101], others combined development set into training set [11, 97]. Huang et al. chose to split POS dataset differently [53], while the usage of different development sets for chunking was performed by Liu et al. [66] and Hashimoto et al. [49]. The challenges challenges in achieving high accuracy documented in all these studies can be classified into the following groups:

a) Preprocessing.

b) Features.

c) Hyperparameters.

d) Evaluation.

e) Hardware environment.

When it comes to data preprocessing the goal is to normalize digit characters [11, 64, 113]. The survey of various methods used by other researchers indicate that there have been myriad of approaches in the data preprocessing whereby, for less common words, Gurevych and Reimers [101] uses fine-grained representation approach while Ma and Hovy [70] skipped preprocessing at all. When it comes to Features we can see that Strubell et al. [113] and Chiu and Nichols [11] utilized word orthography properties approach while a further integration of context features approach is done by Huang et al. [53]. The usage of neural features to represent external gazetteer information approach is done by Collobert et al. [15] and Huang et al. [53]. The utilization of end-to-end structure without handcrafted features was employed by Lample et al. [64] and Ma and Hovy [70]. Another noted challenge is that of selection of Hyperparameters including learning rate, dropout rate [110], number of layers, hidden size etc. It was demonstrated by Chiu and Nichols [11] that hyperparameters can strongly affect the model performance and concluded it is sensitive to the selection of hyperparameters given system performance. Existing models, however, use various parameter settings that affect fair comparison. When it comes to Evaluation; some opted to go for utilizing various random seeds, whereas others have documented findings using standard deviations and mean [11, 66, 97] approach, while trial's best result from various tests was an approach reported by others [70] without showing a connection on how all these various approaches or methods can be compared directly to one against the other. Lastly, it was demonstrated by Liu et al. [66] that entity identification, entity chunking and entity extraction (also referred as entity recognition or NER) is more accurate when trained in GPU-based machine than using CPU-based hence concluding that the Hardware environment can also affect the system accuracy.

Given the outlined challenges, it is important to examine the landscape and see—what commercial-off-the-shelf(COTS) available, what features they offer, and any other relevant information that we can use to compare to the Recurrent Neural Network Method that I am proposing.

## 1.1 Commercial Methods

The process of de-identification has been largely a manual and labor intensive task because of the fact that both the data's sensitive nature and the limited availability of software to automate the task. This has led to a relatively small number of open health data sets available for public use. The readily de-identification tools accessible, for structured data, that exist are only accessible internal to organizations and therefore are not generally available, or have been developed for personal use (by researchers) [54] are:

- The Privacy Analytics Inc.'s PARAT tool integrates risk mitigation for three identity risk types [54].
- The $\mu$-Argus ($\mu$-Anti-Re-identification General Utility System (ARGUS)) is a tool produced by the Dutch government statistical agency [54].
- The Cornell Anonymization Toolkit (CAT), an open-source utility based on k-anonymity algorithm [54].
- Anonymization Toolbox, from The University of Texas at Dallas is open-source documentation-capable Java-based k-anonymous with attribute privacy algorithms [54].

- sdMicro, R-based utility that offers some fundamental de-identifying features [54].

However, when it comes to unstructured data, like clinical notes, there are not many options. Government agencies have also been involved in the efforts to help different players in the industry develop tools that can address broader patient privacy concerns and in particular the de-identification problems as demonstrated by budget allocation depicted on Figure 1.3. The NLM-Scrubber, for instance, is a software tool designed and developed at the National Library of Medicine [56]. The NLM-Scrubber can de-identify various clinical document types with great precision and was put together by the U.S. National Library of Medicine (NLM) after noticing capability deficiencies presented by the existing de-identification tools [56]. The software uses a combination of both recognition of deterministic and probabilistic patterns together with the computational linguistic methods that use large dictionaries of locations, individuals and organizational names [56]. Reports are accepted in plain-text or HL7 standard by the application. The application software leverages its capability by fully utilizing HL7-encoded patient data to locate such data in the text whenever the input reports are submitted as HL7-formatted messages.

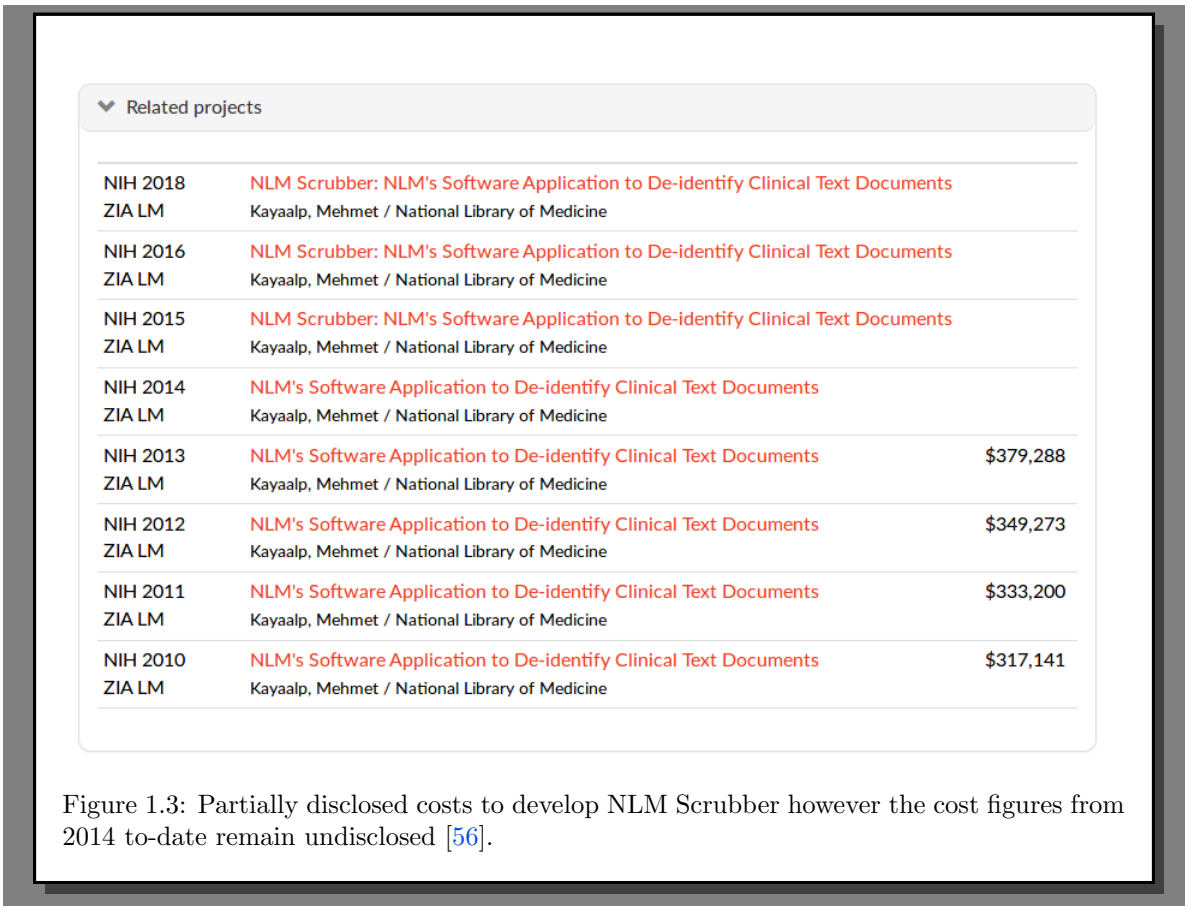



Figure 1.3: Partially disclosed costs to develop NLM Scrubber however the cost figures from 2014 to-date remain undisclosed [56].

## 1.2 Research Objectives

This research work seek to solve the de-identification of textual medical notes or clinical notes problem whereby all personal identifiable information(PII) relating to personal health information(PHI) will be automatically tagged by the recurrent neural network model. Current advancement in the areas of sequential tagging together with deep learning methods provide motivation to pursue research in this area.

Background and Significance will be covered in Chapter 2 while Chapter 3 will cover the review of related work. Machine Learning methods together with Feature Representation will be on Chapter 4 and Chapter 5 respectively. Research methods will be discussed in Chapter 6, whereas Chapter 7 will cover the research findings. Conclusions and future work will be on Chapter 8.

# Chapter 2

# Background and Significance

## 2.1 Introduction

In order for any medical information to be released and used for research, the HIPAA privacy rule mandates that either;

- PHIs ought to be completely and thoroughly purged using de-identification procedures,
- a patient's agreement must be sought and granted, or
- a consent waiver should be granted by the institutional review board.

The Electronic Health Record (EHR)'s narrative clinical text is increasingly recognized and acknowledged as an integral part of computerized support for decision, quality enhancement, and patient safety [81]. The relevance of NLP as an empowering tool for accessing extensive valuable information from EHR notes can not be overstated, since NLP can extract information from clinical free-text and present clinical knowledge in a structured format. The safety of patients and clinical research could certainly benefit from text formatted medical information, which cannot be found in either EHR database-formatted entries or data from administration of the patient [89].

In the mid 1980s, Parts-of-Speech(PoS) disambiguation researchers began using hidden Markov models (HMMs) to tag the English corpus of Lancaster-Oslo/Bergen (LOB) [37, 55, 74]. HMMs involve enumerating cases and tabulating probability based on certain sequences. Such work laid a very important basis and great foundation in computation linguistic.

The process of flagging up a word in a textual information records as corresponding to a respective part-of-speech is referred to as word-category disambiguation or grammatical tagging. But before we dive deeper into how this study was conducted, it is important that we cover the background and significance. In this section, I will cover de-identification and risks, Mathematical aspects of the de-identification process, and the important concept of backpropagation.

## 2.2 De-identification and re-identification Risks

Forty-eight U.S. states collect and diffuse data for hospital, ambulatory and emergency visits, which they subsequently share with public as well as private organizations [68]. State-wide collection of hospital data is helpful both to researchers, regulators and companies. Given that South Africa has POPIA, then many industries leaders in South Africa could leverage the availability of such medical data to create myriad of solutions that could help develop enhanced healthcare delivery. Such a move, if taken by the industry leaders, could help expand the medical research collaboration with other technologically advanced nations, widen the intelligent medical devices manufacturing base in South Africa, and most

importantly, help to grow South Africa's economy by potentially attracting additional direct foreign investment. Majority of the U.S. states follow a version of the Health Information Portability and Accountability Act (HIPAA) Privacy Rule's Safe Harbor de-identification method. The guidelines for Safe Harbor de-identification are commonly recognized as de facto "level of quality" accepted for de-identifying health data of various forms despite requirement to follow HIPAA.

In Figure 2.1, for example, being sole 1953-born male in the population [8]. Accordingly, the information disclosed in the voting list can readily be determined as related to the record of patients in the medical information published.

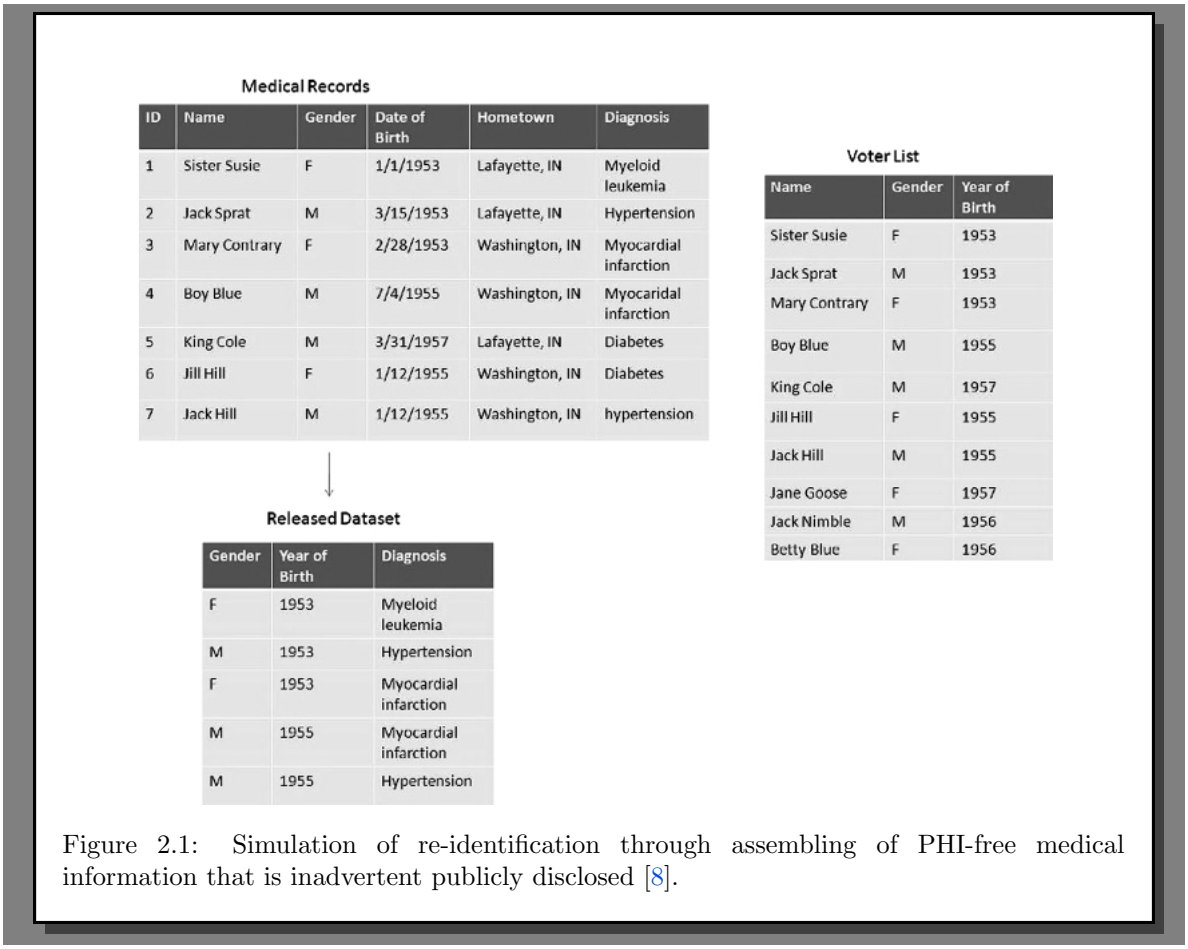



Figure 2.1: Simulation of re-identification through assembling of PHI-free medical information that is inadvertent publicly disclosed [8].

An extensively colossal assault on data target, which is prone to re-identification, could easier be achieved when an attacker has the accessibility of a well identified dataset that encompass wider population in question. Public records, in this regard, tend to provide for an easily available resource that many times includes intricately vast population specifics and characteristics. For instance, a theft of a laptop that has a storage volume without encryption with lists stored on such a hard unencrypted disk —exploiting unencrypted storage vulnerability through such unlawful act and means (theft) presents an attacker with identified data with myriad characteristics associated with de-identified data. (e.g., see Tennessee [75]), or hacking a state-owned website (e.g., see Illinois [39]). Legal avenues equally allow potential assailants to access and obtain certain public records by exploiting the legal gaps, e.g. voter registrants records without crime allegedly committed. The re-identification of state records could create massive disruptions given the unauthorized identification could harm the organizations that have not explicitly and legally allowed State records to be re-identified. By opting to encourage researchers to report vulnerabilities, States can utilize such a mechanism to reduce disruption, however prohibition of experimental re-identification may create adverse effect.

### 2.2.1 De-identification: What does the process entail?

De-identification is a blend or infusion of practices, algorithms, and procedures that could be deployed to various data types with differing levels of efficiencies in order to remove personal information from data that organization is collecting, using, archiving, and sharing with other organizations. The Figure 2.2 depicts the de-identification in a nutshell. Certain practitioners that share de-identified data might want to serve a Data Use Agreement (DUA) with subsequent users in order to counteract the re-identification risk because a DUA, in such instance, might not allow a de-identified data recipient from sharing such data without permission [1], or use of received data to try to reconstitute the identity by injecting the PII of individual person(s) or connect them to external data.

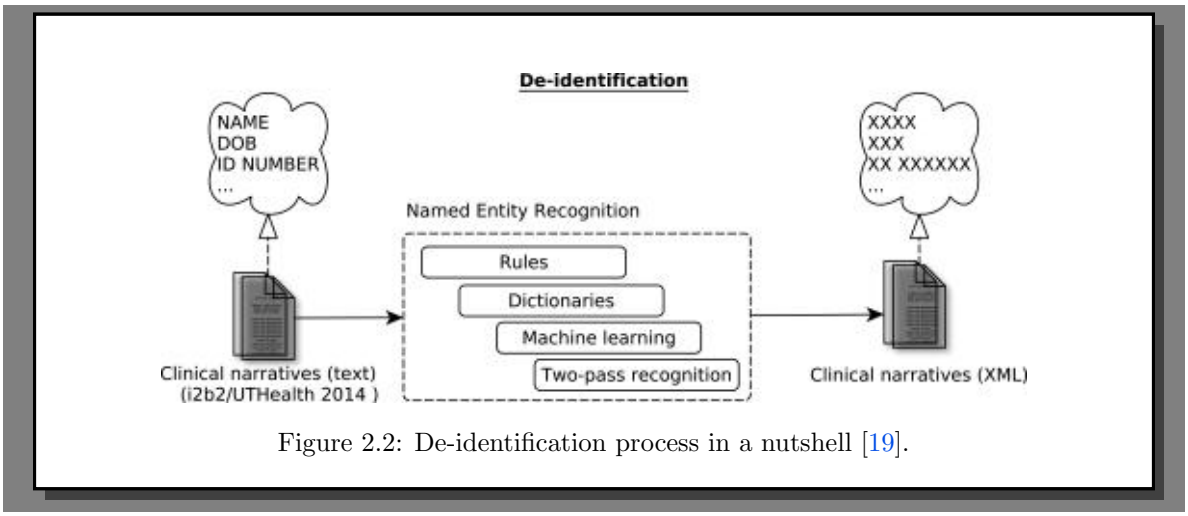



Figure 2.2: De-identification process in a nutshell [19].

As the privacy continue to become a major concern for customers and enterprises alike in the corporate marketing strategies, equally will the Privacy preserving methods increase in its importance. Hence, the challenging questions and issues not only about privacy and information use, but also protection. One approach of privacy is based on who and for what purpose is one allowed to access private information. For instance, prohibiting use of personal information given to hospitals by patients for sole record purposes, and not for advertising.

### 2.2.2 Private information use Privacy-Preserving Models

In structured data; two separate usage methodologies that address handling personal information while protecting individuals' privacy are classified as:

- Privacy Preserving Data Mining (PPDM)
- Privacy Preserving Data Publishing (PPDP)

#### 2.2.2.1 Data Mining under Privacy Preserving models

Under Privacy Preserving Data Mining (PPDM) methodology data is not publicly published, but rather viewed as an input into further processes of statistics or computational algorithmic loops aiming to unearth the underlying trends or patterns instead[2]. Aggregated/ summarized statistical tabulation, classifiers implementing algorithms for computational learning together with additional results types could be one way whereby calculation results are released. Statistical Disclosure Limitation "is the discipline concerned with re-identification and disclosure prevention through modifying statistical data accessible by the third parties." [128]

[1] The HIPAA Privacy Rule requires data use arrangements for the sharing of limitable, less standardized data sets. DUA may not be necessary during de-identified datasets dissemination using the Privacy Rule

[2] *Machine Learning* is a combination of computational algorithmic methods and practices which enable computing platforms, without being explicitly programmed, to classify data and recognize patterns

#### 2.2.2.2 Data Publishing under Privacy Preserving models

Data is processed under Privacy Preserving Data Publishing (PPDP) model and presented forward as synthetic, new, or de-identified data for end-user distribution. Under PPDP, the individual identity protection makes learning whether the processed data is related to a particular person in a dataset difficult and/or impossible and, at the same time, preserving some usefulness of the dataset for alternative uses —given that the utility of the processed data is the goals of de-identification. De-identification is principal among many of PPDP's tools. To produce a similar range of data, **Synthetic data generation** uses certain PPDM methodologies making any or all derived information aspects impossible to be mapped to respective subjects. There is, therefore, a fusion between PPDM and PPDP when it comes to making the synthetic data.

The core of all NER systems constitute a statistical models set representing the various named entities to be recognized. Since medical notes (text), just like any textual material in natural language processing, has temporal composition and can be encoded as a vectoral arrangement spanning the text frequency range, the HMM offers an entirely logical paradigm to build these prototypes. Now let us dive deeper and examine the mathematical aspect of HMMs.

## 2.3 Mathematical Aspect of the Process

A Markov model is a statistical model theory whereby the modeling mechanism is anticipated to be a non-observed (i.e. hidden). Markov processes were developed and represented by L. E. Baum et al. at the Institute for Defense Analyses in Princeton as the simplest dynamic Bayesian network [3, 4]. The first fundamental result is due to Baum and Petrie for finite state Markov chains with finite-range read-outs [4]. The Baum and Petrie analytical assessment is presented on the hypothesis covered under the Shannon-Breiman-McMillan theorem. The earliest description to ever appear on the subject of forward-backward procedure are attributed to Ruslan L. Stratonovich; actually the problem of optimal nonlinear filtering is closely related to Hidden Markov Model (HMM).

The HMM is predicated on the Markov chain extension whereby this model Markov chain is capable of informing us on the random variables sequences probabilities and states each with ability to accept values from some set. These sets could represent anything be it words, labels, symbols , or the weather. A Markov chain uses the current state if the future in the pattern is to be forecast. The states preceding the present iteration have no potential future effect except through the present state. To predict tomorrow's weather one has to examine today's weather however was not allowed to look at yesterday's weather in isolation of today's. The state's traversing pattern is referred as the *hidden state*; and not the parameters of the model. In this case, the model is referred to as *Hidden Markov Model (HMM)* irrespective whether the parameters are known.

We can calculate a probability for a sequence of observable events using a Markov chain. The events that we are concerned about are hidden in many cases: we do not directly observe or monitor them. For instance, we normally do not observe hidden part-of-speech labels when reading a textual document. We only see phrases, rather, and have to deduce the respective labels of the pattern of words. In this instance, the tags represent the hidden state because they are not directly observed.

### 2.3.1 Hidden Markov Models (HMMs)

In a Hidden Markov Model, one can observe a sequential pattern $\mathbf{o} = o_1, \ldots, o_T$ that is associated with a state sequence that one cannot observe $\mathbf{s} = s_1, \ldots, s_T$. The model describes the joint state and

observation sequence:

$$p(s_1, s_2, \ldots, s_T, o_1, o_2, \ldots, o_T) = p(s_1)p(o_1|s_1)\prod_{i=2}^{T} p(s_i|s_{i-1})p(o_i|s_i)$$

and one gets the probability of the observation sequence by marginalizing:

$$p(o_1, \ldots, o_T) = \sum_{\mathbf{s}} p(o_1, \ldots, o_T, s_1, \ldots, s_T)$$

where; The **observations**, $\mathbf{o} = o_1, \ldots, o_T$ from a finite set V, referred as the observation space.
The **hidden states**, $\mathbf{s} = s_1, \ldots, s_T$ from a finite set Q, referred as the hidden state space.
$p(s_1, s_2, \ldots, s_T, o_1, o_2, \ldots, o_T)$, the joint probability distribution of a sequence of observations.
The **initial state probability**, $p(s_1)$.
The **transition probabilities**, $p(s_i|s_{i-1})$.
The **emission probabilities**, $p(o_i|s_i)$.

There are two key assumptions in an HMM. The first key assumption is that the state sequence is Markov; $p(s_i|s_1, \ldots, s_{i-1}) = p(s_i|s_{i-1})$ , and the second assumption being that the observations are conditionally independent of future and past states and observations given the current state: $p(o_i|s_1, \ldots, s_T, o_1, \ldots, o_{i-1}, o_{i+1}, \ldots, o_T) = p(o_i|s_i)$. It is important to note the presence of two types of Hidden Markov Models, labeled states ($p(o_i|s_i)$) and labeled transitions ($p(o_i|s_{i-1}, s_i)$) and the fact that they can be transformed back and forth.

#### 2.3.1.1 HMM Examples in Language Processing

There are numerous instances to list. For instance however, part-of-speech tagging, identification of named entities in text, punctuation prediction, and recognition of speech patterns are a few examples to be included on that list. Additional information and details of such examples that fall under this category are covered next.

a) **Part-of-speech (POS) Tagging:**

In part-of-speech (POS) tagging; states are POS tags, $p(s_i|s_{i-1})$ that describe POS sequence tendencies; whereas, the observations are words, $p(o_i|s_i)$ that describe word-tag relations. For example;

$$o_1, o_2, o_3, o_4 = I \textit{ saw the boy}$$
$$s_1, s_2, s_3, s_4 = \textit{pronoun verb det noun}$$

where; the **observation**, *I saw the boy*, is denoted as $o_1, o_2, o_3, o_4$
the **hidden state**, *pronoun verb det noun*, denoted as $s_1, s_2, s_3, s_4$.

It could be helpful to have more sophisticated depictions of the words as observations to better model unknown words, i.e. $o_i$ not in a known vocabulary, since the observation space needs to be finite in order to specify $p(o_i|s_i)$. Consider possible observation sequences:

> I saw the zlrxl
> 'Twas brillig and the slithy toves, Did gyre and gimble in the wabe: All mimsy were the borogoves, And the mome raths outgrabe.

One would probably say that "*zlrxl*" is a noun, "*slithy*" is an adjective, and "*toves*" is a plural noun, based on endings of the words and/or neighboring POS types. Hence, one might want to expand $o_i$ to a vector with the word, ending, capitalization, etc as elements of the observation vector. We

equally have to know that there are some limitations of an HMM for POS tagging, so extensions or other models are more often used. For example, the POS tag might be better predicted by using the previous word and not just the previous tag, e.g. some verbs do not take a direct object.

b) **Name Recognition**

In Name Entity Recognition (NER); states include {begin_person (BP), cont_person (CP), begin_location (BL), cont_location (CL), not_a_name ($\phi$)} , $p(s_i|s_{i-1})$ describes name sequence tendencies, while observations are words, $p(o_i|s_i)$ describes word-name relations, for example;

$$o_1, o_2, o_3, o_4, o_5, o_6, o_7 = We\ saw\ Senator\ John\ McCain\ in\ Arizona$$
$$s_1, s_2, s_3, s_4, s_5, s_6, s_7 = \phi\ \phi\ \phi\ BP\ CP\ \phi\ BL$$

where; The **observation**, *'We saw Senator John McCain in Arizona'*, is denoted as $o_1, o_2, o_3, o_4, o_5, o_6, o_7$, while;
The **hidden state**, $\phi\ \phi\ \phi\ BP\ CP\ \phi\ BL$, is denoted as $s_1, s_2, s_3, s_4, s_5, s_6, s_7$

Again, it might be useful to have features for characterizing new words, and it might be useful to condition the predicted state and/or observation on both the previous word and state: $p(s_i|s_{i-1}, o_{i-1})$; $p(o_i|s_i, o_{i-1})$

c) **Other sequence labeling problems includes:**

(i) Sentence segmentation: sequence of boundary vs. no_boundary decisions after every word.

(ii) Topic segmentation: sequence of boundary vs. no_boundary decisions after every sentence.

(iii) Speech act tagging on the sequence of utterances in a conversation.

#### 2.3.1.2 Hidden Variables

Hidden variables are useful in many ways. A good example is their utility in modeling different sources of variability, e.g. temporal variability in an HMM where multi-modal behavior in a mixture model contain some random event that one can not observe that determines the distribution. Moreover, the hidden variables may be what one want to detect (such as in tagging). But it is important to note that hidden variables are useful in learning distributions where some values are missing, e.g. in missing labels for semi-supervised learning. When working with hidden variables, it is important to know that computing probability of observations $p(o)$ require marginalizing (summing out) hidden states and the iterative nature of parameter estimation —a good example is the EM algorithm for maximum likelihood estimation.

#### 2.3.1.3 The Expectation-Maximization (EM) Algorithm

Given training data $\mathcal{X} = \{x_1, \ldots, x_T\}$, a set of unobserved latent data or hidden state $S = \{s_1, \ldots, s_T\}$, a vector of unknown parameters $\theta$ and model $p(x, s|\theta)$; the maximum likelihood parameter estimate requires $\text{argmax}_\theta \log p(\mathcal{X}|\theta) = \text{argmax}_\theta \sum_{i=1}^{T} \log \sum_s p(x_i, s|\theta)$ where the sum over the hidden variable makes the solution untenable. The basic idea here is that to directly maximize log likelihood $\log p(\mathcal{X}|\theta)$ is too hard, but one can maximize $E\big[\log p(\mathcal{X}, \mathcal{S}|\mathcal{X}, \theta)\big]$ and hence indirectly maximize log likelihood. So far no closed form solution found though, therefore one need to iterate by going through the E-step and M-Step as follows:

**The E-step:** In this step we use the current value of the parameters of $\theta$ to find the posterior distribution of the latent variables given by $p(\mathcal{X}, \mathcal{S}|\mathcal{X}, \theta^{(l)})$. This corresponds to the $\gamma_t^{(l)}(j)$ . Then utilize this to find the expectation of the complete data log-likelihood, with respect to the current conditional

distribution of $S$ given the training data $\mathcal{X}$ and the current estimates of the parameters $\theta^{(l)}$, in this case denoted by $Q(\theta|\theta^{(l)})$, by computing:

$$Q(\theta|\theta^{(l)}) = E\big[\log p(\mathcal{X}, \mathcal{S})|\mathcal{X}, \theta^{(l)})\big].$$

where; $Q(\theta|\theta^{(l)})$ denotes the expectation, a vector of unknown parameters is $\theta$, the current estimate of the parameters is $\theta^{(l)}$, the next estimate of the parameters is $\theta^{(l+1)}$ is maximized assuming that the (missing) data from $\theta^{(l)}$ are actually known.

**The M-step:** In this step we will find the new parameter $\theta^{(l+1)}$ through computing the maximized Q:

$$\theta^{(l+1)} = \underset{\theta}{\operatorname{argmax}}\, Q(\theta|\theta^{(l)}).$$

**Note:** Since latent variables can not be observed, we take the whole log-like expectation into account with regard to latent variables

A quantitative statistic is sufficient (or just *sufficient statistics*) in relation to a mathematical model and its related unknown variable when 'there is no extra data concerning the valuation of the parameter which can be computed from same batch' [30]. For problems involving $p(x, s)$ in the exponential family (also known as Koopman-Darmois family); the EM algorithm reduces to finding expected sufficient statistics $E[t^{(l)}]$ of unobserved process in **E-step** where the sufficient statistic is denoted as $t^{(l)}$. Also, we use the $E[t^{(l)}]$ of unobserved process in place of $t$ in the maximum likelihood (ML) update formulas for $\theta^{(l+1)}$ in **M-step**. For many discrete hidden variable problems, the E-step involves estimating the probabilities for different possible state values:

$$\gamma_t^{(l)}(j) = p(s_t = j|x, \theta^{(l)})$$

where $\gamma_t^{(l)}(j)$ is the probability of being in state $j$ at time $t$ given the observed sequence $l$ and the parameters $\theta$

**The Expectation-Maximization for Mixtures:**
Mixture models are mathematical presentations in which a weighted total of conditional probability determines the probability distribution of a statistic. In a mixture distribution; the distribution is a mixture of m component distribution $p_i, \ldots, p_m$; where, the index of the underlying mode is the hidden variable and given by:

$$p(x) = \sum_{i=1}^{m} \lambda_i p_i(x) = \sum_{i=1}^{m} p(z = i)p(x|z = i)$$

where; $z_i \in \{1, \ldots, m\}$ is the latent variable representing the mixture component for $x_i$, $p(x|z_i)$ is the mixture component. $\lambda_i$ is the mixture proportion representing the probability that $x_i$ belongs to the m-th mixture component, the $\lambda_i > 0$ being the mixture proportion or weight, and $\sum_{i=1}^{m} \lambda_i = 1$. We need mixture models for the simple fact that not every variable has equal contribution to the model; hence, the weight should reflect the appropriate contribution.

First; initialize by providing $\{\lambda_i^{(0)}, p_j^{(0)}(x)\}$ and then iterate by computing $\gamma_t^{(l)}(j) = p(z_t = j|x_t, \theta^{(l)}) = p^{(l)}(z_t = j, x_t)/\big[\sum_k p^{(l)}(z_t = k, x_t)\big]$ in **E-step**. By finding a relative frequency estimate using weighted counts —this is achieved by updating component model parameters using $\gamma_t$-weighted observation statistics and update mixture weights in **M-step** by using:

$$\lambda_j^{(l+1)} = \frac{1}{T} \sum_t \gamma_t^{(l)}(j)$$

where; the $\lambda_i > 0$ being the mixture proportion or weight, $\gamma_t^{(l)}(j)$ is the posterior probability of being in

state $j$ at time $t$ given the observed sequence $T$, the parameters $\theta$, and sum of all time $t$ instances t.

The mixture estimation algorithm works for all types of mixtures, with the difference being in the details of how the distributions $p_i(x)$ are estimated. Important examples include the following:

- Mixtures of language models (for topic and genre modeling): The MLE value of n-gram [3] parameters is acquired by receiving counts from a corpus and standardizing the counts 0 to 1. For the general case of maximum likelihood estimation (MLE) N-gram parameter estimation is computed from:

$$p(\kappa_n|\kappa_{n-N+1}^{n-1}) = \frac{c(\kappa_{n-N+1}^{n-1}\kappa_n)}{c(\kappa_{n-N+1}^{n-1})}$$

  whereby; a sequence of N words is either denoted as $\kappa_1 ... \kappa_n$ or $\kappa_n^1$.

  The probabilities of the linguistic model are logarithmic. By definition likelihoods are below or equal to 1, the further probabilities we multiply around each other, thus less product. Adequate N-grams will also cause mathematically undercutting each other given that product of fractional numbers will be smaller fraction. We receive figures which are not as low utilizing log likelihoods rather than raw probabilities. Therefore, we can denote weighted n-gram counts, as:

$$c(\kappa_a) = \sum_{t:\kappa_t=\kappa_a} \gamma_t(j)$$

  where $\kappa_i$ is the word i, $\gamma_t^{(l)}(j)$ is the posterior probability

- Gaussian mixtures model or method is a stochastic approach where all information points are produced by a combination of a finite amount of unknown Gaussian distributions. Gaussians are simple to predict and understand although they are unimodal, however multimodal sets of data cannot easily be depicted. In a low probability field a single Gaussian fits into a multimodal data.

$$n_j = \sum_t \gamma_t(j); \quad \mu_j = \frac{1}{n_j}\sum_t \gamma_t(j)x_t; \quad \sigma_j^2 = \frac{1}{n_j}\sum_t \gamma_t(j)(x_t - \mu_j)^2$$

where $\mu$ is the mean and $\sigma$ is the standard deviation, $n_j$ is Gaussian curves at $j$. The $(l)$ superscript is dropped to simplify the notation. It should be noted that one can also keep the mixture components fixed and just estimate the mixture weights if the components come from other training data.

**The Expectation-Maximization for HMMs:** From the BaumWelch algorithm, let us consider discrete (categorical) HMMs of length $T$ within the problem of learning the parameterization of $\theta$ from a dataset of $D$ observations; we can start by defining the HMM parameters as follows: The state probability matrix $\pi_j = p(s_1 = j)$; the time-independent stochastic state transition matrix $\alpha_{jk} = p(s_i = k|s_{i-1} = j)$; the emission matrix $\beta_j(o) = p(o|s_t = j)$ we can then compute the probability of being in state j at time t given the observed sequence $T$ and the parameters $\theta$, and will be defined as;

$$\gamma_t^{(l)}(j) = p(s_t = j|o_1, \ldots, o_T, \theta^{(l)})$$

whereas, the expectation of having $j$ and $k$ is obtainable by computing likelihood using parameter $j$ previous state $s_{t-1}$ given the current state $s_t$ respectively given the observed sequence T and parameters $\theta$, and is then presented as;

$$\xi_t^{(l)}(j,k) = p(s_{t-1} = j, s_t = k|o_1, \ldots, o_T, \theta^{(l)})$$

[3] The n-gram is an adjacent series of n parts for a specified batch of text and speech in computational linguistics and probability.

in the **E-step** before one updates component model parameters using $\gamma_t$-weighted observation statistics and update transitions using $\xi_t$ terms in the **M-step**. Using the same weighted counts idea as for mixtures:

$$a_{jk}^{(l+1)} = \frac{\sum_t \xi_t^{(l)}(j,k)}{\sum_k \sum_t \xi_t^{(l)}(j,k)}$$

The update for $\pi_j$ is similar, and the update for the observation distribution depends on the form of $b_j(o)$ but also uses weighted observations(continuous $o$) or weighted counts (discrete $o$). The model-related details are primarily in the E-Step, which is more complicated than the mixture model because of the sequential dependencies. So now, let's go back to the details of the HMM.

#### 2.3.1.4 Details for HMMs

There are many questions that people ask about HMMs but will focus on the most important questions —which happen to be; How does one compute $p(o)$? What is the most likely state sequence $\text{argmax}_s\, p(s|o)$? What is the most likely state at a particular time $t$, $\text{argmax}_j\, p(s_t = j|o)$? How does one estimate the parameters of an HMM? To make it possible to address such important questions raised, we need to reference into the notion that hidden Markov models ought to construe three core concerns —a concept that was introduced through the teachings of Rabiner [98]. Equally I should point out that from Rabiner's work, Jack Ferguson of IDA (Institute for Defense Analysis) provided additional in-depth analysis [27]. And it is from this analysis where we obtain the three key algorithms needed to answer the questions of **Likelihood**, **Decoding**, and **Learning**.

The first problem is known as the (**Likelihood**): Given a Hidden Markov Models (HMM) whereby state probability matrix $\pi_j = p(s_1 = j)$; the time-independent stochastic state transition matrix $\alpha_{jk} = p(s_i = k|s_{i-1} = j)$; the emission matrix $\beta_j(o) = p(o|s_t = j)$ then $\theta = (\alpha, \beta)$ together with an occurrence pattern o, determine the likelihood $p(o|\theta)$. We therefore obtain the **Forward algorithm** which is presented as:

$$\begin{aligned}\alpha_t(k) &= p(o_1, \ldots, o_t, s_t = j) \\ &= \sum_j a_{t-1}(j)\alpha_{jk}\pi_k(o_t)\end{aligned}$$

where; $a_{t-1}(j)$ : the **probability of preceding forward path** from the time step before current state.
$\alpha_{jk}$ : the **transitional probability** from state before current state $s_j$ to present state $s_k$.
$\pi_k(o_t)$ : the **state occurrence likelihood** of the occurrence symbol $o_t$ given the present state k

The second problem is known as the (**Decoding**): For any model, such as a Hidden Markov Models (HMM), containing hidden parameters, the decoding task is the work of identifying which sequence of variables is the underlying source of some inferential pattern. Given a sequence of conjecture $o = o_1, o_2, \ldots, o_t$ and an HMM $\theta = (\alpha, \beta)$, discover the best underlying inferential state pattern $s = s_1, s_2, \ldots, s_t$. Here we then have the **Viterbi algorithm** as:

$$\begin{aligned}\delta_t(k) &= \max_{s_1, s_2, \ldots, s_{t-1}} p(o_1, o_2, \ldots, o_t, s_1, s_2, \ldots, s_{t-1}, s_t = k) \\ &= \max_j \delta_{t-1}(j) a_{jk}\pi_k(o_t)\end{aligned}$$

where; $\delta_{t-1}(j)$ :the **preceding Viterbi path probability** from the time step before current state.
$\alpha_{jk}$ :the **transitional probability** from state before current state $s_j$ to present state $s_k$.
$\pi_k(o_t)$ : the **state occurrence likelihood** of the occurrence symbol $o_t$ given the current state $s_k$

The third problem is known as the (**Learning**): Given some inferential pattern o and a pair of the hidden Markov model states, learn the hidden Markov model variable $\theta = (\alpha, \beta)$. Finally, we have the **Backward algorithm** as follows:

$$\begin{aligned}\beta_t(j) &= p(o_{t+1}, \ldots, o_T | s_t = j) \\ &= \sum_k \beta_{t+1}(k) a_{jk} \pi_k(o_{t+1})\end{aligned}$$

Then we can then use the algorithms from the Likelihood, Decoding, and Learning processes above to answer the questions [29] as follows:

1. Calculate $p(o)$ using the forward algorithm as follows:

$$p(o) = p(o_1, \ldots, o_t) = \sum_j p(o_1, \ldots, o_t, s_t = j) = \sum_j \alpha_t(j) = \alpha_T$$

2. Use the Viterbi algorithm, as shown in (b) above, to monitor the max $j$ at each time step, then trace back from the final best state.

3. Use the forward and backward algorithms to find

$$\begin{aligned}\gamma_t(j) &= p(s_t = j | o_1, o_2, \ldots, o_t) = p(s_t = j, o_1, o_2, \ldots, o_t)/p(\mathbf{o}) \\ &= \alpha_t(j)\beta_t(j)/\alpha_T\end{aligned}$$

   and use $\gamma_t(j) = p(s_t = j|o)$ in finding the argmax.

4. And to estimate the parameters of an HMM; utilize the forward and backward algorithms while in E-step to find $\gamma_t(j)$ as above and

$$\begin{aligned}\xi_t(j,k) &= p(s_{t-1} = j, s_t = k | o_1, \ldots, o_T) \\ &= \alpha_t(j) a_{jk} b_k(o_{t+1}) \beta_{t+1}(k)/\alpha_T\end{aligned}$$

We know that hidden-event language models represent a hidden event. Let's now look into Hidden-Event Language Models as defined by HMM and how those lay a foundation into the different methods that are being used today.

### 2.3.2 Hidden-Event Language Models

Language models are methods that allocate likelihoods to word combinations. N-Gram is the most basic method that allocates such likelihoods to the contiguous series of n items from a specified text or speech sample e.g. phonemes, syllables, letters, words or base pairs as per the selected implementation method. Hidden-event language models represent a hidden event $e_i$ (such as a sentence or topic boundary) after every word $\kappa_i$. It is a bit like an HMM because the hidden event, but the observations are based on n-gram language models. The bigram (2-gram) case would be:

$$\begin{aligned}p(\kappa, \mathbf{e}) &= p(\kappa_1, e_1, \kappa_2, e_2, \ldots, \kappa_T, e_T) \\ &= p(e_1|\kappa_1)p(\kappa_1) \prod_t p(e_t|\kappa_t, \kappa_{t-1}, e_{t-1}) p(\kappa_t|\kappa_{t-1}, e_{t-1})\end{aligned}$$

The $p(e_i|\cdot)$ terms are the hidden state sequence model, but different from and HMM in that the state transitions depend on the words, and the $p(\kappa_t|\cdot)$ terms are the observation model which is an event-dependent language model.

For many problems in speech processing, it is useful to combine this model with another observation model (e.g. $p(f_t|e_t, m_t)$) that characterizes acoustic information $f_t$.

So far, we have seen how we can use the algorithms from the Likelihood, Decoding, and Learning processes above to answer the fundamental questions; on how does one compute $p(o)$, on what is the most likely state sequence $\text{argmax}_s\, p(s|o)$, also the provision of the most likely state at a particular time $t$, $\text{argmax}_j\, p(s_t = j|o)$, together with how does one estimate the parameters of an HMM.

All these questions are key when one tries to utilize HMM algorithm to de-identify textual records as answers to such questions provide the framework to obtain a solution to de-identification task. Further discussions on HMM application on the de-identification task can be seen in subsection 3.2.1 and the results obtained are posted on Table. 3.1.

### 2.3.3 Multilayer Perceptron

A perceptron is a linear classifier, i.e. an algorithm that characterizes the input by dividing a line into two classifications. The input is usually a parameter $x$ compounded by weights $w$ and a bias $b$. A perceptron generates a unique output centered on many re-valued inputs, using its input weights as a direct composite (and transferring the output often through the nonlinear activation mechanism).

Let us take a general neural network into account made of $L$ layers, excluding the input layer as shown in Figure.2.3. In addition, consider a layer $\ell$ arbitrary, having $N_\ell$ neurons as $X_1^{(\ell)}$, $X_2^{(\ell)}$, $\ldots$, $X_{N_\ell}^{(\ell)}$, with each $f^{(\ell)}$ transfer function. Keeping in mind that the transfer function could be different from stack(layer) to stack(layer). As in the extended Delta rule, any differentiable function (linear and non-linear) can be provided as the transfer function. As depicted in Figure 2.3, these neurons get neuronal signals from the previous layer, $\ell - 1$. For instance, neuron $X_j^{(\ell)}$ accepts a signal sent by $X_i^{(\ell-1)}$ with a weight variable $w_{ij}^{(\ell)}$. Hence, one has an $N_{\ell-1}$ by $N_\ell$ weight matrix, $\mathbf{W}^{(\ell)}$, the elements of which are provided as $W_{ij}^{(\ell)}$, for $i = 1, 2, 3, \ldots, N_{\ell-1}$ and $j = 1, 2, 3, \ldots, N_\ell$. Neuron $X_j^{(\ell)}$ has an activation of $a_j^{(\ell)}$ and a bias as well given by $b_j^{(\ell)}$.

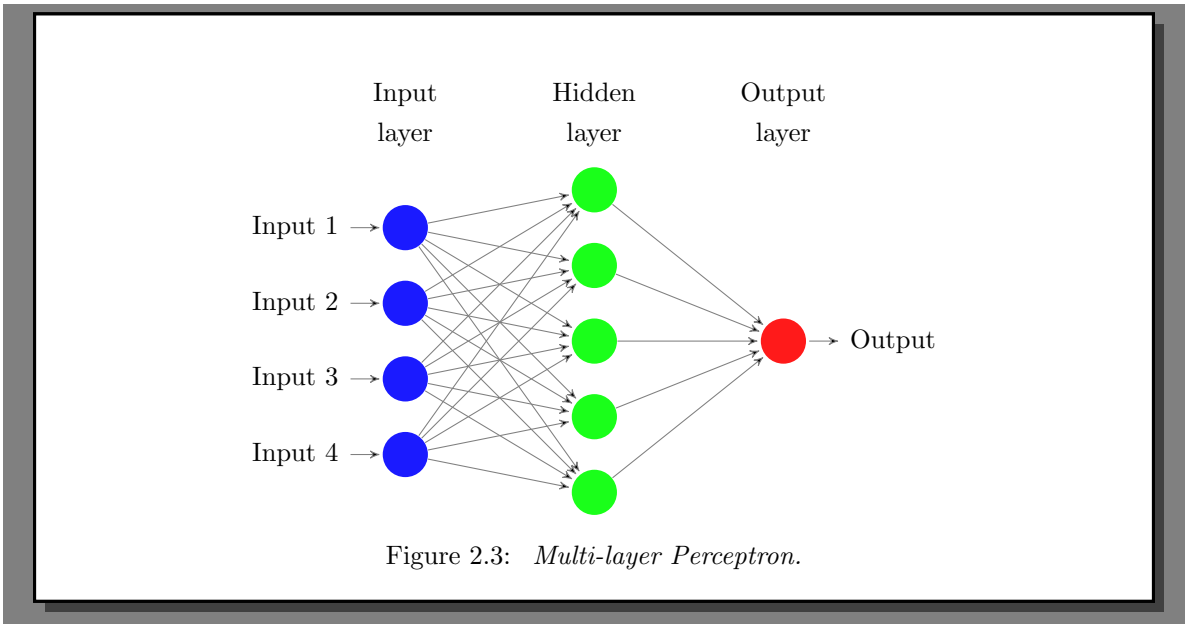



Figure 2.3: *Multi-layer Perceptron.*

Denoting the net input into neuron $X_j^{(\ell)}$, and for simplicity, let's use $n_j^{(\ell)} (= y_{\text{in},j})$. This can therefore be

presented as;

$$n_j^{(\ell)} = \sum_{i=1}^{N_{\ell-1}} a_i^{(\ell-1)} w_{ij}^{(\ell)} + b_j^{(\ell)}, \qquad j = 1, 2, \ldots, N_\ell.$$

We can therefore have the activation of neuron $X_j^{(\ell)}$ as shown in this next segment.

$$a_j^{(\ell)} = f^{(\ell)}(n_j^{(\ell)}) = f^{(\ell)}(\sum_{i=1}^{N_{\ell-1}} a_i^{(\ell-1)} w_{ij}^{(\ell)} + b_j^{(\ell)})$$

where; $f^{(\ell)}$ is the transfer function.

By considering the zeroth layer being the input layer and assuming we have an input scalar $\mathbf{x}$, which possesses $N$ elements, hence $N_0 = N$; whereby, the activation $a_i^{(0)} = x_i, i = 1, 2, 3, \ldots, N_0$ are present in the neurons of the input stack. If we assume that the output scalar $\mathbf{y}$ having $M$ elements and the output stack $L$ of the network; then we can conclude that $N_L = M$. We can write these as; $y_j = a_j^{(L)}, j = 1, 2, 3, \ldots, M$. for each given pair of weights and biases. If given an input vector, we can then use the equations above to determine the activation for every neuron.

### 2.3.4 Backpropagation

Researches discovered the inherent limitation capabilities of Single-layered networks in solving only linearly separable classification problems; hence, they proposed multi-layer networks to to address such limitation. Until Werbos did his 1974 thesis, when the single-layer networks training algorithms were widely inferred to the multi-layer networks models. A number of people in the neural network community rediscovered Werbos work in the middle 1980s. The backpropagation (BP) is a training algorithm based on the Delta (or LMS) single-layer perceptron generality rule —currently the most widely used neural network also featuring the differentiable transfer function within multi-layer networks.

#### 2.3.4.1 The algorithm for Backpropagation

When considering training and use of the Backpropagation (BP) algorithm to create rather general multi-layer perceptron for pattern recognition; together with having the assumption that because the training is carried out supervised on a given a pair of pattern sets (or associations): $\mathbf{s}^{(q)} : \mathbf{t}^{(q)}, q = 1, 2, 3, \ldots, Q$, we can then present the training scalars $\mathbf{s}^{(q)}$ with $N$ elements as;

$$\mathbf{s}^{(q)} = \begin{bmatrix} s_1^{(q)} & s_2^{(q)} & \ldots & s_N^{(q)} \end{bmatrix},$$

and the respective targets $\mathbf{t}^{(q)}$ with $M$ elements as;

$$\mathbf{t}^{(q)} = \begin{bmatrix} t_1^{(q)} & t_2^{(q)} & \ldots & t_M^{(q)} \end{bmatrix}.$$

The training vectors, in a way similar to the Delta rule, are presented to the network methodically each after another. During the training process; at time interval $t$, a training sample $\mathbf{s}^{(q)}$ for a specific $q$ is submitted as input $\mathbf{x}(t)$ to the network. Using the equations in the last section; we can obtain the forward-propagated network input signal together with the latest pair of weights and biases to receive the respective network output, $\mathbf{y}(t)$. Using the square error minimization technique on the steepest descent algorithm for this training vector; thus, we can then adjust the weights and biases as:

$$E = \|\mathbf{y}(t) - \mathbf{t}(t)\|^2,$$

where the respective target scalar for the selected training vector $\mathbf{s}^{(q)}$ is presented as $\mathbf{t}(t) = \mathbf{t}^{(q)}$. In addition, $\mathbf{y}(t)$ depends on the square error $E$, which is the whole network's parameter representation for all weight and bias components. From the steepest descent algorithm, we can obtain:

$$w_{ij}^{(\ell)}(t+1) = w_{ij}^{(\ell)}(t) - \alpha \, \frac{\partial E}{\partial w_{ij}^{(\ell)}(t)}$$

$$b_{j}^{(\ell)}(t+1) = b_{j}^{(\ell)}(t) - \alpha \, \frac{\partial E}{\partial b_{j}^{(\ell)}(t)},$$

where $\alpha(> 0)$ is the learning rate. As mentioned before, $E$ is weights-and-biases dependent, consequently, it too depends on the activation of the last layer, $\mathbf{a}^{(L)}$ and the network output $\mathbf{y}(t)$. It therefore can be said that; the aggregate input to the $L-$th stack, $\mathbf{n}^{(L)}$, depends on the input, the activation of the previous stack, and the weights and biases of the layer $L$. In summary, we can express the relation as the dependence on step $t$ is omitted as follows:

$$\begin{aligned} E &= \|\mathbf{y} - \mathbf{t}(t)\|^2 = \|\mathbf{a}^{(L)} - \mathbf{t}(t)\|^2 = \|f^{(L)}(\mathbf{n}^{(L)}) - \mathbf{t}(t)\|^2 \\ &= \left\| f^{(L)} \left( \sum_{i=1}^{N_{L-1}} a_i^{(L-1)} w_{ij}^{(L)} + b_j^{(L)} \right) - \mathbf{t}(t) \right\|^2 . \end{aligned}$$

Consequently, using the differential equations' chain rule, we can compute the partial derivatives of $E$ with respect to the components of $\mathbf{b}^{(L)}$ and $\mathbf{W}^{(L)}$ to have:

$$\frac{\partial E}{\partial w_{ij}^{(L)}} = \sum_{n=1}^{N_L} \frac{\partial E}{\partial n_n^{(L)}} \frac{\partial n_n^{(L)}}{\partial w_{ij}^{(L)}}.$$

For a general layer $\ell$, we can now explicitly say the sensitivity vector to have the following elements:

$$s_n^{(\ell)} = \frac{\partial E}{\partial n_n^{(\ell)}} \qquad n = 1, 2, 3, \ldots, N_\ell.$$

The $s_n^{(\ell)}$, also referred as the neuron $X_n^{(\ell)}$ sensitivity, offers the resulting error modification, $E$, per every incremental, $n_n^{(\ell)}$, entry received. Again, using chain rule to compute the sensitivity vector for layer $L$; we can obtain;

$$s_n^{(L)} = 2 \left( a_n^{(L)} - t_n(t) \right) \dot{f}^{(L)}(n_n^{(L)}), \qquad n = 1, 2, 3, \ldots, N_L.$$

where; $\dot{f}$ represents the derivative of the transfer mathematical expressions denoted by $f$. Hence,

$$\frac{\partial n_n^{(L)}}{\partial w_{ij}^{(L)}} = \frac{\partial}{\partial w_{ij}^{(L)}} \left( \sum_{m=1}^{N_{L-1}} a_m^{(L-1)} w_{mn}^{(L)} + b_n^{(L)} \right) = \delta_{nj} a_i^{(L-1)}.$$

Therefore,

$$\frac{\partial E}{\partial w_{ij}^{(L)}} = a_i^{(L-1)} \, s_j^{(L)}.$$

Similarly,

$$\frac{\partial E}{\partial b_j^{(L)}} = \sum_{n=1}^{N_L} \frac{\partial E}{\partial n_n^{(L)}} \frac{\partial n_n^{(L)}}{\partial b_j^{(L)}},$$

and since

$$\frac{\partial n_n^{(L)}}{\partial b_j^{(L)}} = \delta_{nj},$$

then

$$\frac{\partial E}{\partial b_j^{(L)}} = s_j^{(L)}.$$

We can then generalize $\ell$ layer, by writing [102];

$$\frac{\partial E}{\partial w_{ij}^{(\ell)}} = \sum_{n=1}^{N_\ell} \frac{\partial E}{\partial n_n^{(\ell)}} \frac{\partial n_n^{(\ell)}}{\partial w_{ij}^{(\ell)}} = \sum_{n=1}^{N_\ell} s_n^{(\ell)} \frac{\partial n_n^{(\ell)}}{\partial w_{ij}^{(\ell)}}.$$

$$\frac{\partial E}{\partial b_j^{(\ell)}} = \sum_{n=1}^{N_\ell} \frac{\partial E}{\partial n_n^{(\ell)}} \frac{\partial n_n^{(\ell)}}{\partial b_j^{(\ell)}} = \sum_{n=1}^{N_\ell} s_n^{(\ell)} \frac{\partial n_n^{(\ell)}}{\partial b_j^{(\ell)}}.$$

Since

$$n_n^{(\ell)} = \sum_{m=1}^{N_{\ell-1}} a_m^{(\ell-1)} w_{mn}^{(\ell)} + b_n^{(\ell)}, \qquad j = 1, 2, \ldots, N_\ell,$$

then

$$\frac{\partial n_n^{(\ell)}}{\partial w_{ij}^{(\ell)}} = \delta_{nj} a_i^{(\ell-1)}$$

$$\frac{\partial n_n^{(\ell)}}{\partial b_j^{(\ell)}} = \delta_{nj},$$

and so

$$\frac{\partial E}{\partial w_{ij}^{(\ell)}} = a_i^{(\ell-1)}\ s_j^{(\ell)}, \qquad \frac{\partial E}{\partial b_j^{(\ell)}} = s_j^{(\ell)}.$$

Once we aggregate back the $t$-step index dependent parameter, and reviewing rules for the weights and biases; we can then obtain:

$$w_{ij}^{(\ell)}(t+1) = w_{ij}^{(\ell)}(t) - \alpha\ a_i^{(\ell-1)}(t)\ s_j^{(\ell)}(t)$$

$$b_j^{(\ell)}(t+1) = b_j^{(\ell)}(t) - \alpha\ s_j^{(\ell)}(t),$$

To apply such corresponding update components, one needs to calculate the vectors of responsiveness $\mathbf{s}^{(\ell)}$ for $\ell = 1, 2, 3, \ldots, L-1$, by its conventional description given as;

$$s_j^{(\ell)} = \frac{\partial E}{\partial n_j^{(\ell)}} \qquad j = 1, 2, 3, \ldots, N_\ell.$$

So far, we have seen that $E$ depends on $n_j^{(\ell)}$. It is important to note that $n_j^{(\ell)}$ depends on $n_i^{(\ell-1)}$ for $i = 1, 2, \ldots, N_{\ell-1}$, due to the fact that layer $\ell$ aggregate input is dependent on the preceding layer $\ell - 1$ activation. We can conclusively say that, $n_j^{(\ell)}$ is also dependent upon the aggregate input for layer $\ell - 1$. In particular,

$$n_j^{(\ell)} = \sum_{i=1}^{N_{\ell-1}} a_i^{(\ell-1)} w_{ij}^{(\ell)} + b_j^{(\ell)} = \sum_{i=1}^{N_{\ell-1}} f^{(\ell-1)}(n_i^{(\ell-1)}) w_{ij}^{(\ell)} + b_j^{(\ell)}$$

for $j = 1, 2, \ldots, N_\ell$. Hence, for the sensitivity of layer $\ell - 1$ [102];

$$\begin{aligned} s_j^{(\ell-1)} &= \frac{\partial E}{\partial n_j^{(\ell-1)}} = \sum_{i=1}^{N_\ell} \frac{\partial E}{\partial n_i^{(\ell)}} \frac{\partial n_i^{(\ell)}}{\partial n_j^{(\ell-1)}} \\ &= \sum_{i=1}^{N_\ell} s_i^{(\ell)} \frac{\partial}{\partial n_j^{(\ell-1)}} \left( \sum_{m=1}^{N_{\ell-1}} f^{(\ell-1)}(n_m^{(\ell-1)}) w_{mi}^{(\ell)} + b_i^{(\ell)} \right) \\ &= \sum_{i=1}^{N_\ell} s_i^{(\ell)} \dot{f}^{(\ell-1)}(n_j^{(\ell-1)}) w_{ji}^{(\ell)} = \dot{f}^{(\ell-1)}(n_j^{(\ell-1)}) \sum_{i=1}^{N_\ell} w_{ji}^{(\ell)} s_i^{(\ell)}. \end{aligned}$$

Thus, sensitivities of all the neurons in layer $\ell$ have direct impact on the sensitivity of a neuron in layer $\ell - 1$.This is a iterators relationship for network sensitivities because the last layer $L$ sensitivities are inherently included in that consideration. For each layer; to find the activation or the net aggregate inputs, the input must be fed from the left side of the network and forwarded to the relevant subsequent layer [65]. But it's not. For the sensitivities of any particularly provided layer; the last layer needs to be the starting point and use the recursion relationship to return to that specific layer in question. The training algorithm is called **Backpropagation** because of recursive relationship of each layer; whereby, the error at the output is propagated to the subsequent shallow layers.

In summary, we can depict the multi-layer sensor training search algorithm for Backpropagation as [65]:

1. Activation setting for the $\alpha$ term together with weight and bias instantiating.
2. During $t = 1, 2, 3, \ldots$, iterate steps a to e below over and over again till convergence is attained.

   a Randomly picked from training dataset, Set $\mathbf{a}^{(0)} = \mathbf{x}(t)$ .

   b In each and every $\ell = 1, 2, \ldots, L$, calculate;

$$\mathbf{n}^{(\ell)} = \mathbf{a}^{(\ell-1)}\mathbf{W}^{(\ell)} + \mathbf{b}^{(\ell)} \qquad \mathbf{a}^{(\ell)} = f^{(\ell)}(\mathbf{n}^{(\ell)}).$$

   c Calculate for $n = 1, 2, 3, \ldots, N_{L-1}, N_L$ the value of;

$$s_n^{(L)} = 2\left(\mathbf{a}_n^{(L)} - \mathbf{t}_n(t)\right)\dot{f}^{(L)}(\mathbf{n}_n^{(L)}).$$

   d In each and every $\ell = L - 1, \ldots, 2, 1$ and $j = 1, 2, \ldots, N_\ell$, calculate;

$$s_j^{(\ell)} = \dot{f}^{(\ell)}(n_j^{(\ell)}) \sum_{i=1}^{N_{\ell+1}} w_{ji}^{(\ell+1)} s_i^{(\ell+1)}.$$

   e Update weights and biases for $\ell = 1, 2, \ldots, L$ as;

$$w_{ij}^{(\ell)}(t+1) = w_{ij}^{(\ell)}(t) - \alpha\, a_i^{(\ell-1)}(t)\, s_j^{(\ell)}(t),$$
$$b_j^{(\ell)}(t+1) = b_j^{(\ell)}(t) - \alpha\, s_j^{(\ell)}(t).$$

One has to computationally determine the activation together with the sensitivities for every one of the layers in order to calculate the updates for weights and biases. To achieve sensitivity, one needs $\dot{f}^{(\ell)}(n_j^{(\ell)})$ —thus, keeping track of all the $n_j^{(\ell)}$ is key to accomplish everything [65].

The two mathematical expressions frequently utilized as the transfer components in the backpropagation algorithm in training neural networks are the Log-Sigmoid function and the hyperbolic tangent Sigmoid function. The differentiable Log Sigmoid function, with a smooth value and monotonously ranging from 0 to the value of 1 for all $x$ approaching 0, is mathematically expressed as;

$$f_{\text{logsig}}(x) = \frac{1}{1 + e^{-x}}$$

While the hyperbolic tangent Sigmoid mathematical expressions is also differentiable, however its range is smooth in value from $-1$ to the value of 1 for each and every $x$ approaching 0, is mathematically expressed as;

$$f_{\text{tansig}}(x) = \frac{1 - e^{-x}}{1 + e^{-x}} = \tanh(x/2)$$

Hence, respectively, the first derivatives of the Log-Sigmoid and the hyperbolic tangent Sigmoid functions

are given as [102]:

$$\dot{f}_{\text{logsig}}(x) = f_{\text{logsig}}(x)\,[1 - f_{\text{logsig}}(x)]$$

$$\dot{f}_{\text{tansig}}(x) = \frac{1}{2}\,[1 + f_{\text{tansig}}(x)]\,[1 - f_{\text{tansig}}(x)]$$

We know, given the information that $f^{(\ell)}(n_j^{(\ell)}) = a_j^{(\ell)}$; thus, the computational implementation of the neural network alleviate the need to keep track of $n_j^{(\ell)}$ at all; hence, frees up computer memory resource.

#### 2.3.4.2 Various forms of the Basic Backpropagation Algorithm

Since a multi-layer neural network with the basic Backpropagation can take exhaustively substantial long time (weeks) to train for many conveniently utilizable problems, then a number of variations in the basic Backpropagation mathematical iterations expressions can be used to speed up the convergence.

**Modified Target Values:** If bipolar-shaped output target vectors and hyperbolic tangent Sigmoid functions for the transfer functions for neurons in the output layer, then the necessary output values of $\pm 1$ occur at transmission function asymptotes; —since those values can never be reached. There must be extremely large levels of input in these neurons. This happens, however, only if the weights are ultimately very large. Therefore, convergence is usually slower than normal, in this case. Similar case occurs when binary vectors are utilized in conjunction with Sigmoid mathematical expressions.
A simpler solution is to see if the calculated results are in between a certain tolerance of certain desired targeted values and if the net has learned a certain training vector. This happens when the range for this tolerance, for the revised valuation, is between $\pm 0.8$ for the hyperbolic tangent Sigmoid mathematical expression, however one may actually prolong the convergence when using a modified values for the hyperbolic tangent Sigmoid too close to 0 [65].

**Alternative Transfer Functions:** When the arc-tangent mathematical expression function is used to depict a transfer mathematical expression function in Backpropagation; the asymptotic values of the function are approached more slowly compared to the hyperbolic tangent Sigmoid function. When scaled so that the mathematical expressions value lies between the window of $-1$ to 1, then the mathematical expressions can be expressed as [102];

$$f_{\text{arctan}}(x) = \frac{2}{\pi} arctan(x),$$

and we can compute the derivative of such a function and expressed as;

$$\dot{f}_{\text{arctan}}(x) = \frac{2}{\pi}\frac{1}{1 + x^2}.$$

A non-saturating transfer mathematical expression function might be used in some applications whereby saturation isn't essential. A great example (where there exist no benefit to saturation) is;

$$f_{\log}(x) = \begin{cases} \log(1 + x) & \text{for } x \geq 0 \\ -\log(1 + x) & \text{for } x < 0. \end{cases}$$

This is completely continuous. We can write its derivative as;

$$\dot{f}_{\log}(x) = \begin{cases} \frac{1}{1+x} & \text{for } x > 0 \\ \frac{1}{1-x} & \text{for } x < 0 \end{cases}$$

and is also non-discrete for all x, even for $x = 0$. Radial basis mathematical expressions functions are characterized by having non-negative responses that are located around specific input target values; also,

are often used as transfer functions in Backpropagation. A prevalent instance with such characteristics is the Gaussian mathematical expression function, which is given as;

$$f_{\text{gaussian}}(x) = \exp(-x^2)$$

This function has values situated around $x = 0$ and its derivative is given by [102];

$$\dot{f}_{\text{gaussian}}(x) = -2x \exp(-x^2) = -2x f_{\text{gaussian}}(x).$$

**Momentum:** The notion that prior weight modifications tend to affect the present trajectory of motion in weight space; therefore, a rapid approaching argument convergence is due to the fact that weight update formulas have a momentum term. In the simplest form of momentum-coupled Backpropagation; it is true that momentum is proportionate to weight difference of the last stage. This can be presented in a formulae as [102]:

$$\Delta w_{ij}^{(\ell)}(t+1) = \alpha \; a_i^{(\ell-1)}(t) \; s_j^{(\ell)}(t) + \mu \Delta w_{ij}^{(\ell)}(t),$$

where; $\mu$ is within the expected window between $(0, 1)$. Therefore, weight updates take place in combination with that of the latest gradient direction. The momentum makes it possible for the net to make considerably large weight changes, as long as the adjustment for several input patterns are in the same general direction. It also prevents a large reaction from a single training pattern to the error.

**Updating in Batches:** Because the whole dataset should not be transferred to the neural net in one go; the ideal solution is to split the dataset in batches, sets, and chunks. Batch size is described as the total count of training examples in a single set batch.

In certain instances; it is beneficial, instead of updating the weights after each pattern, to store the weight accumulation terms for several trends (or for complete epoch when there exist not too many variations) and to simply adjust weight once every weight term.

**Learning Rates that Permit Variability:** As with the original Delta rule in a single neural network stack; convergence could be quickly attained though varying learning rates. It is possible that each weight might have a respective learning rate. Such rates of learning might have a time-variant component as the training of the network progresses.

**Adaptive Slope:** The transfer function may have an adjustable parameter inserted to control the function slope. For example, swapping the use of the given hyperbolic tangent Sigmoid mathematical expression;

$$f_{\text{tansig}}(x) = \frac{1 - e^{-x}}{1 + e^{-x}} = \tanh(x/2)$$

with the hyperbolic tangent Sigmoid mathematical expression given by;

$$f_{\text{tansig}}(x) = \frac{1 - e^{-\sigma x}}{1 + e^{-\sigma x}} = \tanh(\sigma x/2),$$

whereby $\sigma(> 0)$ governs the enormity of the gradient. The derivative of the transfer function is directly proportional to the $\sigma$. The structure may allow each sensor to have its own transfer mathematical expression with a various gradient parameter [65]. Although the aggregated entry into the sensor $X_j^{(\ell)}$ could be provided by the formula below; however the weight, bias and gradient elements can be derived

[102];

$$n_j^{(\ell)} = \sum_{i=1}^{N_{\ell-1}} a_i^{(\ell-1)} w_{ij}^{(\ell)} + b_j^{(\ell)}, \qquad j = 1, 2, \ldots, N_\ell,$$

its activation can be presented as [102];

$$a_j^{(\ell)} = f^{(\ell)}(\sigma_j^{(\ell)} n_j^{(\ell)}) = f^{(\ell)} \left( \sigma_j^{(\ell)} \left[ \sum_{i=1}^{N_{\ell-1}} a_i^{(\ell-1)} w_{ij}^{(\ell)} + b_j^{(\ell)} \right] \right),$$

where; $\sigma_j^{(\ell)}$ is the gradient component for the sensor $X_j^{(\ell)}$. The output layer neurons sensitivities are therefore presented as [65];

$$s_n^{(L)} = 2 \left( a_n^{(L)} - t_n(t) \right) \dot{f}^{(L)}(\sigma_n^{(L)} n_n^{(L)}), \qquad n = 1, 2, \ldots, N_L.$$

From the recursion relation below; one can obtain the sensitivities of all the other layers;

$$s_j^{(\ell-1)} = \dot{f}^{(\ell-1)}(n_j^{(\ell-1)}) \sum_{i=1}^{N_\ell} w_{ji}^{(\ell)} s_i^{(\ell)} \sigma_i^{(\ell)}.$$

The update guidelines for the weights, biases, and gradient components are presented by the expressions [102];

$$\Delta w_{ij}^{(\ell)}(t) = -\alpha \; a_i^{(\ell-1)}(t) \; s_j^{(\ell)}(t) \; \sigma_j^{(\ell)}(t)$$

$$\Delta b_j^{(\ell)}(t) = -\alpha \; s_j^{(\ell)}(t) \; \sigma_j^{(\ell)}(t),$$

$$\Delta \sigma_j^{(\ell)}(t) = -\alpha \frac{\partial E}{\partial \sigma_j^{(\ell)}(t)} = \alpha \; s_j^{(\ell)}(t) \;\; n_j^{(\ell)}(t).$$

Typically, the slope parameters are initialized to 1, however an alternative value might be set if one has a great substitution. The application of variable gradients frequently enhances converging to a solution, despite inducing the expensive complicated per-step computations [65].

## 2.4 Conclusion

The object of the backpropagation prediction model is to alter the weights of a neural network and ultimately to diminish network error relative to the predicted input results. Recurrent neural networks, specifically the Bi_LSTMs, are used to change weight history over time through the back propagation through time (BPTT). In order to accurately identify sequence prediction issues in recurrent neural networks, one has to have a clear conception of what back propagating over time works and how fully customization changes such as truncated back propagation over time affect any network capacity, stability and speed.

I should mention that Mathematical methods covered in in section 2.3 are by no means exhaustive. There exist a tremendous amount of versatility in Multilayer Perceptrons (MLPs), Convolutional Neural Networks and Recurrent Neural Networks (RNNs) groups of networks that have proven to be useful and effective in various specific problems but not others. These various types of Neural Networks do have several sub-types or categories that have been developed to help them address a perceive deficiency and/or optimally specialize in numerous methodological approaches of prediction problems and data sets.

Convolutional Neural Networks (CNNs) were developed to model output variable image data. They have proven to be the best way to tackle any forecasting of image data as an input. CNNs function well with spatially-connected data. The CNN input is usually two-dimensional, a field or a matrix but

can be transformed into one-dimensional, enabling a one-dimensional internal sequence depiction to be created. This permits the use of CNN more broadly on other types of spatial data. A recurrent neural network (RNN) is a type of artificial neural networking, which forms a direct graph in a time sequence with connections between nodes. It can therefore show complex distribution pattern. RNNs may use their inner state (memory) from feedforward neural networks to model the streams of variable input lengths. Highly utilized in activities such as non-sectional, linked script or voice recognition.

In this chapter, I have covered the concepts of Backpropagation, HMM and MLPs. In Chapter 3 sections 3.2.1 and 3.2.2, I cover how de-identification is done under HMM and RNN respectively. In subsection 2.3.3; I cover the two types of classes of artificial neural networks Convolutional Neural Networks (CNNs) in subsection 4.1.3 and will move to discuss in detail theBi_LSTM which falls in Recurrent Neural Networks (RNNs) in section 4.2.

# Chapter 3

# Literature Review

## 3.1 Literature review on De-identification

I reviewed several articles and literature on the topic however the two main articles (1. *Evaluating the state-of-the-art in automatic de-identification* by Uzuner, Özlem and Luo, Yuan and Szolovits, Peter [126] and 2. *State-of-the-art anonymization of medical records using an iterative machine learning framework* by Szarvas, György and Farkas, Richárd and Busa-Fekete, Róbert ) [121] that laid the foundation and shaped the necessity for this research work– they also alluded to the fact that de-identification function is comparatively same as the traditional Named Entity Recognition (NER) task [121, 126]. The articles pointed out that beyond the bio-pharmaceutical field, the Message Understanding Conference (MUC) was used as an opportunity to assess NER systems [45]. The literature stressed the fact that De-identification algorithms need protecting medical records' information credibility and achieve this objective of finding and removing all PHI from medical records despite the presence of:

- *Ambiguities:* PHI and non-PHI can intertwine lexically, e.g., Asperger can, for example, be the terminology of an illness (non-PHI) as well as the name of an individual (PHI).
- *Out-of-vocabulary PHI:* PHI may encompass a misspelling or colloquial expressions not contained in dictionaries.

The articles examined multiple systems which were submitted to the Natural Language Processing (NLP) challenge organized by the i2b2 (Informatics for Integrating Biology to the Bedside) project, I will briefly summarize the findings here below:

- Guo et al. [46] used Support Vector Machines (SVM) [16] and the 'General Architecture for Text Engineering' (GATE) software package [17].
- It was also pointed out that Hara [47] adopted a Support Vector Machinery (SVMs) hybrid rules system. The Hara's rules captured phone number patterns, dates and identification patterns. Trained SVMs could identify medical facilities, sites, patients, physicians and ages for global and local features. The article noted that Hara used binary structured trees of 'n-gram trees' and dependency trees, by deploying the Boosting Algorithm for Classification Trees (BACT) [61]. Hara presented three passes, that vary widely but primarily in the categorization of the phrases.

  'Hara 1' [47] explored utilization of n-grams, 'Hara 2' [47] examined dependency forests and 'Hara 3' [47] completely skipped the paragraph segments categorization. The conclusion demonstrated that paragraph segments categorization impact performance [126].
- For de-identification, Szarvas et al. used an sequential named - entity recognizer [121]. To Szarvas de-identification is considered to be a categorization task and decisions forests with local traits and dictionaries are used. They submitted three packages: 'Szarvas 1, 2, and 3'. In 'Szarvas 1', they

used a boosted decision forests comprising AdaBoost engine [34] and C4.5 categorizers platform [82] trained using word-lookup database and lists, all local attributes, and a few syntactic prompts. In 'Szarvas 2', they combined the votes of three boosted categorizers trained on word-lookup database and lists, all local attributes, and all syntactic prompts. In 'Szarvas 3', they added vetted documents to 'Szarvas 1'.

- Wellner and colleagues developed two NER packages, LingPipe and Carafe, to tackle the de-identification work [127]. They submitted three packages. All using, HMM-driven package, LingPipe but 'Wellner 3' utilizing Carafe. The CRF-based application, Carafe, comprised dual features, an n-grams engine and location look-up table, equally accompanied by standard expressions capable of capturing the more standard PHI.

The de-identification performance evaluation conducted by Özlem Uzuner et al. [126] examined token and instance accuracy, recall, and f-measure. All assessments had been conducted with the same test corpus. After review of the literature; one can conclude that the machine learning techniques together with attributes selected modify the document sequence structure (orthography, tokens, POS tags, to say the least); however, do not completely capture PHIs. Largely, the variation in the recall might relatively be attributable to the applicable comprehensivity of the learning methodologies and functionalities. However, I should point out that these tests were carried long ago and tools and techniques have quite steadily improved ever since. Here are the additional observations:

- Similarly, for medical facilities sites; low performance might partly attributed to the unavailability of sufficiently large number of training examples. Merely 144 cases (302 tokens) from the medical facilities sites presented in the training set. The ability to learn places is additionally restricted by the fact that, unlike medical facilities recognized through syntactic indications like 'admitted to', 'transferred to', 'Medical Center', and 'Hospital', however fewer facilities sites syntactic cues given.
- For phone numbers; review of system reactions indicated that various utilities lacked exhaustively complete modules.
- On the issue of ambiguous and out-of-vocabulary PHI; it was concluded that some imprecise and out-of-vocabulary PHI causes skipped or selective PHI recognition. To conclude;
  - All six top systems typically worked great on patient data; quite a few unclear and unambiguous names of patients like Randdor So are missed. [126]
  - The unclear and out-of-vocabulary submissions in physicians challenge; all the top six systems. [126]
  - Locations marking errors are made by all the top six systems. [126]

Remarkably, it is not inconsequential to effectively delete dates. The authors have had two major unanswered questions. First; do the gains in solving this problem imply applicable success when extrapolated to similar performance in other untested sets of data? Secondly; should healthcare lawmakers depend on this level of achievement to allow patients to disclose health data automatically (or in semi-automation for scientific studies) while precluding the undue enormous risk?

It is without a doubt; many agree on the importance of de-identification to conserve the medically outstanding content of narratives, in order to benefit the lateral research while maintaining the record value for patient care. HIPAA (45 CFR 164.514) describes 18 classifications of PHI that have to be eliminated from a patient's health data before such record can be safely disclosed. Lately, however,studies in Canada have shown that in a period spanning 11-years, solely address records might lead to re-identification [26]. Likewise, American nationals can be positively identified by birth-date, ZIP-code, and sexual orientation information [40, 69], yet HIPAA PHI classifications do not

cover birth-date, ZIP-code, and sexual orientation information for adequately residential areas. This advocates and elevates HIPAA to be a baseline for efficient de-identification. In other words, while HIPAA is a starting point for effective de-identification, however for complete de-identification the legislation might contain deficiencies and is perceived to be inadequate [115]. There is a case to be made to examine further on Machine Learning techniques in order to come up with better de-identification tools.

A recent experiments results carried on the three systems below:

(i) The *MITRE's MIST* tool [1]: An open source utility package for de-identification loaded with PHI mapping and substitution capabilities.

(ii) The *VHA's BoB* package: Using cTAKES [106] engine to pre-process the records and built on the Apache UIMA architecture (Apache, 2008).

(iii) The *in-house utility tool for de-identification* [20] from Cincinnati Children's Hospital Medical Center's (CCHMC) system: Predicated around MALLET software [76] architecture,

These systems have overall performed well and elevated the stipulated standards and yet made a case for more investigation into Machine Learning techniques for de-identification. There can be many differences in the scores because the training data differences as every cohort had access to data, that at the time of access, might have been unavailable to others.

I, like many, agree on the importance of de-identification in the preservation of medically salient narrative content in order to ensure that this information is beneficial to downstream research and to preserve the patient care record value. I can't overstate on the need (or ability) for various researchers to access, analyze and utilize the information contained in the narrative clinical notes —and how access to such information could pave way to better medicine and smart patient care. The EHR narrative medical message; certainly, becomes increasingly recognized as a key element of computer-aided decision support, efficiency improvements, and service delivery. [81].

## 3.2 Review of related work

### 3.2.1 De-Identification task using HMMs

Predictive graphical representations are some of the NLP society's many significant resources. In specific, the capacity to train generative concepts utilizing Expectation-Maximization (EM), Variational Inference (VI), and sampling methods like MCMC has made it possible to build computational unsupervised tag and grammar exploration processes, alignment, topic models and more. Such latent variable models detect invisible structures in text that are aligned with recognized language events and with readily recognizable patterns.

Hidden Markov models seem to be easy generative concepts which work in several NLP functions such as Part-of-Speech tagging and Named Entity Recognition [73].Normally, these don't need a lot of features engineering. But, its powerful premise of independence assumption compromises its output reliability. For instance, since it mimics the mutual likelihood of tags and words, the Conditional Random Fields(CRF) is much more versatile [10].

A contextual circumstance cue has a crucial role to play throughout the de-identification task. As training data, generally, does not constitute all identifying phrases and several phrases may or may not be identified depending on the circumstances.

In HMM, contextual indices are represented by the likelihood of transition among labels, which determines how well HMM could comprehend contextual indications. Desirable training information would provide unique id labels as well as labeling reference terms. It would not be realistic to derive all reference phrases manually. New labels which are often specifically placed around descriptors will be provided as a prevalent automatic remedy.

With the amount of latent factors, the probability of a particular mixture model rises. A system with a big amount of latent parameters, though it is probable to over-fit and under-perform with fresh information, may appear well fitted to training information. When one compares HMMs and recurrent neural networks(RNNs) models will also have to examine the following two fronts:

(i) **Bayesian vs Maximum likelihood approximation:** Suppose the dataset X is best explained by a set of likelihood distribution variables, $\theta$. To assess the $\theta$ metrics using the Bayes Rule:

$$p(\theta|X) = \frac{p(X|\theta) * p(\theta)}{p(X)}$$
$$posterior = \frac{likelihood * prior}{evidence}$$

where; $p(\theta|X)$ is posterior probability that is the probability of the parameters $\theta$ given the evidence $X$, the likelihood $p(X|\theta)$ is the probability of the evidence $X$ given the parameters $\theta$, $p(\theta)$ is the probability distribution function for $\theta$.

- **Maximum Likelihood Estimate:** From Maximum Likelihood Estimate, the goal is to obtain a specific $\theta$ value that optimizes the probability, $p(\mathcal{X}|\theta)$, shown in the equation(s) above. This value can be referred to as $\hat{\theta}$. From computation of the Maximum Likelihood Estimate, $\hat{\theta}$ is the target estimate and not the random element. Maximum Likelihood Estimate regard the $\frac{p(\theta)}{p(\mathcal{X})}$ variable as a constant; and thus, does not enable us to introduce $p(\theta)$, which was our previous assumption, into approximate computations for the probable values for $\theta$.
- **Bayesian Estimate:** In contradistinction; the Bayesian estimate computes the posterior distribution $p(\theta|\mathcal{X})$ in full (or occasionally estimate). The $\theta$ is treated by Bayesian assertion as an arbitrary parameter. In a Bayesian assessment, one utilizes probability density function instead of a single event as in the Maximum Likelihood Estimate so as to deduce probable density features.

  Among the $\theta$ values from the yield distribution $p(\theta|\mathcal{X})$, one should pick the most feasible value. In the absence of a variation, for instance, one can choose the expected value of $\theta$. The fluctuation computed from the post-distribution parameter $\theta$ permits conveying confidence in any particular value approximated. When the variance is extremely big to obtain a decent $\theta$ estimate, one can declare $\theta$ to be non-existent. The realization that we now have to cope with the denominator in the Bayes principle is making Bayesian's assessment complicated as a compromise, i.e. evidence. Hence, the evidence or probability of evidence is given by:

$$p(X) = \int_{\theta} p(X|\theta) * p(\theta) d\theta$$

where; the notation $p(X)$ stands for the probability of $X$ and where $p(X|\theta)$ means the conditional probability of $X$ given $\theta$. Hence, the likelihood $p(X|\theta)$ is the probability of the evidence $X$ given the parameters $\theta$, the probability distribution function for the evidence is

$p(X)$, and $p(\theta)$ is the probability distribution function for $\theta$.

When HMM is examined; the best possible parameters may be calculated with closed-shape solutions but not applicable to HMM with latent variables. This is due to the fact that aggregation of latent variables does not equate the logarithmic function's representation.

An expectation maximization (EM) mechanism enables deriving closed form expressions for modifications and offers a sequential algorithm for finding an optimal solution locally. Bayesian model studying represents a more computational conundrum because it not only necessitates that the latent variables are summarized but also the prior record are integrated [10].

Several sampling methods research have given a useful tool to explore Bayesian concepts [44]. However, selection techniques have to accomplish the generally challenging job of blending diagnosis to guarantee that the specimens produced to follow the expected distribution [59]. Moreover, debugging is much harder because separate runs produce distinct specimens [10].

(ii) **Deterministic vs Random:** RNN is an HMM's deterministic estimation (where learning patterns are involved). Also, the transition and emission probability conditional distributions are deterministic. In that manner, based on the input, there exists only one output for the RNN. But in an HMM, several results based on the random variable of input are feasible— in other words, what is a variable in the RNN it is a random variable in the HMM.

The result of these two constraints is that complicated transitional probability distributions and emission conditional probability distributions can be trained far more easily in the case of RNN, whereas it is much harder for HMM.

When it comes to the results obtained in the study by Chen et al. [10] seen on the Table.3.1; the performance of HMM-based methods fall behind the top performing CRF models in the challenge. Such a conclusion motivates for one to embark on examining what RNNs offer to achieve a better option to tackle the de-identification task.

The Dirichlet method models an innumerable amount of latent elements hypothetically but an infinite number can not be calculated, hence Dirichlet is stopped in reality at K [9]. In the challenge, Chen et al. set $K = 20$ for non-identifier tags and $K = 8$ for large category identifier tags to limit the computational cost [10]. The conceptual HMM-DP method is characterized as: every tag includes an infinite amount; the sub tag of a tag establishes the subsequent tag ; the sub tag of a tag and the subsequent tag dictate the sub tag of the next tag ; the word at place is set by the sub-tag of a tag [10].

| | HMM-DP submitted | | | HMM-DP updated | | | HMM-DP+CRF | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| NAME (4829) | 0.922 | 0.910 | 0.916 | 0.949 | 0.924 | 0.936 | 0.939 | 0.936 | 0.937 |
| PROFESSION (340) | 0.716 | 0.297 | 0.420 | 0.825 | 0.553 | 0.662 | 0.847 | 0.650 | 0.735 |
| LOCATION (3001) | 0.861 | 0.809 | 0.834 | 0.979 | 0.814 | 0.889 | 0.964 | 0.860 | 0.909 |
| AGE (790) | 0.786 | 0.475 | 0.592 | 0.788 | 0.481 | 0.597 | 0.902 | 0.724 | 0.803 |
| DATE (12518) | 0.979 | 0.926 | 0.951 | 0.981 | 0.926 | 0.953 | 0.982 | 0.934 | 0.957 |
| CONTACT (419) | 0.963 | 0.876 | 0.918 | 0.972 | 0.897 | 0.933 | 0.979 | 0.905 | 0.940 |
| ID (1126) | 0.942 | 0.874 | 0.906 | 0.991 | 0.877 | 0.931 | 0.991 | 0.877 | 0.931 |
| TOTAL (23047) | 0.943 | 0.879 | 0.910 | 0.968 | 0.887 | 0.926 | 0.966 | 0.910 | 0.937 |

Table 3.1: Results on category. The HMM-DP used in the challenge is denoted as "HMM-DP submitted", the HMM-DP developed after the challenged is denoted as "HMM-DP updated", and the HMM-DP with CRF alignment is denoted as "HMM-DP+CRF" [10].

The primary benefit of using a recurring algorithm over Markov chains and hidden Markov model is that the neural networks' representational strength and their capacity to execute smart modeling, along with transfer learning, takes syntactic and semantic characteristics into consideration. This is yet another benefit of using RNNs over HMMs.

### 3.2.2 Prior De-Identification using RNNs

A multidisciplinary group of US-VHA co-operative researchers from the Consortium for Healthcare Informatics Research (CHIR) are tasked with the goal to improve health for veterans through basic and applied computer research and to foster the efficiency of unstructured and other clinical data in the EHR [1].

The CHIR de-identification taskforce examined the latest state-of-the-art automated PHI-driven de-identification application solutions for textual medical information [80] in order to support the development of and research for a best-of-breed VHA textual medical information and records [28] de-identification application software. In addition, CHIR assessed the overall influence on future analytical functions together with the real threat of re-identification from de-identified clinical texts.

Meystre, Stéphane et al. in [79] assessed a variety of de-identification systems within the framework of the CHIR de-identification task-force. Meystre, Stéphane et al. presented analytical effect of automated textual de-identification on content in medical notes premised on a gradual methodology. Meystre, Stéphane et al. began with an advanced assessment of the review on interpretability and typesetting of clinical notes as well as a thorough analysis of the adverse consequences for certain types of textual medical information as seen on Figure 3.1. Every step was guided by a question from the studies and comprised one of the studies as depicted in the figure.

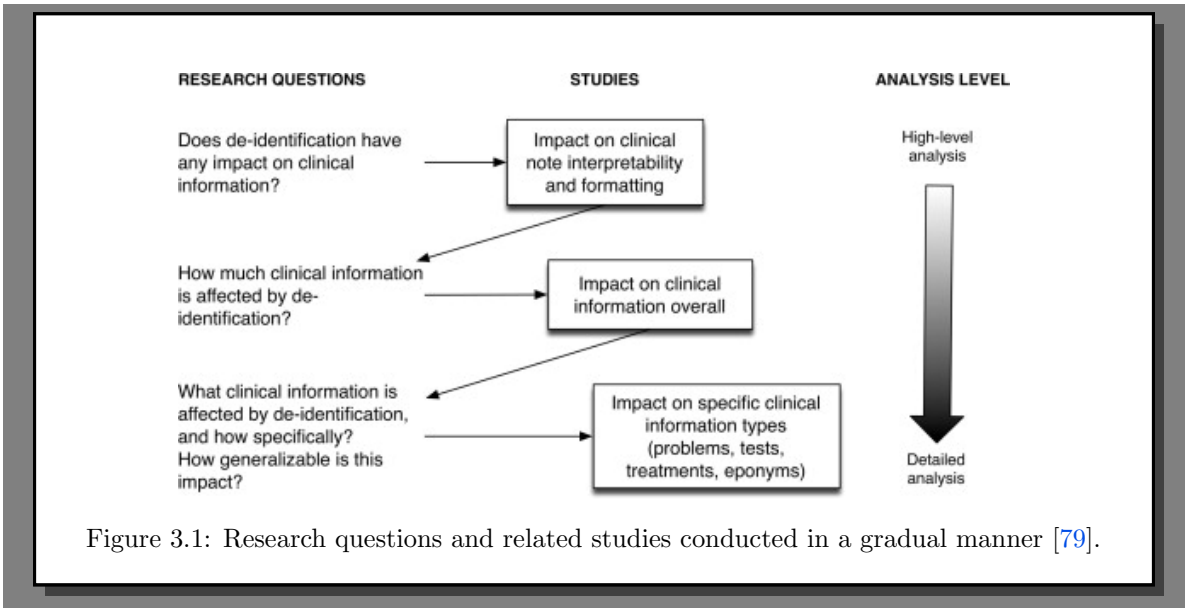



Figure 3.1: Research questions and related studies conducted in a gradual manner [79].

Five distinct textual medical information de-identification application programs were employed to investigate and examine the informativeness (i.e. information on document type, textual medical information or medical data types and interpretation and conclusion) of VHA textual medical information notes. It was reported that these systems had altered the information effectiveness minimally and that only one system had altered the formatting. While investigating the effect of de-identifying process on the extraction of clinical information; a comparison of the SNOMED-CT concepts found in the original (i.e. not de-identified) version of the VHA textual medical information note corpus and within the same

[1] Consortium for Health Care Informatics Research(CHIR) primary research objective is the development of methods and tools needed for an effective text processing based research program. https://medicine.utah.edu/internalmedicine/epidemiology/research_programs/chir_info.php

post-identification corpus by an open-source textual medical information extraction software utility. Barely, about 1.2–3% shy, SNOMED-CT concepts could be unearthed in de-identified editions of the textual medical information records, and majority of these notions were PHI that were inaccurately marked as medical information. [79]

The study investigated the commonality of shortlisted clinical PHI information annotated in the 2010 i2b2 NLP challenge corpus [79] and electronically selected PHI annotations from the best-of-breed (BoB') VHA clinical text de-identification system. I should also point out that a VHA textual medical information corpus was a subsection of a reference standard made by eight hundred de-identified textual medical information transcripts performed by hand. Overall, only 0.81% of the clinical information exactly overlapped with PHI, and 1.78% partly overlapped.

## 3.3 How the previous study was conducted

The experiments presented by Stéphane et al. [79] predicated on two distinct corpora of textual medical information:

1. The corpus from the 2010 i2b2 NLP task competition.
2. Corpus of VHA clinical notes. The VHA textual medical information corpus was a subsection of a reference standard made by eight hundred de-identified textual medical information transcripts performed by hand.

### 3.3.1 Methodology used by previous study(Stéphane et al.)

First, the documents were selected from the one hundred (100) most frequent textual medical information note types available in a vast VHA research registry using a class division random selection methodology. Second, two reviewers recorded each selected document and a third reviewer adjudicated discrepancies. Last but not least, the third examiner noted that further clarification was necessary, while a fourth and final examination reviewed any unambiguous or cases arbitration. It is important to note that these tasks were based on the markup rules and data structure-based mechanism for the 18 PHI categories stipulated as per HIPAA legislation on 'Safe Harbor' section. Figure 3.2 provide the schematic depiction of the process used. Beyond the provisions of this Regulation, Meystre, Stéphane et al. embraced a pragmatic methodology, and included organizational annotation class (Other Organization Nomenclature), references to healthcare facilities (Health Unit Name), military specific information (Deployment). All annotation tasks have been performed with the an open source annotation tool, Knowtator [93].

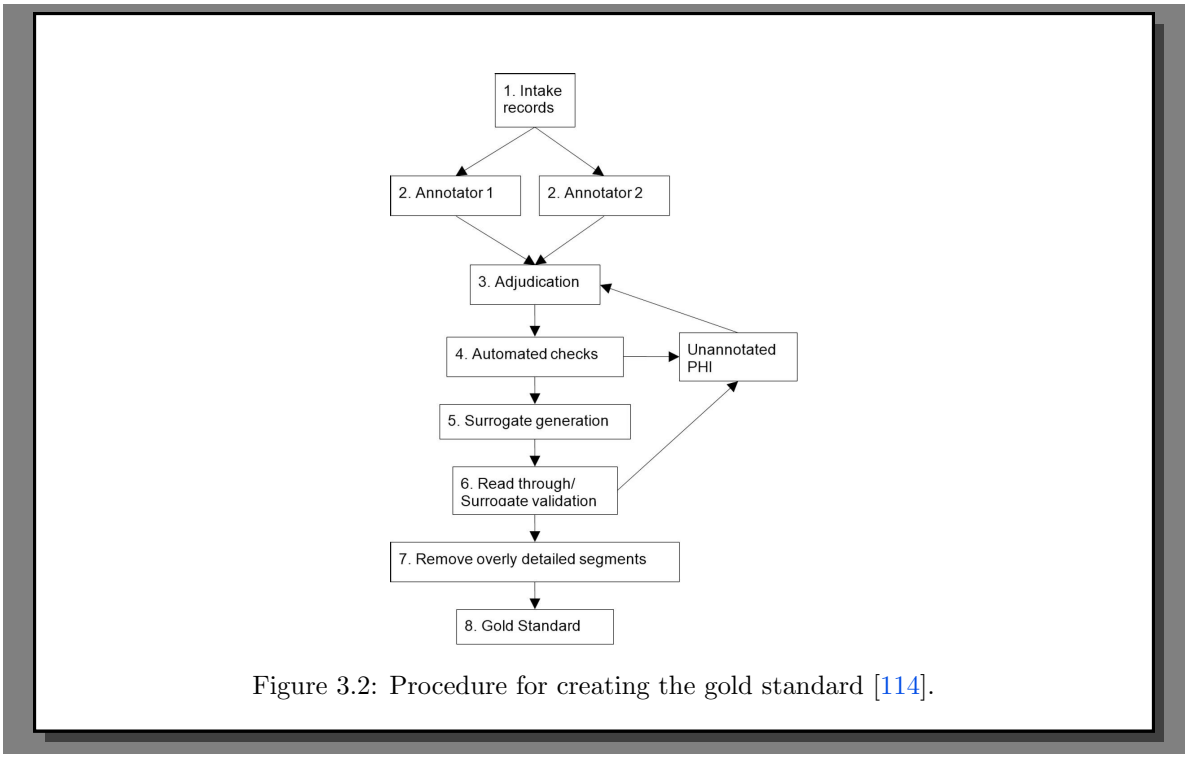



Figure 3.2: Procedure for creating the gold standard [114].

### 3.3.2 Interpretability and formatting impact on De-identified clinical notes

Five non-commercially accessible textual medical information de-identification utility tools were evaluated in Stéphane et al. appraisal [79]. Those are the Medical De-identification System (MeDS [35]), the HMS Scrubber (HMS [5]), the Health Information DE-identification (HIDE [36]Gardner:2010vj), the MITRE Identification Scrubber Toolkit (MIST [1]) and lastly, the MIT System (MIT [90]). The assessment conducted by Stéphane et al. categorized the assessments under the following classifications [79]:

1. Assessment on the de-Identified reports interpretability.
2. Assessment on the de-Identified reports‘ document structure changes.

## 3.4 Assessment of De-identified reports

### 3.4.1 Assessment on the De-identified reports interpretability

Stéphane et al. reviewed the sub-set of such documents, as five selected systems (listed above) automatically discovered them. A randomly chosen subgroup of fifty (50) hand-noted for identification of PHI and clinical eponyms was selected by Stéphane et al. as explained above. It had 1205 PHI edits with textual medical information edits, findings, or interpretations performed by hand.

Medical eponyms are the clinical name or noun formed after a person (include references to diseases, procedure, apparatus or anatomy containing the names of the individuals or places). Some examples of this include Alzheimer's, fundoplication of Nissen, Hutchison's disease, catheter of Swan-Ganz, Aarskog's syndrome, or tendon of Achilles. The type of report was characterized in the document as 'Discharge sheet Summary'. Stéphane et al. created the '*interpretability score*' (IS) for every textual medical information de-identification software package that was a three-tier ranking system that defined **IS1** when significantly substantial textual medical information retained (1 for 'yes' score, 0.5 for 'partial' score, and 0 for 'no' score), **IS2** when the record type or attributes of textual medical information retained ((1 for 'yes' score, 0.5 for 'partial' score, and 0 for 'no' score) and lastly, **IS3** when conclusion or interpretation preserved (1 for 'yes' score, 0.5 for 'partial' score, and 0 for 'no' score).

### 3.4.2 Examination of de-identified reports' format alterations

A manual assessment or appraisal was carried out in order to evaluate the extent to which the formatting of the original report has been maintained in the reports de-identified. Textual medical information document formatting is defined as a document structure that consists of lexical characteristics such as word case structure, paragraph breaks, line width and indent locations, punctuation, and line numbers. This was a descriptive and qualitative statistics and detailed formatting changes were not registered.

The study used BoB to create 7 distinct editions of a random subset of three hundred VHA corpus textual medical information records. These textual medical information aggregated records editions [79] were classified as;

(a) The original textual medical information records (not records with identities removed edition).

(b) Records with identities removed edition by 4 binary SVM-filters and re-synthesized PHI labels.

(c) Records with identities removed edition using 4 binary SVM filters and a general 'PHI' label.

(d) Records with identities removed edition using 4 binary SVM-filters and PHI labels that included categories.

(e) Records with identities removed edition using 1 multi-class SVM-filter and re-synthesized PHI labels.

(f) Records with identities removed edition using 1 multi-class SVM-filter and a general 'PHI' label. and finally,

(g) The records with identities removed edition using 1 multi-class SVM-filter and PHI labels that included classifiers.

The study utilized cTAKES version 3.0.0, with a AggregatePlaintextUMLS processor as the analysis core engine to de-identify while using different PHI hiding methods and approaches as an exploit (in its configuration) to harvest such clinical concepts from the VHA textual medical information records. The cTAKES were restricted to the look-up database SNOMED-CT and were supported by the corpus' seven distinct editions. The pairing of the original corpus with any de-identified editions of the corpus was then made by Stéphane et al. and all SNOMED-CT concepts were harvested from every record of the textual medical information records.

For the statistical analysis, a zero assumption showed that there was no distinction in premise tally in proportion between the unmodified medical information records edition of the corpus (used by Stéphane et al.) and an unspecified version, mean level 0.05, and two distinct methodologies: a bundled Student Test (2-tailed) to compile textual medical information records editions and a log-likelihood analogy-coefficient [99] technique, i.e. for multiple comparisons, Stéphane et al. used correction Bonferroni correction for multiple comparisons [79].

The Expected count E1 together with Expected count E2 provided comparison for each de-identified textual medical information records editions of the corpus against the original. These Expected count($E_i$) were then calculated as follows;

$$E_1 = c(a+b)/(c+d)$$
$$E_2 = d(a+b)/(c+d)$$

where; $a$ is concept qualification in de-identified textual medical information records, $b$ is concept qualification in original textual medical information records, $c$ is all concepts' qualification in de-identified textual medical information records, and $d$ is all concepts' qualification in original textual medical information records. Moreover, the log-likelihood value LL for every conceptual notion in each de-identified textual medical information records edition of the corpus was calculated as [79]:

$$LL = 2[(a \cdot ln(a/E_1)) + (b \cdot ln(b/E_2))]$$

#### 3.4.2.1 De-identification process effects on Report's specific clinical information

Stéphane et al. utilized the 2010 i2b2 NLP challenge textual medical information records as well as the VHA records [79] as an evaluation of how generalizable the results could be in the VHA corpus and the detailed the effects of de-identification process.

#### 3.4.2.2 De-identification process effects on reported medical problems, diagnoses, and therapies

The NLP Contest for i2b2 2010 focused on the extraction, testing, treatment and local context (e.g., 'new abdominal pain') of medical problems as well as on extracting special relationships between those concepts. The study by Stéphane et al. used the following to instantaneously annotate all PHIs that could appear in i2b2 textual medical information records or documents. The software developed by Stéphane et al. referred as best-of-breed(or "BoB") [28] also annotated medical name or noun formed after a person (include references to diseases, procedure, apparatus or anatomy containing the names of the individuals or places) that were not in the PHI list. This was a precaution knowing that these medical names or nouns could mistakenly grouped for sensitive PHI categorization like 'individual names'. Since consequent usage of the textual medical information records in other tasks such as harvesting medical data is determined by overlap rate; the analysis was performed for overlapping automated PHI annotations with the reference standard of the 2010 i2b2 NLP problem, testing, and treatment annotation were also performed.

#### 3.4.2.3 Clinical eponyms impact from Report de-identification

The second VHA corpus evaluation evaluated the number of instances in which the PHI clinical eponyms were identified in an evaluated detection system, in which the trial identified all the human notations of clinical name or noun formed after a person (include references to diseases, procedure, apparatus or anatomy containing the names of the individuals or places) and grouped such data into four groupings as devices, diseases, anatomy, and procedures.

### 3.4.3 Summary of Results

The assessment conducted by Stéphane et al. categorized the Summary of Results under the following classifications [79]:

1. Assessment on the de-Identified reports interpretability.
2. Assessment on the de-Identified reports' document structure changes

#### 3.4.3.1 Assessment on the de-Identified reports interpretability

Not all of the 50 documents were reported to have explicitly specified types of report. About thirty percent of the training textual medical information records did not indicate the record type explicitly prior to de-identification. Each system received an IS2 score of one, where no record type was specifically declared. Only three de-identified textual medical information records (two produced by HMS Scrubber along with one by MeDS) had their record type removed from each and every report processed on the

5 systems. Also, some records had rightly characterized [79] as a conclusion. Refer to Table. 3.2 for complete summary.

| Software | Interpretability rating (highest rating is 50 for every classification) | | | |
|---|---|---|---|---|
| | IS1 (medical information) | IS2 (record type) | IS3 (conclusion) | % Totals(Rating sum) |
| HMS | 43 | 48 | 50 | 94% (141) |
| MIT | 48 | 50 | 50 | 99% (148) |
| HIDE | 46 | 50 | 50 | 97% (146) |
| MeDS | 39 | 49 | 50 | 92% (138) |
| MIST | 48 | 50 | 50 | 99% (148) |

Table 3.2: Calculations on respective system de-identified report interpretability [79].

#### 3.4.3.2 Assessment on the de-Identified reports‘ document structure changes

In de-identified textual medical information records, it was discovered that majority of the utility tools left almost all of the format structure unchanged as shown in Figure. 3.4. In roughly 5% of de-identified textual medical information records the MeDS utility tool removed some textual medical information record's format structure (e.g. line-breaks).

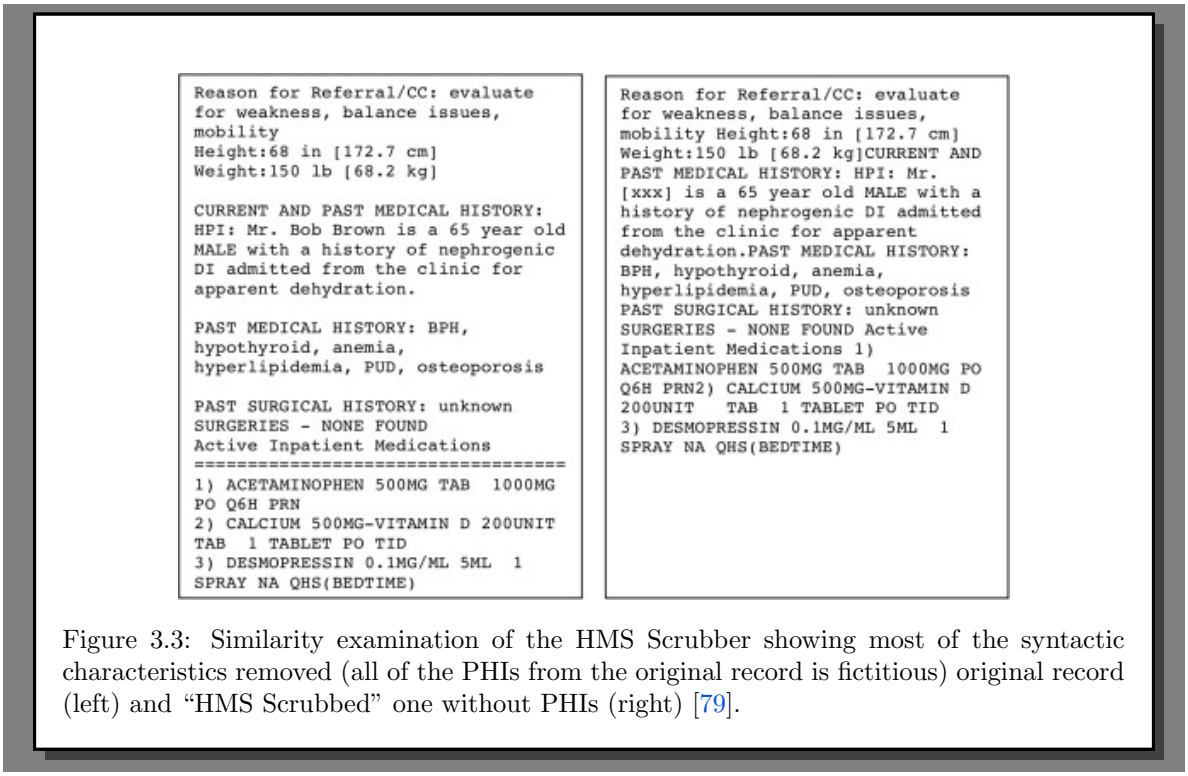



Figure 3.3: Similarity examination of the HMS Scrubber showing most of the syntactic characteristics removed (all of the PHIs from the original record is fictitious) original record (left) and "HMS Scrubbed" one without PHIs (right) [79].

### 3.4.4 Post-processing assessment of overall impact on clinical information after reports de-identification

Stéphane et al. used a null hypothesis that stated there was no concept count proportion difference between the original and a de-identified version of the corpus when examining post-processing assessment of overall impact on clinical information after reports de-identification. SNOMED-CT's abstractions were found in various editions of the textual medical information records to average 456,6 (0.950 CI: 136892.4-137060.7) concepts per document (0.95 CC: 456.31-456.87). These concepts are based on 8794 different concepts of SNOMED-CT. In the de-identified corpus 1117 instances of the most common abstraction were discovered from various de-identified editions of the corpus 926 different abstractions were only discovered.

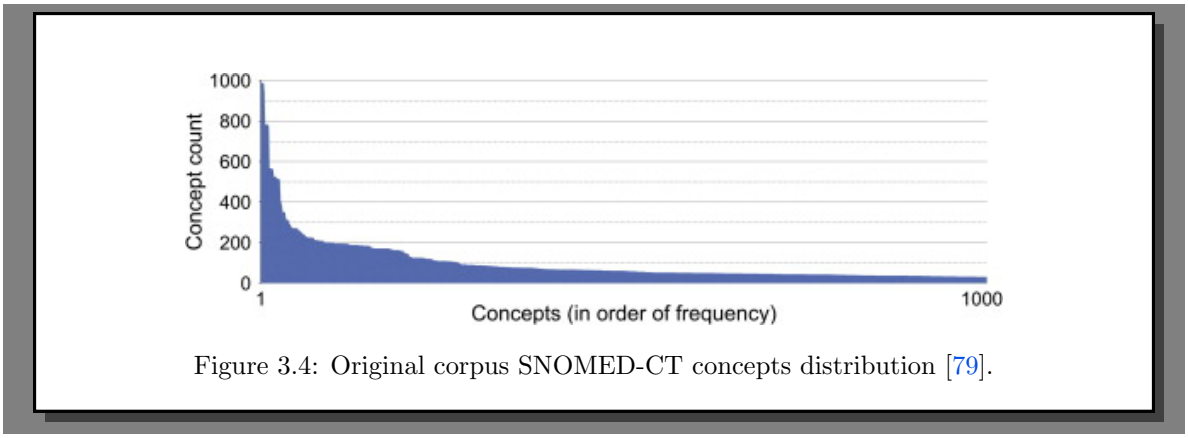



Figure 3.4: Original corpus SNOMED-CT concepts distribution [79].

When most de-identified corpus were compared against the original textual medical information records (Table. 3.3), the likelihood for the t-test led to the rejection of the null hypothesis. Only versions of the PHI classifications (C5 together with C7 from Table. 3.3) de-identified with a single filter from SVM, re-articulated PHIs or labels, did not vary substantially compared to the original textual medical information records [79].

| | **C1** | **C2** | **C3** | **C4** | **C5** | **C6** | **C7** |
|---|---|---|---|---|---|---|---|
| All abstractions counted | 138,137 | 136,990 | 136,473 | 136,574 | 137,482 | 135,883 | 137,297 |
| All abstractions counted disparity from original report (C1) | | -0.83% | -1.20% | -1.13% | -0.47% | -1.63% | -0.61% |
| Mean for the abstractions counted disparity from original report (C1) | | -2.28% | -2.75% | -3.04% | -1.18% | -1.87% | -1.59% |
| Abstractions counted disparity | | -2.28% | -2.75% | -3.04% | -1.18% | -1.87% | -1.59% |
| 95% confidence interval | | -3.36% | -3.81% | -4.11% | -2.28% | -2.79% | -2.66% |
| t-test Probability | | 0.0014 | $\leq 0.001$ | 0.002 | 0.068 | $\leq 0.001$ | 0.022 |
| Log-likelihood Mean | | 0.275 | 0.277 | 0.322 | 0.275 | 0.211 | 0.261 |
| C1 represents Original report (not de-identified); C2 represents De-identified utilizing 4 binary SVM filters and re-articulated PHI; C3 represents De-identified utilizing 4 binary SVM filters and a general 'PHI' label; C4 represents De-identified utilizing 4 binary SVM filters and labels combined with 'PHI' classifications; C5 represents De-identified utilizing 1 multi-class SVM filter combined with re-articulated 'PHI'; C6 represents De-identified utilizing 1 multi-class SVM filter combined with a general 'PHI' label; C7 represents De-identified utilizing 1 multi-class SVM filter combined with labels that included 'PHI' categories. | | | | | | | |

Table 3.3: Corpora' SNOMED-CT concept count differences [79].

The log-likelihood ratio method was originally developed to compare word frequencies between different corpora, and Stéphane et al. adapted it for concept frequencies. The likelihood ratio tests if such proportion differs considerably from one or if its natural logarithm is considerably distinct from zero. If the observational data accept the null hypothesis, then the difference of two probabilities should not exceed the sampling error. Although there were no different versions of the corpus, several concepts had a log-likelihood rate over 6.96, and thus had considerably different numbers from the initial transcription version. Refer to Table 3.4

Most overlaps correspond, as expected, to ambiguities of the category PHI Person Name. In particular, 88% of the total count of precise overlap is for this category and 76% for partial overlap. There were also some clinical data overlap, although less frequently, with other PHI classifications, such as Health Care unit and Country or State. We can also see from Table. 3.6 that 386 PHI annotations are precisely overlapping under Problem, Test, or Treatment annotations —this translated to 0.81% of the clinical information concerned were erroneously declared. This proportion evidently rose slightly to 1.78% when

| SNOMED abstractions (code) | Complemented words (correct match) | Counted abstractions in every record edition | | | | | | | LL Mean-value (and extremes) |
|---|---|---|---|---|---|---|---|---|---|
| | | C1 | C2 | C3 | C4 | C5 | C6 | C7 | |
| Vertebral artery (85234005) | VA (Veterans Admin.) | 112 | 36⋆ | 0⋆ | 0⋆ | 4⋆ | 0⋆ | 0⋆ | 82.91 (40.32-125.50) |
| Carney complex (239132009) | Carney Hospital | 53 | 64 | 61 | 382⋆ | 61 | 59 | 63 | 48.02 (0.43-284.40) |
| Phencyclidine measurement (20857001) | PCP (Primary Care Provider) | 47 | 8⋆ | 6⋆ | 6⋆ | 7⋆ | 7⋆ | 6⋆ | 33.80 (30.30-35.79) |
| Pneumonia, pneumocystis carinii (415125002) | PCP (Primary Care Provider) | 42 | 6⋆ | 5⋆ | 4⋆ | 7⋆ | 8⋆ | 6⋆ | 30.27 (24.79-36.16) |
| Diabetic retinopathy (4855003) | DR (Doctor) | 51 | 24⋆ | 15⋆ | 17⋆ | 26⋆ | 28 | 26⋆ | 11.69 (6.42-20.31) |
| Endoplasmic reticulum (33761008) | ER (Emergency Room) | 20 | 4⋆ | 4⋆ | 4⋆ | 5⋆ | 4⋆ | 5⋆ | 10.82 (9.54-11.51) |
| Mixed antiglobulin reaction test (117360002) | MAR | 3 | 15⋆ | 0 | 0 | 12⋆ | 0 | 0 | 7.33 (5.83-8.83) |

Table 3.4: Significant count differences on SNOMED-CT concepts marked with ⋆ [79].

| i2b2 concepts | Total | Detected as PHI (exact match) | % | Detected as PHI (partial match) | % |
|---|---|---|---|---|---|
| Problem | 19,667 | 65 | 0.33 | 187 | 0.95 |
| Test | 13,833 | 40 | 0.29 | 180 | 1.30 |
| Treatment | 14,185 | 281 | 1.98 | 482 | 3.40 |
| Overall | 47,685 | 386 | 0.81 | 849 | 1.78 |

Table 3.5: Respective PHI annotation overlap in Problem, Tests and Treatment categories as detected by Stéphane et al. [79].

taking into account on the partial overlaps. We can also see that 1.98% of treatment annotations were fully de-identified and 3.4% partially implying that Treatment was the category highly plagued by de-identification errors as compared to Problem and Test.

| Type of PHI | Accurately paired | | | | Partially paired | | | |
|---|---|---|---|---|---|---|---|---|
| | Problem | Test | Treat | Combined | Problem | Test | Treat | Combined |
| Individual's name | 0.31% | 0.25% | 1.73% | 0.72% | 0.82% | 0.74% | 2.70% | 1.36% |
| Street/City | 0 | 0 | 0.01% | 0 | 0.01% | 0.01% | 0.02% | 0.01% |
| State/Country | 0.01% | 0.01% | 0.01% | 0.01% | 0.06% | 0.09% | 0.13% | 0.09% |
| Deployment | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ZIP code | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Healthcare Unit | 0 | 0 | 0.19% | 0.06% | 0 | 0.12% | 0.37% | 0.15% |
| Other Org Name | 0 | 0 | 0.01% | 0 | 0 | 0.07% | 0.11% | 0.05% |
| Date | 0 | 0 | 0 | 0 | 0.02% | 0.14% | 0.01% | 0.05% |
| $Age > 89$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Phone Number | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 3.6: The overlap in testing and treatment rates for each type of PHI observed [79].

### 3.4.5 Assessment of clinical eponyms impact

It was reported that an aggregate of 65 medical annotations were found in 50 VHA sub-corpus documents by human annotators. The number of clinical name or noun formed after a person (include references to diseases, procedure, apparatus or anatomy containing the names of the individuals or places) that each

system recognized as PHI [79] is shown in Table. 3.7. In most systems, Stéphane et al. found that some 10 % of eponyms (e.g., diseases, procedure, apparatus or anatomy containing the names of the individuals or places) had been wrongly classified as PHI, whereas two utilities (MeDS together with HMS Scrubber) had accuracy performance issues and wrongly categorized the PHIs to a much greater extent as compared to the rest [79].

| **Utility used** | **Eponyms categorized as PHI** | |
|---|---|---|
| | Total | Total eponyms % |
| MIT | 8 | 12.31 |
| HIDE | 9 | 13.85 |
| HMS | 26 | 40.00 |
| MeDS | 32 | 49.23 |
| MIST | 7 | 10.77 |

Table 3.7: Clinical eponyms determined as PHIs by respective system [79].

## 3.5 Conclusion

The correlation between PHI and selected textual medical information was also very narrow. Such is the conclusion when examining the details in order to draw some kind of conclusivity over the appraisal or evaluation of the effects of de-identification process on textual medical data records. In the "medications" section of the i2b2 medical information records, a chart of medicines used by the medical services recipient was reported on in Stéphane et al.'s analysis. Some medications were characterized as PHI due to the absence of contexts in these sections [79].

In summary, the findings of Stéphane et al. [79] show that better de-identification technology for ambiguous medical conditions is needed. This challenge still remains unresolved and involves the research into and resolution of semanticized variability and ambiguity across the entire NLP research community [79].

The study by Stéphane et al. demonstrated that even a powerful text de-identification system such as BoB could mistakenly mislabel non-PHI and, consequently, hide or delete clinical information. Even though such classification error was found to be small, but should not be considered insignificant. A detailed review also looked at the effects of textual de-identification in the ensuing computerized removal of medications terminologies and discovered that no substantial impact on such [20], however only a minute fraction of the medical data found in textual medical information, as summarized by Stéphane et al. This result probably was due to the limited "likeness" of "PHI drug names" [79].

# Chapter 4

# NLP with LSTM Approach

## 4.1 Introduction

How does one decide what type of representative is a big question for developers who work in NLP? What other data are relevant? Besides, which information is relevant, how does one determine such? That is the issue within the framework —determining relevance and method to represent data.

Text tagging is the process by which various components of unstructured data have been manually or automatically added to tags or annotations in the process of preparing those data for analysis. Tagging takes place at a level that is more granular than categorizing, and can offer further insight benefits. "Named Entity Extraction" (NER) is a common form of markup [33]. This method allows the identification of names of persons, products, organizations, locations, or dates by a batch of unstructured data. Such an approach could be helpful in determining interactive relationships and patterns between the named entities.

The problem I faced here in medical notes de-identification task can be approached as a special case of NER task —Hence, one can approach the solution as having access to labeled data on the domain of origin, but unlabeled data on the domain of destination(medical notes with PHI data). Exploiting useful information from the medical notes with PHI data can be the key to improving de-identification notes necessary for tagging and for de-identification in particular. Towards this end, I adopted the idea of learning representations which has been demonstrated useful in capturing hidden regularities underlying the raw input data.

### 4.1.1 Word Embedding and NLP

Natural language processing (NLP), through computation-based algorithms or programs, is designed to analyze, methodologically assess, evaluate, and represent the human's syntactic dialectical language in an automatic way. A wide range of applications has been enabled by NLP system, e.g. Amazon Alexa. NLP can also be used to teach machines to perform complex tasks related to the natural language, such as machine syntactic dialectical language translation and dialogue generation. The idea of *Word representational embedding* was originally introduced by Bengio, et al. [6,7] more than a decade ago.

Word embeddings are essentially distributional vectors based on the so-called distributional hypothesis, meaning similar to words appearing in a similar syntactic context. Word embedding is pre-trained for a task in which the purpose of predicting or forecast a word using a shallow neural algorithmic network usually is its context. A **corpus** is typically a collection of text documents usually belonging to one or more subjects. An embedding $\kappa : words \rightarrow \mathbb{R}^n$ is a parametrical mathematical expressions that

maps phrase segments for textual information into high-dimensional vectors of 200 to 500 dimensional components). Typically, the function is parametric database matrix, $\Theta$, built word per row: $\kappa_\Theta(k_n) = \Theta_n$. Each word in $\kappa$ is random vector initialized. Success in task execution require appropriate scalar learning. It is useful to consider morphological information within a word for textual analysis tasks such as POS, and recognition of named entity also known as NER —and this is known as word embedding. Before one can fully utilize the corpus to perform a specific task in NLP, some text pre-processing may be important. However, in this task I want to fully utilize the corpus to perform the de-identification task without having to perform tedious text pre-processing steps.

### 4.1.2 Text pre-processing

Some of the most important ways of cleaning and pre-processing textual data that are used heavily in Natural Language Processing (NLP) pipelines are [88]:

- Removing tags: Textual material often contains unnecessary material content like HTML tags and or labels, which do not add much or significant and considerable value when analyzing or evaluating text. e.g. Removing the <br> from the text → *Hey John, <br> Please send the files soon.*
- Removing accented characters: In any text corpus, accented characters/letters need to be converted, transformed and standardized into ASCII components or characters. e.g. removing accent from a name as in; *Özlem* → *Ozlem*.
- Removing special characters: Simple regular expressions (regexes) can be used to remove and/or purge special characters and symbols which are usually non alphanumeric characters often add to the extra unwanted noise in unstructured textual material or content. e.g. removing apostrophe and segmentation as in; → *she's*.
- Removing stopwords: These are words that have little, insignificant or no meaning especially during the process of building meaningful textual material or content characteristics are called stopwords or stop-words. Usually these words have the highest presence frequency if you use a single term or word frequency in a body of the textual material. Words such as "a", "an", "the", etc. are thought to be stop-words. No universal stopword list is provided but one could utilize nltk's standard English stopwords lookup list. e.g. ~~*The*~~ *quick brown fox jumps over* ~~*the*~~ *lazy dog*.
- Expanding contractions: All shortened versions of words or syllables need converting to its expanded original form to help with text standardization. e.g. *can't* → *can not*.
- Stemming and lemmatization: Word trunks generally form the basis of probable words that can be generated by adding prefixes and prefixes in the trunk to create new words. It's called inflection. The reverse process is known as stemming to obtain the base form of a word. Lemmatization is very much like stemming, where we remove and omit a word attachments to the basic structural form of a word. In this case, however, the base structural form is known as the syntactic root word but not the root stem. The distinction is that the root word is always a correct lexicographic word, but the root stem is not. e.g. *better* → *good*, *read* → *reading*.

### 4.1.3 A general assessment of Machine Learning in NLP

The Convolutional Neural Network(CNN) is an algorithm that can take the image as an input, give significance to different facets / objects in the picture and distinguish between them (learn-able weights and biases). In order to model sentences with a basic algorithm in CNN, sentences are first tokenized into words that are further converted and transformed into a word embedding matrix of d-dimensions (e.g. the input embedding-layer). This input embedding layer contains convolutional filters that are

applied to produce what is known as a feature map by using a filter of any window size. Followed by a max-pooling procedural operation. A max-pooling is a maxing operation that is applied to each filter to produce a fixed length resulting output and to reduce the size of the output. Consequently, it gives the final computed representation of the sentence. An inherent deficiency with basic CNNs is the inability, which could be critical for different NLP tasks, to model long-distance dependencies [129]. CNNs can be linked to the time-delayed neural networks (TDNN) which allow a wider contextual window/range during training to address this deficiency. CNNs are generally effective because in contextual windows they can mine semantic clues, but they struggle to preserve long-distance context information and model sequence. Recurrent models are better suited, appropriate , and ideal for such type of learning.

#### 4.1.3.1 Recurrent Neural Network (RNN)

Traditional RNN had issues with the eruption or disappearance of gradients, in which gradient vectors can increase drastically or decline as this propagates to earlier measures. This challenge makes it hard to train RNN in sequence in order to generate long distance dependence. [7, 51] However, Modern RNN models are appropriate in inputs of arbitrary length to model context dependence. For this type of learning, recurrent models are best suited to produce a correct input composition because they are capable of storing the results of past computations and use such data in the existing calculation.

When denoting the values taken by categorical (discrete) features, the input expected from an RNN is usually one-hot encoding or word representation embeddings [1]. However, in some particular cases linked to the abstract representations built up in a CNN model. Other models were later introduced to address the vanishing gradient issue that makes it hard for RNNs to learn and fine-tune previous layers' parameters. Long Short-Term Memory (LSTM), residual networks (ResNets), together with gated-recurring networks (GRU) were later presented and introduced to tackle the slope disappearance or exponential gradient diminishing problem [52] which makes the situation difficult for RNNs to learn and fine-tune the settings parameters.

Attention Mechanism is a technique that inspires the decoders to use the last hidden condition state as well as the information (contextual vector) determined using the input hidden condition state sequence of the above-mentioned RNN-based Framework. This is especially advantageous for tasks which require alignment of the input and output text.

#### 4.1.3.2 Reinforcement Learning

Reinforcement Learning includes machine computational learning practices, training agents or modules to conduct discrete rewarding actions [119]. In several fields of natural language generation (NLG), Reinforcement Learning has gained momentum, especially in areas such as summarization of text [119]. The Adversarial training concept, which seeks to fool and circumvent a discriminator who is able to differentiate generated synthesized sequences from true sequences, is also employed in training language generators.

#### 4.1.3.3 Unsupervised Learning

The unsupervised mapping of phrases to vectors of fixed size is known as Unsupervised phrase representation learning [85,96]. The distributed representations capture semantical, syntactic, and lexical language properties and are trained by an auxiliary task. The forecast task aims to obtain the next adjoining sentence on the basis of a center sentence was proposed as a Skip-thought model (algorithms similar to the one for learning word embeddings) [85]. In this model, the decoder generates objective sequential outputs and the encoder is viewed as a generic feature characteristic extractor, including word

[1] https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing-categorical-features

insertion, and is then trained using the seq2seq frame [96]. The model aims to learn distributed embeds for the input sentences, which is similar to how word embedding in previous language modeling methods was learnt for each word [96].

#### 4.1.3.4 Deep Generative Models

The use of deep generative models, examples of which that come to mind being the Variational Autoenconders (VAEs) together with the Generative Adversarial Networks(GANs), is a recent application for discovering a wealth of structural application in natural words through the process of creating realistic synthesized sentences from a latent code area of NLP [78].

#### 4.1.3.5 Memory-Augmented Network

During the token generation phase, the hidden state vector arrays accessed using the attention mechanism represent the internal memory of the model. The neural networks are linked to some kind of memory in the Network to provide solutions to tasks or creative chores such as visual question answering(QA), language modeling, POS tagging, and sentiment analysis [62], which can provide the model with supporting facts or knowledge of the common sense as a form of remembrance. Dynamic memory networks utilize the neural networks structure for attention, input portrayal representation or depiction, and for answering mechanisms. Also, Dynamic memory networks are deemed superior to the previous memory-based models [62].

## 4.2 Recurrent Neural Networks and LSTM

Feed forward neural networks (FFNNs) assume that the observation at time $t$ and the output at time $t' \neq t$ are independent. Since RNNs do not make this assumption, they can learn to extract features depending on previous observations. This is useful for many tasks where there is a dependency on previous inputs, such as language modeling [60, 84], acoustic modeling [43, 105], and machine translation [117].

#### 4.2.0.1 Back Propagation Through Time

In order to be applied to a RNN, Back propagation needs to be modified to incorporate the time recursion. This form of back propagation is known as back propagation through time (BPTT) [87].

#### 4.2.0.2 Vanishing and Exploding Gradient

RNNs are set back by the inherent problem of the vanishing and exponentially increasing slope or gradient [7, 95], where the gradient tend to either vanish to zero or to increase exponentially during BPTT. One of the advantages of long-short term memory (LSTM) layers is that they are more resilient to the rapidly diminishing gradient problem [42]. Assuming a simple layer, comprised of one unit, with linear activation, the output of this unit, at time $t$ is;

$$y_t = x_t W + y_{t-1} R,$$

The contribution of the output $y_{t-d}$ to the output $y_t$ is multiplied by an exponential function of the recurrent weight. The gradient of $y_t$ is back propagated to $y_{t-d}$ by;

$$\frac{\partial y_t}{\partial y_{t-d}} = R^d$$

where; $R^d$ is the gradient.

As seen on Figure. 4.1, when more layers are introduced to neural nets using activation features, the loss function slope goes to zero, making it very difficult to train the network. Such activation features like the sgmoid function squash a big input room between 0 and 1.

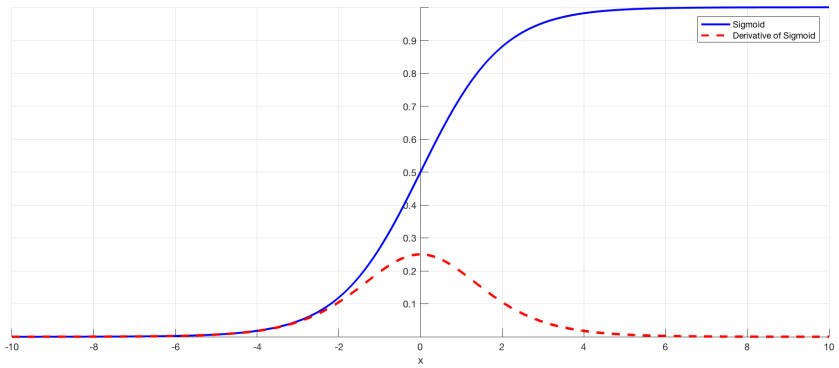



Figure 4.1: The sigmoid function makes a large input space into a small input space between 0 and 1 [41].

A big shift in the input of the sigmoid component thus results in a minor change in the results. In Figure. 4.1, is the sigmoid parameter and its derivative. The derivative is becoming minimal. The easiest alternative is to use other activation features such as ReLU that do not generate a tiny derivative, if the inputs of the sigmoid variable are bigger or lower (as becomes larger). The problem can also be resolved by batch normalization layers.

In the proximity of highly steep cliffs as seen in Figure. 4.2, slope slicing can create gradient descent that is fairly effective. All such steep cliffs usually arise in recurring networks close to a linear approximation. As the weight matrix is multiplied by itself once for each step this causes exponentially steep cliff after such steps. As depicted on the left of the figure, the slope descent of the cliff without the clipping of the slope leads to overflows into the ravine, thereafter, from the face of the cliff, goes a very big abrupt gradient. Disastrously, the big slope propels the variables out of the axis of the graph. While on the right, the descent of gradients with gradient cuttings reacts to the cliff more moderately. Whilst the cliff is climbed, the steps are limited so that the steep area near the solution cannot be carried away from the solution.

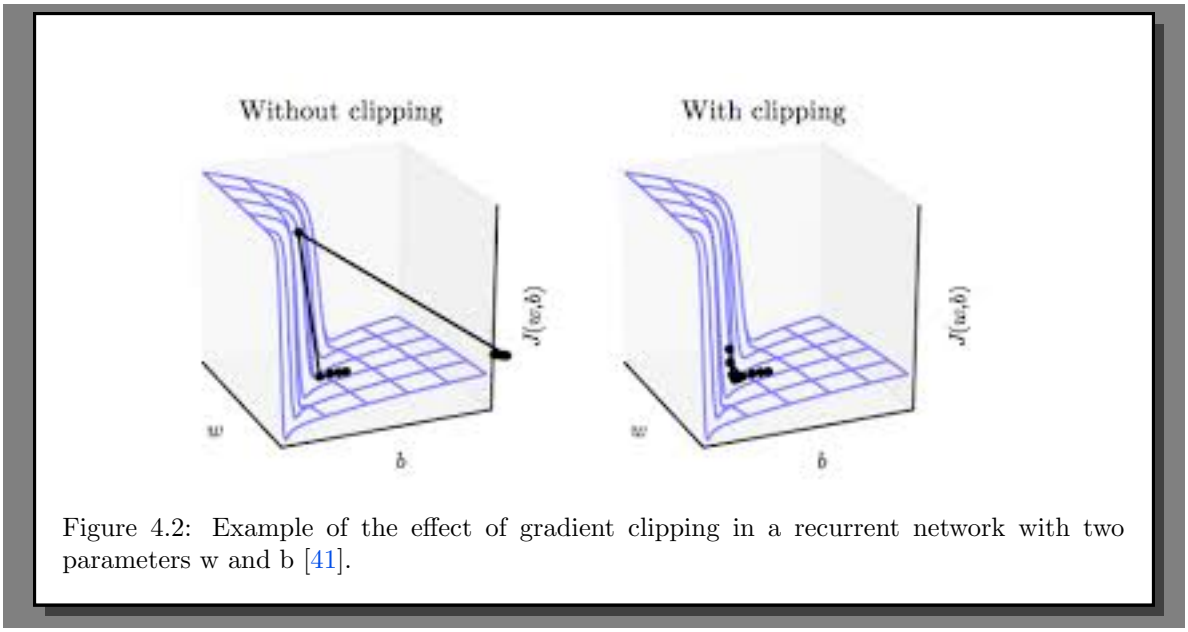



Figure 4.2: Example of the effect of gradient clipping in a recurrent network with two parameters w and b [41].

This evidently shows the fundamental nature of the problem. The repeated multiplication by the recurrent weights will tend to drive the gradient towards either zero or infinite values. The exact threshold depends on the activation function and the overall weight matrix [95].

#### 4.2.0.3 Continuous Word Representation

In order for the neural networks to learn useful features and perform feature analysis, their input must be in a vectorial form. The element whose index is the index of the represented word in the lookup list vocabulary will be non-zero and zero otherwise. In such a setup the most basic representation of words is a very large, very sparse vector since similar words do not have similar vectors and the neural network(NN) cannot generalize to words that were not seen during training. When we create an alternative representation where similar words have similar vectors then the very large, very sparse one-hot vector is replaced by a continuous vector of much smaller dimensionality.

#### 4.2.0.4 LSTM and Bi-directional LSTM(Bi-LSTM)

The core elements of the ANN model are recurrent neural networks (RNNs). LSTM stacks address some issues with RNNs. In particular, the recursion is now controlled by the input. There is also an added internal memory, with an associated recurrent connection, that can preserve observations independently of the output. Access to this internal memory is controlled by gates. The gates protect the internal memory, which reduces the impact of the vanishing gradient problem. As depicted in Figure.4.3, LSTM for text in NLP is composed of and primarily made up by three(3) stacks [22]:

- Character-enhanced token embeds stack;
- Label forecast stack;
- Label pattern optimization stack.

Each token is mapped into a vector representation with a character enhanced token embedded [22]. The vector representation sequence relating to the pattern of the tokens is entered into the tag/label forecast stack, which produces the sequential pattern of scalars with the corresponding likelihood of each token. Finally, the sequence/pattern optimization layer provides the optimal and most probable sequence of forecast labels on the basis of the preceding layer sequence of probability vectors. All layers are learned jointly.

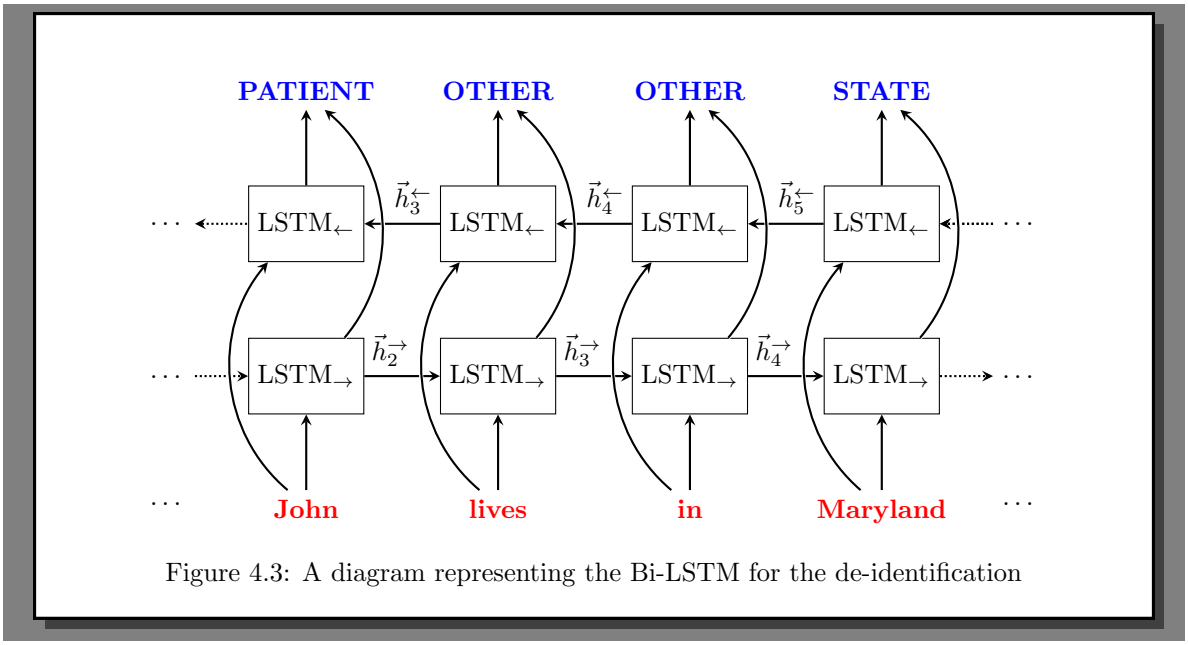



Figure 4.3: A diagram representing the Bi-LSTM for the de-identification

#### 4.2.0.5 Implementation of Bi-directional LSTM_CRF

Bidirectional LSTM_CRF (BiLSTM_CRF) for text-tagging in NLP is a deep learning method or mechanism for sequential pattern labeling/tagging problem [64]. Bi-directional LSTM (BLSTM) layers are the concatenation of two sub-layers. The first sub-layer remember previous inputs. The second sub-layer reverses the recursion and remembers future inputs. This anti-causal sub-layer can be obtained by reversing the recursion used for the causal LSTM layer. The sub-layers cannot be connected to each other, as this would create cycles in the computation graphs. The main advantage of this approach is that the entire layer has seen the entire input at all t. This can be used by the subsequent layers to create features based on the entire input, rather than being limited to a sliding window, or to the current and previous inputs.

For this research the implementation is done as shown in Figure.4.4 and composed of and primarily made up by three layers below [79];

1. An input layer that produces/generates vectoral sequential pattern representation for each and every word, phrase, or smallest sentence components —this comprises two sections: the character-enhanced and token-level representation. This study employed the word representation at token level as implemented through PyTorch engine. A bidirectional LSTM as illustrated in the Figure. 4.3 was able to learn directly from the sequence of each word's character. This could collect information for each word (i.e., John). The last two vectors of the backwards and forward LSTMs are linked in this two-way LSTM to form the final representation the sentence segment (John).

2. LSTM layer, which encompasses the forward-LSTM together with the backward-LSTM, vectoral sequential pattern representation for each and every word, phrase, or smallest sentence components as an input and produces a new forecast word sequential pattern, which depicts the contextual lexical or syntactic information;

3. The CRF layer encapsulates the structural dependencies among subsequent labels by maintaining a matrix of label transition for the CRF algorithm and speculates the best sequential pattern of the tag with the right structures. For NER, the BI-LSTM architecture is the same as suggested by Lample et al.'s architecture for NER [64] and it is the same method as described by Zengjian Liu [67] et al.

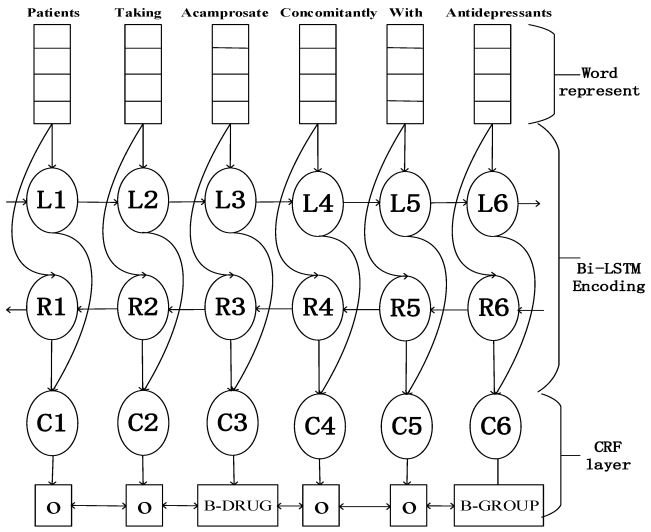



Figure 4.4: The diagrammatic portrayal of a dual unidirectional LSTM with CRF layer, word- and character-level scalar representation implementation [131].

#### 4.2.0.6 LSTM layer

Given a phrase $s = \kappa 1 \kappa_2, ..., \kappa_n$ with phrase segment $\kappa_t (1 \leq t \leq n)$ represented by $x_t$, the LSTM stack accepts word-embed pattern $x = x_1 x_2, ..., x_n$ as input and outputs a different word depiction pattern $h = h_1 h_2 ... h_n$. Let us assume that $h_{ft}$ and $h_{bt}$ denote the hidden state of the forward and backward LSTM respectively at time $t$, whereby $h_t = [h_{ft}^T, h_{bt}^T]^T (1 \leq t \leq n)$ is a transposition of forward LSTM $h_{ft}$ components and backward LSTM $h_{bt}$ products at step $t$. In particular, an LSTM module (with an input door, a forget door, an output door, and a memory sub-component) at step $t$ that accepts phrase segment $x_t$, hidden state $h_{t-1}$ from time $t-1$ and a cell memory $c_{t-1}$ from time $t-1$ as its input then generates $h_t$ and new cell memory, $c_t$ and via the formula below:

$$\begin{cases} i_t & = \sigma \left( W_i \cdot [x_t, h_{t-1}] + b_i \right) \\ f_t & = \sigma \left( W_f \cdot [x_t, h_{t-1}] + b_{(f)} \right) \\ o_t & = \sigma \left( W_o \cdot [x_t, h_{t-1}] + b_o \right) \\ g_t & = \tanh \left( W_g \cdot [x_t, h_{t-1}] + b_g \right) \\ c_t & = f_t \odot c_{t-1} + i_t \odot \tanh \left( W_g \cdot [x_{(tbegin)}, h_{t-1}] + b_g \right) \\ h_t & = o_t \odot \tanh (c_t) \end{cases}$$

where; $W$ are the weights and $b$ are the biases with subnote $f, i, o$ denoting forget, input, and output,

$f_t$ is the forget gate's activation vector,

$i_t$ is the input vector to the LSTM unit,

$o^t$ is the output gate's activation vector of the LSTM block,

$\sigma$ denotes the sigmoid function,

$g_t$ is the cell's new candidate value to be added to the cell memory,

$c_t$ is the new cell memory, and

$h_t$ is the hidden state vector or cell output given that $W_j, U_j, b_j$ for $j \in (i, f, o, g)$

Note that $\sigma$ denotes the sigmoid mathematical expressions function and tanh is the hyperbolic tangent, $x_t$ is the input vector, $h_t$ is the output vector, $c_t$ is a cell condition representation or state vector. In the mathematical expression, $W$ are the weights and $b$ are the biases. $f_t$, $i_t$, and $o_t$ are called the forget,

input, and output gates of the LSTM block. At $t = 1$ , $h_0$ and $c_0$ are activated to zero scalars, given that $W_j, U_j, b_j$ for $j \in (i, f, o, g)$ respectively.

The subscripts $_t$ denote the index of the current time step. Using the above equations, one can compute the final memory cell and hidden state respectively:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$
$$h_t = o_t \odot \tanh(c_t)$$

whereby, $\odot$ is the vectors‘ element by element multiplication operator which represents the Hadamard Shur product.

The new memory $c_t$ and unobservable state $h_t$ are passed along for use in computing the subsequent time step $(t + 1)$ values, and the unobservable state $h_t$ is passed forward to serve as input/entry to the next arranged stack or sub-stack level/layer in the current time step $t$ (*i.e.* the next LSTM layer or the output layer). Thus, there are eight learned matrices in each LSTM layer: four each of the $W_{(\cdot)}$ and $U_{(\cdot)}$ matrices in Equations$(i_{(t)})$ and $(c^{(t)})$ respectively.

Memory cells in the LSTM are time-dependent additive, hence solves the gradiential vanishing issue or disappearing slope dilemma. This is the most important distinction of LSTM in the sense that it allows for an uninterrupted gradient flow through the memory cell. Gradient exponentially increasing or exploding is still regarded as an issue, though in practice simple gradient clipping works well. LSTMs have outperformed vanilla RNNs on many tasks, including on language modeling [116]

#### 4.2.0.7 CRF layer

Conditional random fields (CRFs) are a class of graphical models [63, 118]. CRFs are an extension of Maximum Entropy Markov Models(MEMMs), intended to address the label bias problem where the model is biased towards states with few outgoing transitions. CRFs are used to model a conditional distribution $p(y|h)$ over an input sequence h and an output sequence y. CRFs are defined by a factor graph. The distribution $p(y|h)$ is factorized according to this graph. The factor functions found in the factor graphs are similar to distributions over features extracted from the sequences y and h. CRFs replace the global distribution over a large number of variables by many factor functions over a smaller number of variables. CRFs are commonly used for chunking and NER, as well as for other sequence labeling task. CRF have been regularly used for information extraction since their introduction.

The CRF layer takes sequence $h = h_1 h_2, ..., h_n$ as entry input, then produces efficiently probable tag/label sequential pattern $y = y_1 y_2, ..., y_n$. With training set $\mathcal{X}$ provided, all parameter components of CRF (expressively indicated as $\theta$) are computed through mathematical estimation by maximum optimization the mathematical expressions for the log-likelihood here below:

$$L(\theta) = \sum_{(s,y)\in D} \log p(y|h, \theta)$$

where y is the appropriate tag pattern of statement/phrase segment s, h is the word pattern depiction for the sentence s as produced by the network and p, is the occurrence-dependent likelihood of y given sentence s and $\theta$. The rank is assumed to be $z_{\theta^{(h,y)}}$ for the tag pattern depiction $y$ for sentence $h$, the

occurrence-dependent probability/likelihood p computed using:

$$p(y|h,\theta) \quad = \quad \frac{e^{Z_\theta(h,y)}}{\sum_{y'} e^{Z_\theta(h,y')}}$$

where y' is a highly probable tag sequential pattern of h. The log-likelihood of p is:

$$\log p(y|h,\theta) \quad = \quad Z_\theta(h,y) - \log \sum_{y'} e^{Z_\theta(h,y')}$$

A transfer matrix T with emission matrix E is used to articulate and represent the pattern frameworks between neighboring tags/labels to as $Z_\theta(h,y)$ follow:

$$Z_\theta(h,y) = \sum_{t=1}^{n} (E_{y_t,t} + T_{y_{t-1},y_t})$$

where, $E_{y_t,t}$ is the likelihood that token $h_t$ with tag $y_t$, and is the probability/likelihood that $h_{t-1}$ with corresponding tag $y_{t-1}$ preceded by $h_t$ with tag $y_t$. Thus, maximization of the log-likelihood equation, $L(\theta)$ is obtained by maximum optimization of $\sum_{(s,y)\in D} \log p(y|h,\theta)$ , iteratively going through all training D set computationally. Whence, obtain the best tag pattern for every sentence/ paragraph segments, using the optimal total score of $Z_\theta(h,y)$ which is obtained by maximum optimization of $\sum_{t=1}^{n}(E_{y_t,t} + T_{y_{t-1},y_t})$ and verify output using Viterbi mathematical expression algorithm at test phase.

## 4.3 Conclusion

In the data science field, difficulties with sequence prediction are regarded one of the most challenging challenges to overcome. These include a range of problems, from sales prediction to patterns in the stock market data, from film plots and language translations to the prediction of your new words on your smartphone's keyboard, through the recognition of your way of speaking.

In theory, recurrent networks will store past inputs to generate the desired output. Recurrent networks in the estimation of time series and process control are used because of this property. Time dependencies covering several time steps , for example, between the respective inputs and desired outputs, require practical applications. Nonetheless, it takes too long for gradient-based approaches to understand. In training recurrent neural networks, practical difficulties have been noted to accomplish tasks where temporal contingencies in the input / output sequences span long periods. The exceptionally high learning time comes because the error disappears as it gets propagated back.

This section addressed why gradient-based learning algorithms face a daunting challenge as their duration increases. Such findings show a balance between effective incremental descent learning and long time knowledge latching. As of of the latest advancements in data science, Long Term Memory Networks (LSTMs) are already established as the most successful solution for virtually innumerable such sequence predictive problems. LSTMs in many ways have an advantage over traditional neural and RNN feed-forwards networks. It is due to its property of recalling patterns systematically for long periods. When LSTM is used in conjunction with CRF the benefits are immense. For prediction tasks where contextual knowledge or neighborly status affects the current prediction, I have used CRF as a probabilistic system for the labeling and segmentation of sequential data. As previously defined, CRF is an undirected graphic design which represents a single log-linear label sequence distribution given the specific observation series. The major benefit of the CRFs over the hidden Markov models is their conditional existence and therefore the freedom of HMMs is relaxed to achieve a tenable deduction [63, 118].

De-identification of medical notes has crucial role to play in unearthing the wealth of clinical information in order to make an enormous leap forward in the areas of ML-aided disease detection, ML-driven triage and avoidance of disorders, in the clustering of illness sub-types, in defect identification, and in an ML-augmented physician deployment. In Chapter 5, I will cover Feature Representation and how those will act as a building block to help the implementation of the De-identification of medical notes using LSTM.

# Chapter 5

# Feature Representation

## 5.1 Introduction

Features proposed and implemented in information retrieval [72], document classification [107], question answering [122], named entity recognition [125], and parsing with compositional vector grammars [108] all have one thing in common—they are real-valued vector representing each word and are the building block of the Semantic vector space models of language.

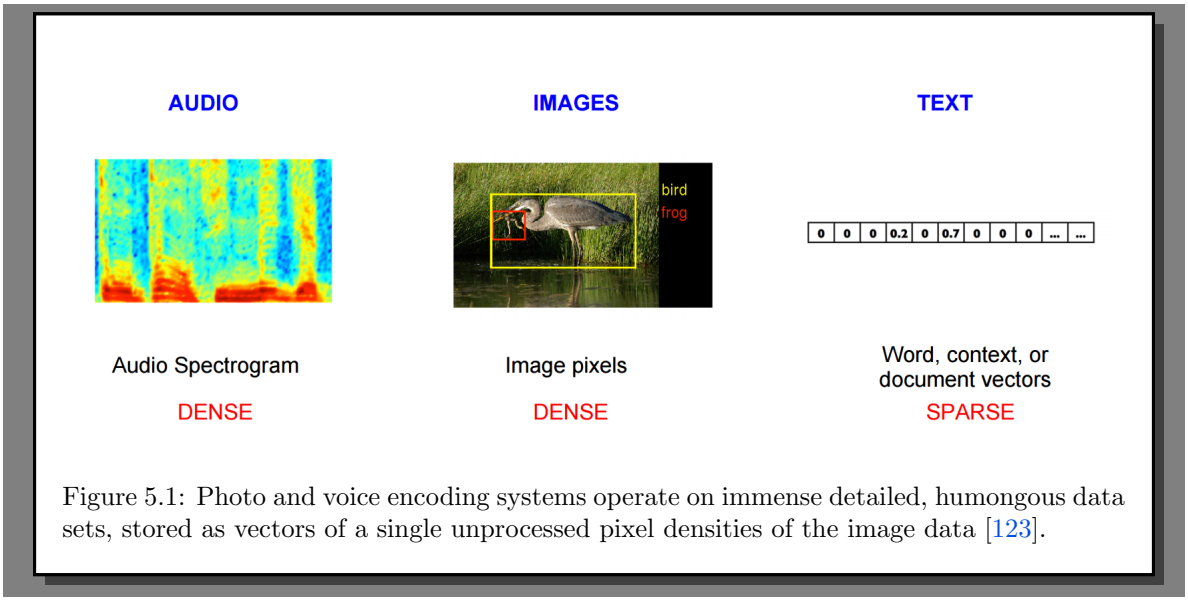



Figure 5.1: Photo and voice encoding systems operate on immense detailed, humongous data sets, stored as vectors of a single unprocessed pixel densities of the image data [123].

Word representations are methods of the word vector that are defined by various ways such as the distance or angle of vector pairs between words. Mikolov et al. introduced the new system of measurement, based on word analogies, which examines not the vector distance but the different dimensions of difference, as to the finer structure of a word vector area [86]. These can generalized into 2 major model categories for computationally learning lexical composition of the smallest parts of a sentence and the associated vectors as named here below:

- Matrix factorization methods: These use low grade estimates to break down vast matrices which collect statistical data regarding a aggregate information records. Matrix factorization method varies according to application the specific type of information obtained from such matrices. Although methods such as latent semantic analysis (LSA) [18] efficiently use statistical information, the word 'analogy' task shows a vector space structure that is relatively poor. In LSA, matrices are of type 'term-document', i.e. rows match words or terms, and the columns are the same as various corpus documents.

- Shallow Window-Based Methods (local context window methods): These methods learn word representations that help to make local context predictions. Procedures such as Mikolov et al's

skip-gram model [86] might perform better with an analogy, but they are poorly when comes to using corpus statistics, as instead of using global co-occurrence numbers, they train in separate local context windows. In a single neural network architecture, Bengio et al. introduced a model to learn word vector representations as a segment of a non-complex neural network architecture for language enhancement configuration [6]. Instead of the above scenario as with language component models Weston and Collobert [14] separated word vector training from the subsequent training goals, thus opening up [15] using the complete contextual word for learning representations, rather instead of using only previous state in contextual model within languages.

The inherent deficiencies of the two methods above led to emerging on newer approach like `Word2vec` and the GloVe methods [85].

### 5.1.1 `Word2vec`

`Word2vec` is a methodology used by a squad of Google data scientists led by Tomas Mikolov in the calculation of vector depictions of words [85]. Google is hosting a `Word2vec` open-source application programming interface (API) code published in Apache 2.0. license [1]. `Word2vec` takes a significant body of a textual message as its input and creates a scalar space, generally several hundred parameters, with the correlating parameter in space being allocated to each distinctive word within the body. Word vectors are placed in the space of the vector, so that phrases that share similar contexts are placed near one another in space. [83].

To help discern patterns no wonder `Word2vec` has found usage in many areas ranging from genes, code, likes, playlists, social media graphs, etc. `Word2vec` may make incredibly precise predictions when provided sufficient data, usage and scenarios about a word's significance given past instances .`Word2vec` achieves this by measuring cosine similarity; when no similitude as an angle of ninety, and when the total overlapping overall resemblance is zero degrees. In bio-informatics software solutions Asgari and Mofrad have suggested extensions of n-grams word matrices in genetic sequences (e.g. DNA, RNA, and Proteins)—named as BioVec [2]. A comparable version, dna2vec, showed that the the cosine similarity of dna2vec and Needleman-Wunsch similarity score are related [91].

`Word2vec` encodes each word into a scalar and train phrases within the feedback textual information against other phrase parts by either using contextual prediction target word (through non-discrete bag of words (CBOW) method) or by using a phrase to anticipate a target context(also known as skip-gram method).

### 5.1.2 The GloVe method

The basis of GloVe method suggest that word vector training should be related to the probability of co-occurrence of specific words rather than the respective raw probabilities. Google is hosting a GloVe open-source application programming interface (API) code published in Apache 2.0. license [2]. Consider two words $\kappa_i$ and $\kappa_j$ whereby the connection of these phrases could be analyzed by learning the ratio of their co-occurrence likelihood with respect to various test words,$\kappa_k$ [96]. The Glove method present the case by letting the word-to-word co-occurrence matrices be marked with X; whose inputs $X_{ij}$ and tabulation word $\kappa_j$ occurrence given word $\kappa_i$. Therefore, one can denote, in the context of the word $\kappa_i$, as $X_i = \sum_k X_{ij}$ is the number of times any word will appear. By having, $P_{ij} = P(j|i) = \frac{X_{ij}}{X_i}$ as the probability that word $\kappa_j$ appear in the context of word $\kappa_i$. The ratio $\frac{P_{ik}}{P_{jk}}$ depends on three words $\kappa_i$,$\kappa_j$,

[1] `Word2vec` is unsupervised learning algorithm for obtaining vector representations for words. For additional information, please visit; https://code.google.com/archive/p/word2vec/

[2] For additional information, please visit;https://nlp.stanford.edu/projects/glove/

and $\kappa_k$, the most general model takes the form, $F(\kappa_i, \kappa_j, \tilde{\kappa}_k) = \frac{P_{ik}}{P_{jk}}$, where $\kappa \in \mathbb{R}^d$ are word vectors and $\tilde{\kappa} \in \mathbb{R}^d$ are separate context word vectors.

Since vector environments are intrinsically sequential systems, the information logical encoding method present the ratio $\frac{P_{ik}}{P_{jk}}$ in the word scalar space with the scalar differences. Hence, $F(\kappa_i - \kappa_j, \tilde{\kappa}_k) = \frac{P_{ik}}{P_{jk}}$. Take the arguments‘ directional multiplication to prevent obfuscation of the linear structure of F to obtain, $F\Big((\kappa_i - \kappa_j)^T \tilde{\kappa}_k)\Big) = \frac{P_{ik}}{P_{jk}}$. For word-to-word co-occurrence matrices, the difference among a word and a context word is arbitrary and that we are free to exchange the two roles. For consistency, not only exchange $\kappa \leftrightarrow \tilde{\kappa}$ but also $X \leftrightarrow X^T$. With F structure-preserving map among the subgroups $(\mathbb{R}, +)$ and$(\mathbb{R}_{>0}, \times)$ [96];

$$F\Big((\kappa_i - \kappa_j)^T \tilde{\kappa}_k)\Big) = \frac{F(\kappa_i^T \tilde{\kappa}_k)}{F(\kappa_j^T \tilde{\kappa}_k)}$$

whereby; $F(\kappa_i^T \tilde{\kappa}_k) = P_{ik} = \frac{X_{ik}}{X_i}$. With F=exp, then; $\kappa_i^T \tilde{\kappa}_k = log(P_{ik}) = log(X_{ik}) - log(X_i)$; however, the term $log(X_i)$ is $k$ independent so it can be incorporated into the bias term $b_i$ for $w_i$ to obtain [96];

$$\kappa_i^T \tilde{\kappa}_k + b_i + \tilde{b}_k = log(X_{ik}) - log(X_i)$$

Adding a new weighted least squares regression model to address the problems of logarithm diverges whenever its argument is zero and the problem of equal consideration for all events' likelihood of mutual occurrence even those occurring rarely or never to obtain;

$$J = \sum_{i,j=1}^{V} f\Big(X_{ij}\Big)\Big(\kappa_i^T \tilde{\kappa}_j + b_i + \tilde{b}_j - log(X_{ij})\Big)^2$$

where V is the size of the vocabulary. The weighting function should obey the following properties [96]:

1. $f(0) = 0$. If f is viewed as a non-discrete function, it will approach zero as $x \to 0$ rapidly as the $\lim_{x \to\ 0} f(x) log^2 x$ is finite.
2. f(x) must be non-decreasing so as to avoid over-weighting rare co-occurrences.
3. f(x) should be relatively small in the case of large values of x, to avoid over-weighting frequent co-occurrences.

Therefore, the ratio of the probability of co-occurrence with various sample words $\kappa_i$ and $\kappa_j$ must be examined in the context of with various probe words, $\kappa_k$ [96], so to speak.

### 5.1.3 The PyTorch text method

PyTorch is written in C++ and CUDA [3], a C++-like language from NVIDIA that can be compiled to run with massive parallelism on GPUs. PyTorch is a library that provides multidimensional arrays, or tensors. The implementation of PyTorch text (also referred as TorchText) follows the distributional hypothesis. The distributional hypothesis in linguistics is derived from the semantic theory of language usage, i.e. words that are used and occur in the same contexts tend to purport similar meanings [48]. Statistical semantics apply analytical techniques, ideally through unsupervised learning, to scientific analyze the question of identifying the meaning of words or sentences to the degree necessary for information retrieval.

[3] www.geforce.com/hardware/technology/cuda

Suppose given the following phrases:

- The astronaut went to the moon.
- The cosmonaut went to the moon.
- The astronaut landed on planet Mars.

Encode semantic similarity in words by:

- If each attribute is a dimension, then we might give each word a vector, like this:

$$q_{\text{astronaut}} = \begin{bmatrix} \overbrace{5.1}^{\text{can fly}}, & \overbrace{7.7}^{\text{likes to travel}}, & \overbrace{-4.9}^{\text{majored in Astrophysics}}, & \dots \end{bmatrix}$$

$$q_{\text{cosmonaut}} = \begin{bmatrix} \overbrace{5.4}^{\text{can fly}}, & \overbrace{7.2}^{\text{likes to travel}}, & \overbrace{4.6}^{\text{majored in Astrophysics}}, & \dots \end{bmatrix}$$

- Get a measure of similarity between these words by doing:

$$\text{Similarity}(\text{cosmonaut}, \text{astronaut}) = q_{\text{cosmonaut}} \cdot q_{\text{astronaut}}$$

- Normalize by the lengths:

$$\text{Similarity}(\text{cosmonaut}, \text{astronaut}) = \frac{q_{\text{cosmonaut}} \cdot q_{\text{astronaut}}}{\|q_{\text{cosmonaut}}\| \|q_{\text{astronaut}}\|} = \cos(\phi)$$

Where $\phi$ is the angle between the two vectors. Hence, extremely similar words will have similarity 1 and extremely dissimilar words should have similarity -1. In the process of making of a single hot vector, one has to specify an index on each word while using embedding, too. These are keys to a search list. The embeddings matrix has $|V| \times D$ dimensions, where $D$ is the dimensionality of the embeddings, where embedding in the i-th row of the matrix refer to word in index i.

## 5.2 Conclusion

Words can be transformed to scalars across many dialects with `Word2vec` or Glove, and resulting vectors learned, however it should be noted that NLP preprocessing could be quite confined to the specified dialect and may require specific tools for the particular task and could be beyond existing libraries.
Every neural network which process phrasal extracted words into a lower dimensional space and which then fine-tunes with backpropagation necessarily results in word embedding as the first layer weights, which is usually called an embedding layer. This means that words are embedded in the first layer. The main difference between such a network producing word embeddings as a by-product and a technique like `Word2vec`, the explicit objective of which is to generate word embedding is its computational complexity. The generation of words with a very profound architecture is just too computationally expensive for a large vocabulary [104].

Also `Word2vec` and GloVe are focused towards the production of word-integration that encode general semantic relationships that are beneficial to many downstream tasks; in particular, such trained word-incorporation is not useful for tasks that don't depend on this type of relation. Regular neural networks generate task-specific embedding systems which are only limited elsewhere, by contrast [104].

Inability to handle unknown or out-of-vocabulary (OOV) words is another challenge with word embeddings since one is forced to use a random vector every time the model hasn't encountered unknown

or OOV words due to the fact that there hasn't been a logical way on how to build a vector for these OOV words.

No shared representations at sub-word levels seem to be another challenge for the Word Embeddings since every word is represented as an independent vector. Also new embedding matrices are created every time a new languages is involved which leads to new embedding matrices and does not permit parameter sharing, meaning cross-lingual use of the same model isn't an option and neither is scaling.

In many cases one can't leverage the benefits of pre-training and is forced to randomize embeddings due to the fact that the increasingly popularity has led to some of the pre-training word embeddings to be done on weakly supervised or unsupervised data.

# Chapter 6

# Research Methodology

## 6.1 Introduction

In Chapter 3, Chapter 4, and Chapter 5 I covered in details some techniques that were used in the de-identification problem. Least amount of preprocessing is the key since deep learning will play a critical part in learning of the features automatically. This chapter outlines the research methodology to solve the problem.

## 6.2 Research Hypothesis

*Combining carefully-extracted features together with gradient-based algorithms and/or deep learning methods into an inherently-sequential model, a recurrent neural network using LSTM or LSTM-variant, will allow for accurate and robust Text de-identification of the Clinical notes.*

## 6.3 Research Methodology

Using the hypothesis above, I examined various options in order to develop methods that can utilize gradient-based algorithms and/or deep learning methods that will achieve accurate and robust Text de-identification of the Clinical notes. In the process, I developed the de-identification LSTM_CRFs model that could fully utilize a clinical corpus which was developed for the 2014 i2b2 challenge and the 2016 CEGS NGRID. Furthermore, I evaluated the performance using clinical notes collected and compared the performance. The results obtained in this study are further compared to five different word embeddings trained from the general English text, de-identified clinical text, and biomedical literature. The developed LSTM-CRFs model for de-identification achieved superior performance compared with other ML-based methods. Figure 6.1 captures the entire process in a nutshell.

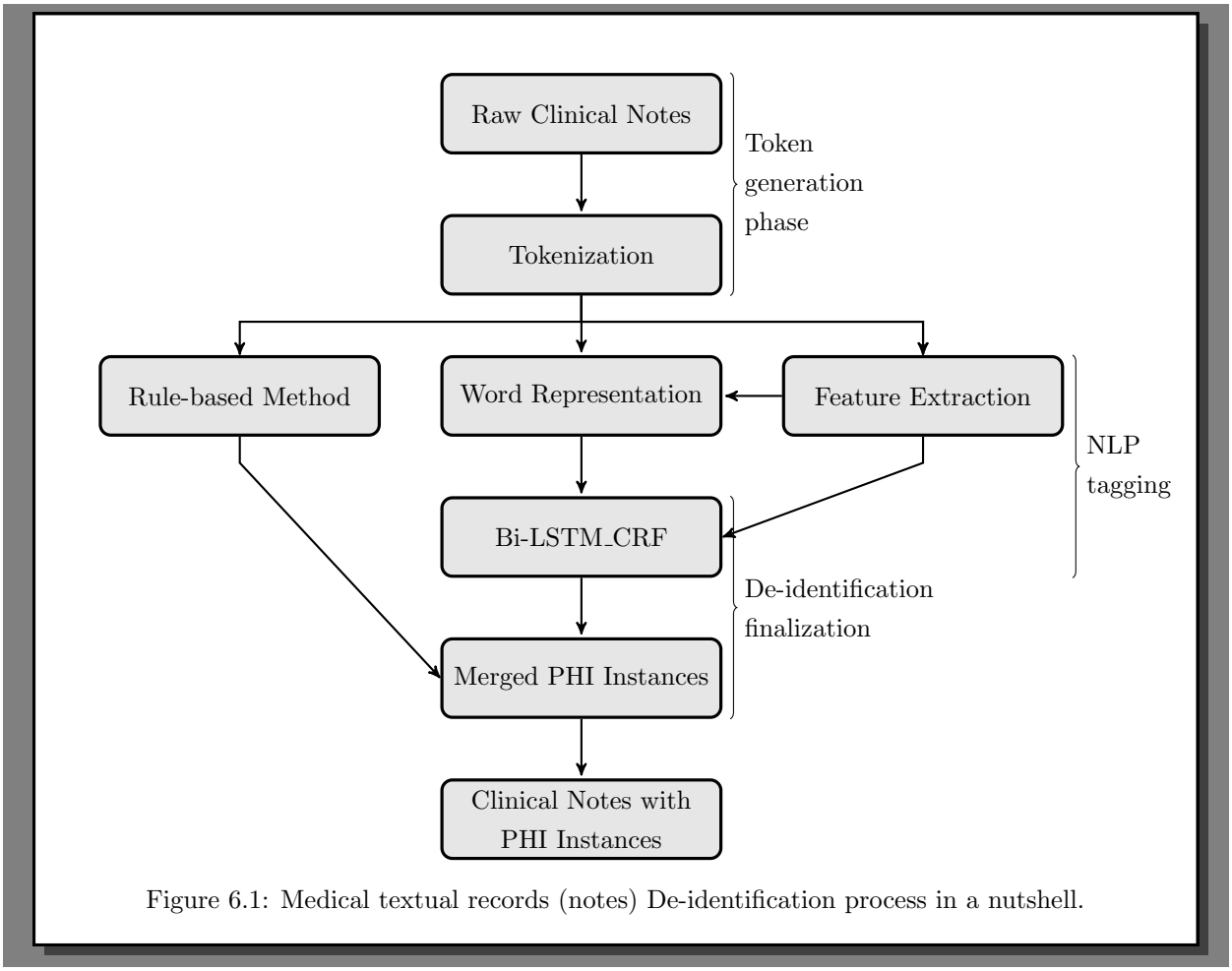



Figure 6.1: Medical textual records (notes) De-identification process in a nutshell.

### 6.3.1 Phase 1: Implementation

This initial phase consisted of the the following three sub-phases:

1. **Dataset collection:** I collected widely-studied Text de-identification clinical-notes datasets from the i2b2 research facilities portal. Primarily the dataset that was used for this study are (i)the i2b2 2014 NLP Contest and (ii) the 2016 CEGS N-GRID textual medical information records corpora.

2. **Assess Tokenization needs** - Identifying parts that require no further decomposition and word representations needed to be fed into our learning algorithms.

3. **Constructing learning model** - Build the learning models that I chose to summarize the text inputs.

#### 6.3.1.1 The textual medical information records sets

The textual medical information records sets used for this study is the i2b2 2014 NLP Contest and the 2016 CEGS N-GRID corpora. The 2016 Centers of Excellence in Genomic Science (CEGS) and Neuropsychiatric Genome-Scale and RDOC Individualized Domains (N-GRID) shared tasks for medical records is a 3-tracks categorized.

Track1 is the de-identification related corpus of over 1000 documents of psychiatric intake. Mental health documents seem to vary considerably from documents seen in i2b2 2014 NLP challenge in terms of their content because they comprise considerably more text and more personally identifiable information on the life of patients including test findings, measurements, disease family history etc.

The i2b2 2014 NLP Challenge dataset contains medical records in XML format. The summarized statistic description of the dataset are presented in Table.6.1 below whereby the corpora total tokens are 1,862,452 and 805,118, the maximum tokens of 4610 and 2984, the minimum tokens of 304 and 617 with average of 1,862.4 and 617.4 tokens per record for the 2016 CEGS N-GRID and the i2b2 2014 respectively.

| | **2016 CEGS N-GRID** | **i2b2 2014** |
|---|---|---|
| Corpus total tokens | 1,862,452 | 805,118 |
| Per record Mean | 1,862.4 | 617.4 |
| Max(Per record) | 4610 | 2984 |
| Min(Per record) | 304 | 617 |

Table 6.1: Token comparison for the textual medical information records used.

Here below is additional statistics snapshot for the i2b2 2014 dataset given in the Table.6.2

| | **i2b2 2014** |
|---|---|
| Records counted | 1304 |
| Tokens counted | 805,118 |
| PHIs counted | 28,862 |
| PHI tokens counted | 38,435 |
| Vocabulary Size | 41,879 |
| ID (Percentage of total) | 3.6% |
| DATE (Percentage of total) | 43.2% |
| HOSPITAL (Percentage of total) | 8.0% |
| DOCTOR (Percentage of total) | 16.6% |
| PATIENT (Percentage of total) | 7.6% |
| AGE (Percentage of total) | 6.9% |

Table 6.2: Overview of the i2b2 2014 datasets.

The annotation rules established in the 2014 i2b2/UTH shared assignment were used by 2016 CEGS N-GRID de-identification task. These rules were designed to include some classifications of the HIPAA PHI to include physicians names and various locations (LOCATION:HOSPITAL), (LOCATION: STATE, Country), (LOCATION: Other) etc. These criteria (shown on Table.6.3) intend to enhance meanings of some HIPAA PHI categories and provide much more granular information.

Table 6.3: Categorical PHI in the textual medical information records used. *Note: "NA" denotes no subcategory, and asterisks denote the HIPAA-defined categories.* [67]

| Main Category | Sub category | i2b2 2014 | 2016 N-GRID |
|---|---|---|---|
| NAME | PATIENT*, DOCTOR USERNAME | 7348 | 6095 |
| PROFESSION | NA | 413 | 2,481 |
| LOCATION | COUNTRY, STATE, CITY* STREET*, ZIP*, HOSPITAL ORGANIZATION*, ROOM DEPARTMENT, OTHER | 4580 | 9896 |
| AGE* | NA | 1,997 | 5,991 |
| DATE* | NA | 12,482 | 9,544 |
| CONTACT | PHONE*, FAX*, EMAIL* URL, IPADDR | 541 | 280 |
| ID | MEDICALRECORD*, SSN* DEVICE*, IDNUM*, BIOID* HEALTHPLAN*, VEHICLE* ACCOUNT*, LICENSE* | 1506 | 77 |
| Total | NA | 28,867 | 34,364 |

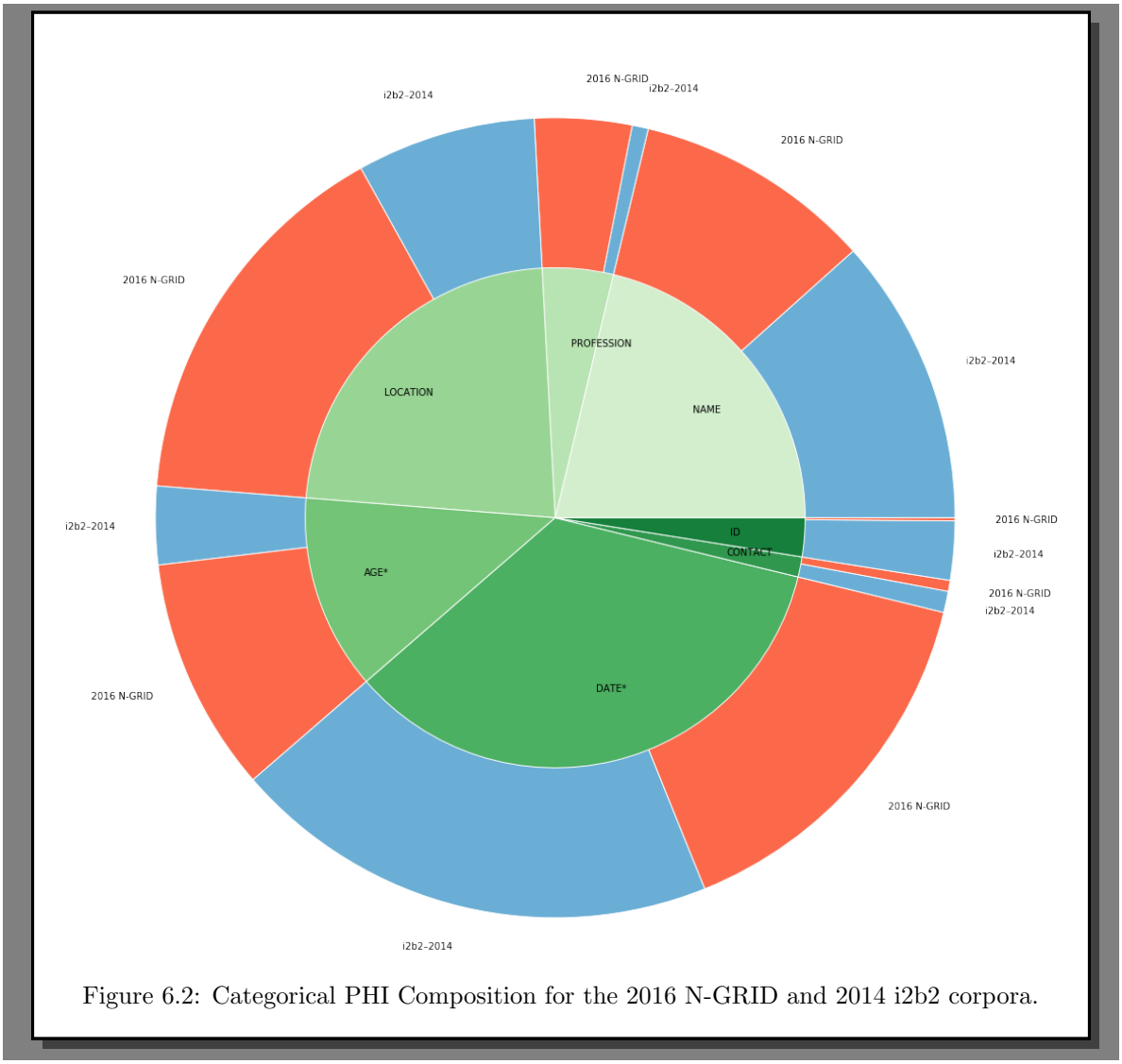



Figure 6.2: Categorical PHI Composition for the 2016 N-GRID and 2014 i2b2 corpora.

#### 6.3.1.2 Tokenization

Identifying parts that require no further decomposition for more reprocessing is of extreme importance. Mistakes at this step will most probably lead to further inaccuracies in the further processing phase. The fact that overall English differs from biomedical text in vocabulary and grammar is particularly difficult to use in biomedical literature [23], where lots of phrases appear to contain punctuation marks and or at times a phrasal syntactic entry.

- **Complexities that are commonly found in English language** [23]:
  - **Hyphenated compound terminology:** For example:
    - Abnormal abdominal x-ray with potential tuberculosis-bacilli from Mycobacterium tuberculosis.
    - 5-year 8-month old female with diarrhea.
  - **Terminology containing letters and slashes:** Apart from frequently being used to indicate distinct or separate entries references (e.g. CD34/IL-102), slashes additionally are used to articulate options (for instance differentiation/activation) or measuring units (for instance kg/L).
  - **Terminology containing letters and apostrophes:** e.g. Small, punctured right lung, below more placing abnormality in 10 percentile given patient's age.
  - **Abbreviations in capital letters and acronyms:**. e.g. Epi Pen were obtained from Dr. M. Narendra: After results of A1c, T1.5, and HbA1c showing abnormalities in order to address the recurrence of type II diabetes.
  - **Terminology containing letters and periods:** Terms with a terminating punctuation mark generally show a sentence's end. They can, however, only be abbreviations, for example i.e. and e.g.
  - **Terminology containing letters and numbers:**. The 31st and 23rd amino acids often times encoded by stop codi, for example, Selenocysteine and Pyrrolysine.
  - **Terminology containing numbers and one type of punctuation:** e.g Previous 1/2 of the x-rays showed no pneumonia present in the patient.
  - **Numeration:** This is the counting or numbering act or process.
  - **A hypertext markup symbol:** Some hypertext marking symbols often observed are < and "
- **Biomedical English complexities:** The fact that general English differs from that of biomedical language in vocabulary and grammar poses a challenge in medical text tokenization. Biomedical literature is linked to medical terminology which is spelled inconsistently and often times with many instances of typographical mistakes and capitalized drugs names. Bio-medical terminology include numbers, letters in various languages from Latin to Greek mixed with letters and Roman numbers, units of measurement next to alphabets, list and enumeration, tabular details, hyphens and other unique symbols [23]. Some examples include:
  - **A DNA sequence:** These are characterized by a sequence of letters and numbers.
  - **Temporal expressions:**e.g. The stroke was last detected on the Impedance Cardiography dated 4/15/1981.
  - **Chemical substances:** e.g. Aspirin, which were generated from 2-acetoxybenzoic acid acetylsalicylate acetylsalicylic acid O-acetylsalicylic acid or Bismuth subsalicylate, also known as Pepto-Bismol

In de-identification methods presented to the NLP challenge; the character level tokenization and the organizer's tokenization procedures were used. The tokenization at the character level immediately broke all phrases in characters and digits words into one character e.g. "70y/o Chilean" into "7 0 y / o C h i l e a n". In 2014 i2b2, the tokenization tool had errors splitting abbreviations in the corpus that appeared adjacent to punctuation. Thus, the existing character-level tokenization module for the 2014 i2b2 NLP Challenge Tool required a little tweak to match that of the 2016 CEGS N-GRID NLP Challenge.

#### 6.3.1.3 Model

The purpose and intention here was to design and implement an appropriate model for the system to learn how to de-identify clinical reports automatically and as general as possible. The Figure. 6.1 visually represents the different steps and components of the de-identication model. I chose to implement a Recurrent Neural Network with Bi-Directional LSTM_CRF as the model to address challenges of the task at hand. This model was fed the tokens as inputs. I tested and observed performance of various optimization parameters e.g. Adagrad, Adadelta, RMSprop, Adam, SGD, and ASGD in the model I developed. See convergence graphs using various optimization methods from Adagrad (Figure. 7.3b), Adadelta (Figure. 7.6b), RMSProp (Figure. 7.1b), Adam (Figure. 7.2b), SGD (Figure.7.5b) and ASGD (Figure.7.4b). However after multiple testing I implemented using Adam as the Optimization parameter of choice. I choose the learning rate of $\eta = 0.01$ to ensure that I have reasonably great convergence, with decay rate $\rho = 0.0001$, hidden dimension to be 50, embedding dimension to be 25, and t is the number of epoch completed. Refer to Figure.6.3 on the importance of appropriate learning rate and other important parameters. The RNN architecture is BiLSTM_CRF.

**Pre-trained Word Embeddings** are a commonly utilized NLP technology and thus can dramatically increase convergence performance substantially [15]. Word embedding (dense vectors usually in 100 to 300 dimensions) generalizes unseen words well because they are capable of capturing both general syntactic and semantic properties. Nevertheless, the outcomes published in several word embedding procedures are not recognized and depends on a variety of considerations, along with the data sets with which they are created and used.

Table 6.4: Hyper-parameters used for the neural network.

| **Hyper-parameter** | **Value** |
|---|---|
| Dimension of token-level word representation | 50 |
| Dimension of character word representation | 25 |
| Dimension of feature representation | 50 |
| Character-level LSTM size | 25 |
| Token-level LSTM size | 100 |
| Learning rate | 0.01 |
| Decay rate | 0.0001 |
| Training epochs | 300 |

In this study, I used a Bi-LSTM_CRF at the character level. Each character $c_i$ of a word $\kappa = [c_1, , c_p]$ is then associated to a vector $c_i \in \mathbb{R}^{d_3}$. When running a Bi-LSTM_CRF over the sequence of character embeddings and concatenate the final states to obtain a fixed-size vector $\kappa_{chars} \in \mathbb{R}^{d_2}$. It should be noted that this vector captures the morphology of the word. By concatenating $\kappa_{chars}$ to the word embedding $\kappa_{torchtext}$ to get a vector representing our word $\kappa = [\kappa_{torchtext}, \kappa_{chars}] \in \mathbb{R}^n$ with $n = d_1 + d_2$. The Contextual Word Representation for each word in its context; the Bi-LSTM_CRF model helps to get a meaningful representation $h \in \mathbb{R}^k$. There are multiple options of pre-trained word embeddings

$\kappa_{GloVe} \in \mathbb{R}^{d_1}$ for GloVe, $\kappa_{Word2vec} \in \mathbb{R}^{d_1}$ for Word2Vec, $\kappa_{senna} \in \mathbb{R}^{d_1}$ for Senna, etc., to choose from in order to generate the dense representation $\kappa \in \mathbb{R}^n$ for each word and PyTorch supports all those embedding vector options.

An **optimizer** ensures that the objective function of the neural network is minimized. A commonly selected optimizer for a large number of published studies on machine learning systems is stochastic gradient descent (SGD), which proves to be an effective and efficient method for optimizing. SGD can, however, be quite sensitive to learning rate selection. The choice of abnormally high rate could lead the system to tie resources on heavy computation of the objective mathematical expressions function and result in a slow learning process when a too low rate is chosen. In addition, SGD has difficulties navigating ravines and saddles. Utilization of other gradient-based optimization (Adagrad [24], Adadelta [130], RMSProp [50], Adam [57], and Nadam [124]algorithms have been previously proposed to deal with SGD shortcomings. For this study I chose Adam Optimizer.

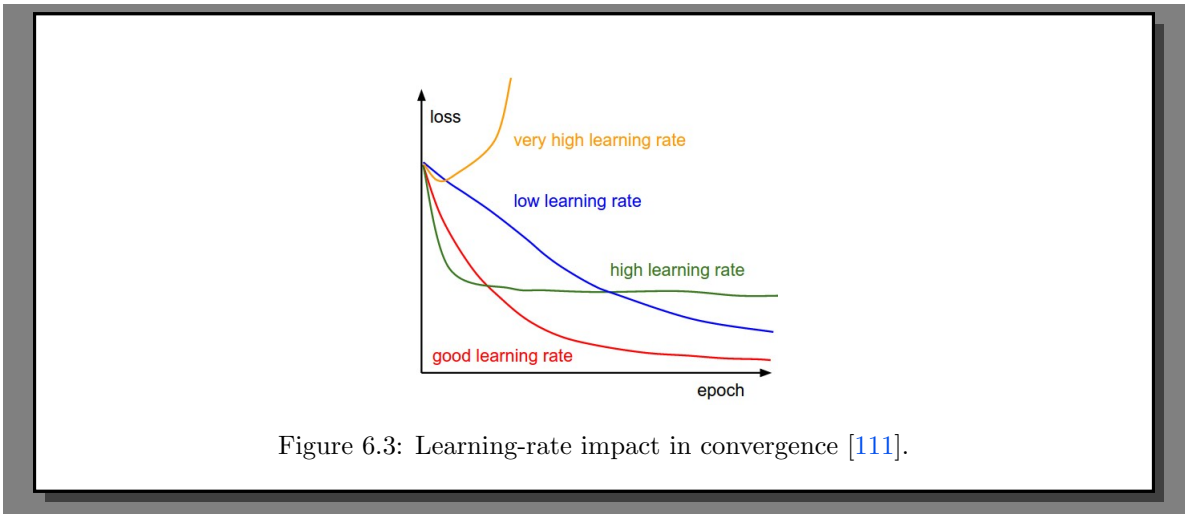



Figure 6.3: Learning-rate impact in convergence [111].

**Classifier:** The BiLSTM_CRF approach I used applied the Softmax together with the CRF classifier, produces the probability distribution for each of the tags in the sentence. The combination allows the tag probability for the entire sentence to be maximized. This is particularly helpful for tasks with highly dependent token tags/labels [53].

**Tagging schemes:** The scheme applied used the associated HIPAA tags from the XML medical records.

**Dropout:** Dropout is a popular successful approach/method to deal with over-fitting for connectionist computing paradigm mirroring the biologic structure of brain or nervous system referred as neural networks. [110].I tried a variation of 0.0, 0.05, 0.1, 0.25, and 0.5 before selecting the optimal one.

**Packages:** I tried using Keras with Theano and TensorFlow as back-end however due to the mathematical operations and numerical instabilities, the results differed between Theano and TensorFlow. I switched to PyTorch which worked extremely well.

Other packages; *SpaCy* despite being easy to use, very fast, and ready for production however they are not easily customizable and the internals are opaque with very little documentation. *AllenNLP* seem to be an excellent option for general-purpose NLP functionality and is designed for quick prototyping however it is not yet mature and not optimized for speed.

The Figure.6.1 visually represents the different steps and components of a similar de-identication model. I have chosen to implement a Recurrent Neural Network with Bi-Directional LSTM_CRF

as the model to address challenges of the task at hand. This model was fed the word embedding tokens as inputs. Parameter settings optimization is carried out using the implementation of the Adam Optimizer. I choose the learning rate of $\eta = 0.01$, with decay rate $\rho = 0.0001$, hidden dimension to be 50, embedding dimension to be 25, and t is the number of epoch completed. The complete list of hyperparameters used can be found on Table.6.4. The RNN archtecture is BiLSTM_CRF.

### 6.3.2 Phase 2: Training

At the beginning of the study I and as part of experimentation/exploration, I created a 5-fold: by splitting a randomly selected sample dataset into MedRec_A, MedRec_B, MedRec_C, MedRec_D, MedRec_E

- Set 1: Train on MedRec_A, MedRec_B, MedRec_C, MedRec_D — test using MedRec_E
- Set 2: Train on MedRec_A, MedRec_B, MedRec_C, MedRec_E — test using MedRec_D
- Set 3: Train on MedRec_A, MedRec_B, MedRec_D, MedRec_E — test using MedRec_C
- Set 4: Train on MedRec_A, MedRec_C, MedRec_D, MedRec_E — test using MedRec_B
- Set 5: Train on MedRec_B, MedRec_C, MedRec_D, MedRec_E — test using MedRec_A

Using the exploration method that I had setup was able to give me the similar results to the ones obtained by the study performed by Stéphane et al. I used the setup for preliminary training and testing of network.

However, such a method of creating a 5-fold by splitting a randomly selected sample dataset into MedRec_A, MedRec_B, MedRec_C, MedRec_D, MedRec_E was not how the previous study was conducted —hence I abandoned it. I reverted to the method that allowed me to fully utilize the entire organizer's provided Training sets, Testing sets, and Validation methods with thousands of records. Doing so allowed me to utilize over 50 simulations for this study.

### 6.3.3 Phase 3: Testing

Testing of the system is key in determining its effectiveness and potential utility. A hold-out subset of the input dataset will be reserved for this purpose. Testing involves feeding test clinical notes into the system, computing, and analyzing the various metrics for performance.

- The two(2) basic effectiveness measures that will be used: *Precision* and *Recall*. Precision is the percentage of positive predictions that were correct. Recall is how good we find our all positive predictions. Precision(i) is the precision at the top i tags, relevance(i) is 1 if relevant and 0 otherwise. These are calculated as:

$$Precision = \frac{Total\ of\ Relevant\ retrieved}{Total\ of\ Retrieved}$$
$$Recall = \frac{Total\ of\ Relevant\ retrieved}{Total\ of\ Relevant}$$

- The following lists the performance measures used in this work:

| | Is relevant | Is NOT relevant |
|---|---|---|
| Is classified | true positive | false positive |
| Is NOT classified | false negative | true negative |

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$
$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

- *Accuracy*: categorization ratio between relevant to the not relevant that is correct

$$accuracy = \frac{TruePositive + TrueNegative}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

- Recall and precision measures are interdependent.
  - Precision is inversely proportional to the retrieved which implies that the normally lessens as the retrieved number increases.
  - Recall is direct proportional to the retrieved which implies that Recall tends to increase as the retrieved number rises.

- Measure relating precision and recall are provided using the mathematical expressions:

$$F = \frac{1}{\alpha \times \frac{1}{Precision} + (1 - \alpha) \times \frac{1}{Recall}}$$

- When the coefficient $\alpha = 0.5$:

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

- Its better to recognize that when one is dealing with extreme values one should use a harmonic mean rather than an arithmetic mean.

The "strict" micro-average precision, recalls and F1-scores of my system and its subsystems (a list of which is on Table.6.3) are listed in Table.7.4. When compared to the machine learning-based subsystems from Table.7.1, the Bi-LSTM_CRF achieved overall much better Precision, Recall, and some of F1-scores than previous subsystem with "strict" micro-average Precision of 99.08% on the 2014 i2b2 and 92.50% on the 2016 N-GRID corpora, Recall of 97.19% on the 2014 i2b2 and 88.04% on the 2016 N-GRID corpora, and F1-scores of 96.57% on the 2014 i2b2 and 90.27% on the 2016 N-GRID corpora. These results Precision are higher than Ensemble-based system by Stéphane et al. by 2.62% on 2014 i2b2 but lower by 1.55% on the 2016 N-GRID corpus, on Recall higher by 3.39% on 2014 i2b2 also higher by 0.04% on the 2016 N-GRID corpus, on F1-scores higher by 1.46% on 2014 i2b2 but lower by 0.65% on the 2016 N-GRID corpus.

### 6.3.4 Evaluation

In addition, the evaluation script supplied by the challenge organizers permits 3 restrictiveness tiers :

- **Strict** (also known as phrase-term predicated with paragraph components match processing method): Here the primary and terminating offset must respectively resemble but flexible on others.

- **Relaxed:** Term predicated however the last offset up to 2 characters could also be deactivated.
- **Overlap:** (also known as token-predicated match): matches the scheme if the token in a gold standard PHIs is annotated.

There are two variants of PHI detection methods in the assessment code:

- **All PHI:** All gold-standard PHIs, that are more stringently annotated than required by HIPAA.
- **HIPAA PHI:** PHI required solely by HIPAA.
  - Exempts physician names, names of hospitals, occupations, sites bigger than a state.
- Exempts physician names, names of hospitals, occupations, sites bigger than a state.

The evaluation tools of the i2b2 2014 De-identification challenge organizers and the 2016 CEGS N-GRID NLP Contest have been used in this study to conduct official test. Refer to Figure.6.4 showing a sample output. The tools output micro-average precision (p), recall (R), and F2 scores (F1) using the 10 criteria, both of the (i2b2 2014 De-identification contest and the CEGS N-GRID NLP 2016 contest). They are 'token', 'strict', 'relaxed', 'binary strict' HIPAA, 'binary strict', 'binary HIPAA strict' or 'binary strict' HIPAA strict.

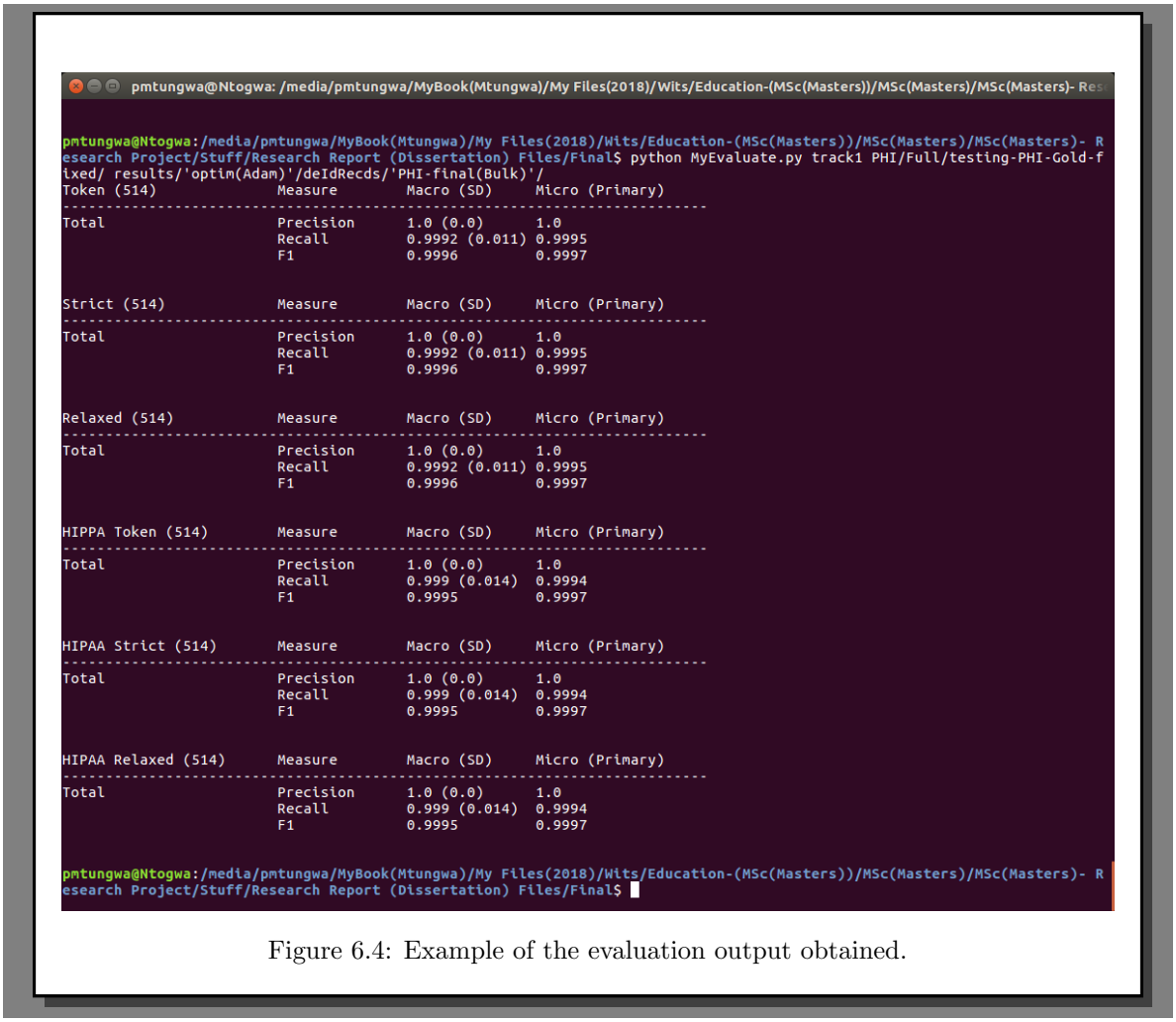



Figure 6.4: Example of the evaluation output obtained.

In this case; 'token' checks in this case if a forecast correct token instance corresponds to a token exactly in the same category in a gold phrase, 'strict' if a PHI outcome is correctly obtained just when it precisely corresponds a gold one from the same category and limit, 'relaxed', if a PHI outcome is properly obtained only when it vastly overlaps with a gold one from the same class, 'HIPAA' for the

19 attributes of HIPAA-defined PHIs, and binary for the boundaries of PHI instances no matter their classifications.

# Chapter 7

# Findings

## 7.1 De-identification results from previous studies

The textual medical information records sets used for this study is the i2b2 2014 NLP Contest and the 2016 CEGS N-GRID corpora. The section 6.3 of Chapter 6 provides comprehensive detail characteristics of and the information about the datasets.

The Table.7.1 and Table.7.2, provide the necessary metric (precision, recall and F1) values side-by-side view from studies that were conducted previously on the same datasets using various systems. Until the ascension of BiLSTM_CRF the Nottingham [22] scheme was the greatest de-identification challenge in i2b2 2014. The MIST utility is a PHI removal commercial tool and Dernoncourt et al. has suggested CRF + ANN. Bi-LSTM with Bi-GRU (Bidirectional Gated Recurrent Unit) are the traditional multi-modal RNN model.

Table 7.1: Comparison between some of available tools or methods [22].

| **Model** | **i2b2-2014** | | |
|---|---|---|---|
| | **Precision** | **Recall** | **F1-score** |
| Wellner | - | - | - |
| Nottingham | 0.9900 | 0.9640 | 0.9768 |
| MIST | 0.9529 | 0.7569 | 0.8467 |
| CRF | 0.9842 | 0.9663 | 0.9752 |
| CRF + ANN | 0.9792 | 0.9784 | 0.9788 |
| Bi-LSTM | 0.9878 | 0.9389 | 0.9627 |
| Bi-GRU | 0.9750 | 0.9704 | 0.9727 |

Table 7.2: Previous study results on the same datasets utilizing different techniques (strict, %) [67].

| **Methods** | **i2b2 2014** | | | **2016 N-GRID** | | |
|---|---|---|---|---|---|---|
| | **Precision** | **Recall** | **F1-score** | **Precision** | **Recall** | **F1-score** |
| CRF-based | 95.16 | 90.12 | 92.58 | 92.47 | 84.61 | 88.37 |
| BI-LSTM | 95.26 | 93.34 | 94.29 | 90.05 | 88.21 | 89.12 |
| BI-LSTM-FEA | 95.43 | 93.61 | 94.51 | 91.43 | 88.57 | 89.98 |
| Ensemble | 96.46 | 93.80 | 95.11 | 94.05 | 88.00 | 90.92 |
| Rule-based | NA | NA | NA | 91.10 | 0.984 | 1.947 |
| **Overall** | 96.46 | 93.80 | 95.11 | 94.22 | 88.81 | 91.43 |

## 7.2 De-identification results from this study

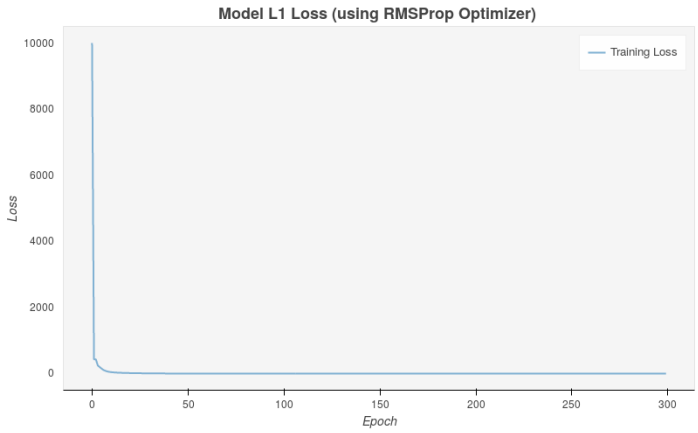
#### 7.2.0.1 Training and testing

In an effort to ensure that I examine different ways to achieve the best possible results I applied and tested different optimization methods. For instance, a single learning rate (Alpha) is maintained for all weight updates, while the learning rate does not change during training with Stochastic gradient descent. For each network weight (parameter) and separately adapted as learning takes place, a learning rate is maintained. With Adaptive Gradient Algorithm (AdaGrad) it is known to maintain a learning rate per parameter, which increases performance in sparse-gradient problems. RMSProp maintain average of the last gradients by parameter of the weight so the algorithm performs well in non-stationary problems, also using the root mean Square propagation (RMMSProp) similar observations were achieved. However Adam calculates an exponentially moving average of the gradient and the squared gradient and beta1 and beta2 control the rate of decay of these moving averages hence realizes the benefits of both AdaGrad and RMSProp. Given those benefits I decided to implement Adam as the best choice for the study,

When setting up the EMBEDDING_DIM to 25 and HIDDEN_DIM to 50 as suggested by Zengjian Liu [67] at al with the optimal parameters then I was able to obtain the wide range of convergence results. Refer to convergence graphs using various optimization methods from Adagrad (Figure.7.3b), Adadelta (Figure.7.6b), RMSProp (Figure.7.1b), Adam (Figure.7.2b), SGD (Figure.7.5b) and ASGD (Figure.7.4b). The corresponding classification reports from Adagrad (Figure.7.3a), Adadelta (Figure.7.6a), RMSProp (Figure.7.1a), Adam (Figure.7.2a), SGD (Figure.7.5a) and ASGD (Figure.7.4a)

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| HOSPITAL | 1.00 | 1.00 | 1.00 | 4 |
| DOCTOR | 1.00 | 1.00 | 1.00 | 10 |
| PATIENT | 1.00 | 1.00 | 1.00 | 4 |
| COUNTRY | 1.00 | 1.00 | 1.00 | 1 |
| STATE | 1.00 | 1.00 | 1.00 | 2 |
| DATE | 1.00 | 1.00 | 1.00 | 47 |
| AGE | 1.00 | 1.00 | 1.00 | 6 |
| avg / total | 1.00 | 1.00 | 1.00 | 74 |

(a) Classification report using RMSprop optimizer.




(b) Loss using RMSprop optimizer.

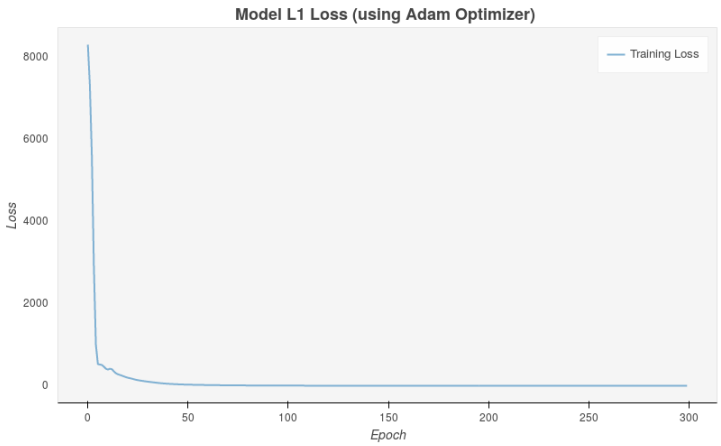
Figure 7.1: Training using RMSprop optimizer resulting the F1-score: 100.0%.

As I indicated earlier in section 4.2 that RNNs are set back by the inherent problem of the vanishing and exponentially increasing slope or gradient [7, 95], where the gradient tend to either vanish to zero or to increase exponentially during BPTT. One of the advantages of long-short term memory (LSTM) layers

is that they are more resilient to the rapidly diminishing gradient problem [42]. However, we can clearly see in Figure.7.1b that this is a typical case of a quickly vanishing as the model is trying to learn during the training process.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| STATE | 1.00 | 1.00 | 1.00 | 2 |
| DATE | 1.00 | 1.00 | 1.00 | 47 |
| PATIENT | 1.00 | 1.00 | 1.00 | 4 |
| DOCTOR | 1.00 | 1.00 | 1.00 | 10 |
| AGE | 1.00 | 1.00 | 1.00 | 6 |
| HOSPITAL | 1.00 | 1.00 | 1.00 | 4 |
| COUNTRY | 1.00 | 1.00 | 1.00 | 1 |
| avg / total | 1.00 | 1.00 | 1.00 | 74 |

(a) Classification report using Adam optimizer.




(b) Loss using Adam optimizer.

Figure 7.2: Training using Adam optimizer resulting the F1-score: 100.0%.

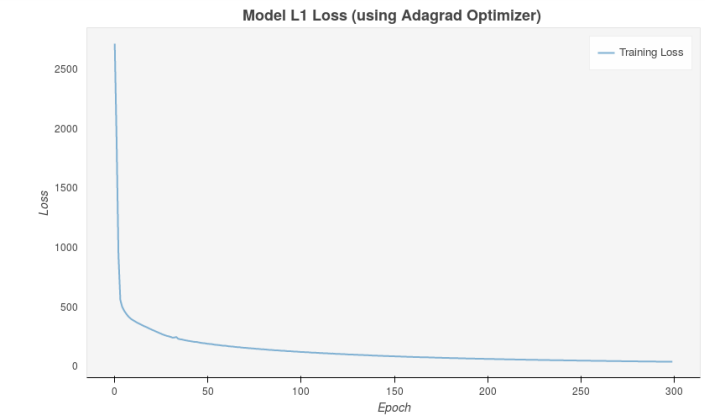
With Adam optimizer on Figure.7.2b despite observing a sharp decline in the gradient one can see that a great gentle slope developing right after 5 epochs which demonstrates that the model is learning well as it the losses diminish and convergence is attained after a reasonable number of epochs.

While using Adagrad as seen on Figure.7.3b I could see that the neural network was struggling a bit to learn as it took a substantial amount of time for convergence to happen. This is also evidenced by lower precision, recall, and F1-scores as compared to Adam optimizer.

Despite a few vast losses ASGD proved to be an alternative optimizer as depicted on Figure.7.4b.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| PATIENT | 1.00 | 0.75 | 0.86 | 4 |
| DOCTOR | 1.00 | 0.70 | 0.82 | 10 |
| STATE | 0.00 | 0.00 | 0.00 | 2 |
| DATE | 1.00 | 1.00 | 1.00 | 47 |
| HOSPITAL | 1.00 | 1.00 | 1.00 | 4 |
| AGE | 1.00 | 0.50 | 0.67 | 6 |
| COUNTRY | 0.00 | 0.00 | 0.00 | 1 |
| avg / total | 0.96 | 0.86 | 0.90 | 74 |

(a) Classification report using Adagrad optimizer.

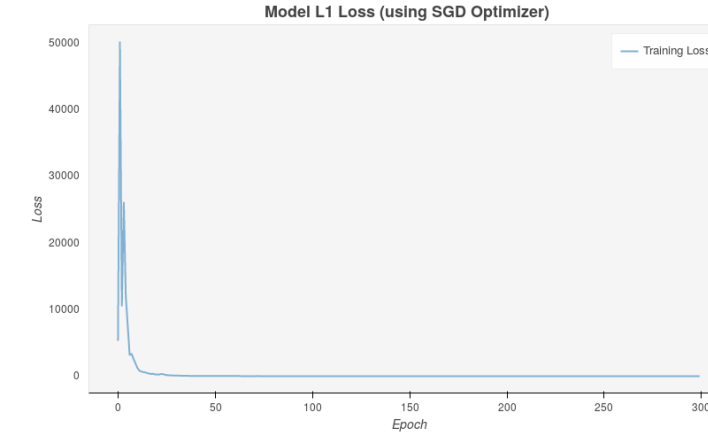



(b) Loss using Adagrad optimizer.

Figure 7.3: Training using Adagrad optimizer resulting F1-score: 92.8%.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| DATE | 1.00 | 1.00 | 1.00 | 47 |
| COUNTRY | 1.00 | 1.00 | 1.00 | 1 |
| PATIENT | 1.00 | 1.00 | 1.00 | 4 |
| STATE | 1.00 | 1.00 | 1.00 | 2 |
| DOCTOR | 1.00 | 1.00 | 1.00 | 10 |
| HOSPITAL | 1.00 | 1.00 | 1.00 | 4 |
| AGE | 1.00 | 1.00 | 1.00 | 6 |
| avg / total | 1.00 | 1.00 | 1.00 | 74 |

(a) Classification report using ASGD optimizer.

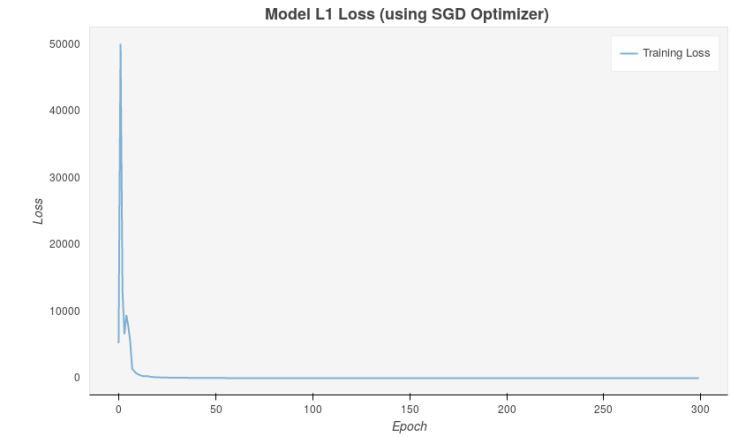



(b) Loss using ASGD optimizer.

Figure 7.4: Training using ASGD optimizer resulting the F1-score: 100.0% .

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| COUNTRY | 0.00 | 0.00 | 0.00 | 1 |
| DATE | 1.00 | 1.00 | 1.00 | 47 |
| HOSPITAL | 1.00 | 1.00 | 1.00 | 4 |
| STATE | 0.67 | 1.00 | 0.80 | 2 |
| AGE | 1.00 | 1.00 | 1.00 | 6 |
| DOCTOR | 1.00 | 1.00 | 1.00 | 10 |
| PATIENT | 1.00 | 1.00 | 1.00 | 4 |
| avg / total | 0.98 | 0.99 | 0.98 | 74 |

(a) Classification report using SGD optimizer.

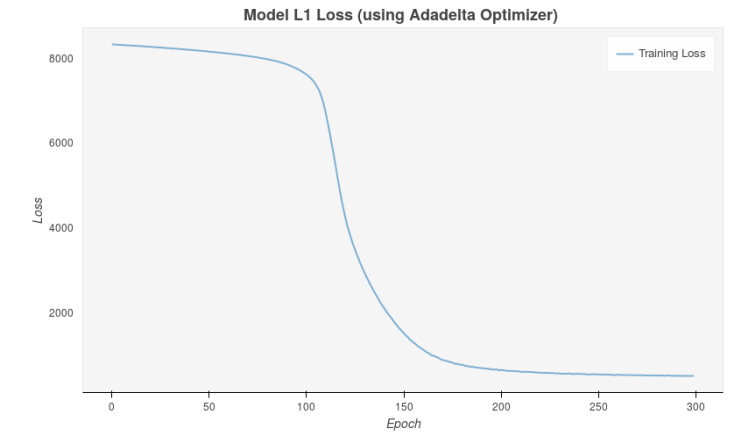



(b) Loss using SGD optimizer.

Figure 7.5: Training using SGD optimizer resulting the F1-score: 98.6%.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| COUNTRY | 0.00 | 0.00 | 0.00 | 1 |
| HOSPITAL | 0.00 | 0.00 | 0.00 | 4 |
| STATE | 0.00 | 0.00 | 0.00 | 2 |
| PATIENT | 0.00 | 0.00 | 0.00 | 4 |
| AGE | 0.00 | 0.00 | 0.00 | 6 |
| DOCTOR | 0.00 | 0.00 | 0.00 | 10 |
| DATE | 0.00 | 0.00 | 0.00 | 47 |
| avg / total | 0.00 | 0.00 | 0.00 | 74 |

(a) Classification report using Adadelta optimizer.




(b) Loss using Adadelta optimizer.

Figure 7.6: Training using Adadelta optimizer resulting the F1-score: 0.0%.

Just like ASGD, despite a few vast early losses SGD proved to be an alternative optimizer as depicted on Figure.7.5b despite lower precision, recall, and F1-scores as compared to Adam.

On the other hand, Adadelta proved not to be useful optimizer for this task given that a high plateau was sustained for almost 150 epochs before a sharp cliff and another plateau. This was indicative of inability to converge and arrive to reasonable log-likelihood that will provide better predictions of the PHIs and arrive to the goals of this study. As evidenced by figures from Adagrad 7.3b, Adadelta 7.6b, RMSProp 7.1b, Adam 7.2b, SGD 7.5b and ASGD 7.4b that a great choice of an optimizer and other hyperparameters are essential in order to obtain convergence and hence accurate predictions. The Table 7.3 summarizes these classification reports results from Adagrad, Adadelta, RMSProp, Adam, SGD and ASGD optimizers.

Table 7.3: Optimizers Classification Summary.

| PHI Type | Method | i2b2 2014 | | | 2016 N-GRID | | |
|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1-score | Precision | Recall | F1-score |
| AGE | Adadelta | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Adagrad | 1.00 | 0.50 | 0.67 | 1.00 | 0.55 | 0.70 |
| | Adam | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | RMSprop | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | SGD | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | ASGD | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | min | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | max | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | mean | 0.83 | 0.75 | 0.78 | 0.83 | 0.76 | 0.78 |
| COUNTRY | Adadelta | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Adagrad | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Adam | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | RMSprop | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | SGD | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | ASGD | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | min | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | max | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | mean | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 |
| DATE | Adadelta | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Adagrad | 1.00 | 0.50 | 0.67 | 1.00 | 0.52 | 0.68 |
| | Adam | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | RMSprop | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | SGD | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | ASGD | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | min | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | max | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | mean | 0.83 | 0.75 | 0.78 | 0.83 | 0.75 | 0.78 |
| DOCTOR | Adadelta | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Adagrad | 1.00 | 0.70 | 0.82 | 1.00 | 0.75 | 0.86 |
| | Adam | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

continued on next page

| PHI Type | Method | i2b2 2014 | | | 2016 N-GRID | | |
|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1-score | Precision | Recall | F1-score |
| | RMSprop | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | SGD | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | ASGD | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | min | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | max | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | mean | 0.83 | 0.78 | 0.80 | 0.83 | 0.79 | 0.81 |
| HOSPITAL | Adadelta | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Adagrad | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | Adam | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | RMSprop | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | SGD | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | ASGD | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | min | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | max | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | mean | 0.83 | 0.83 | 0.83 | 0.83 | 0.83 | 0.83 |
| PATIENT | Adadelta | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Adagrad | 1.00 | 0.75 | 0.86 | 1.00 | 0.70 | 0.82 |
| | Adam | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | RMSprop | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | SGD | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | ASGD | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | min | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | max | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | mean | 0.83 | 0.79 | 0.81 | 0.83 | 0.78 | 0.80 |
| STATE | Adadelta | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Adagrad | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Adam | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | RMSprop | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | SGD | 0.67 | 1.00 | 0.80 | 0.71 | 1.00 | 0.83 |
| | ASGD | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | min | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | max | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | mean | 0.61 | 0.67 | 0.63 | 0.62 | 0.67 | 0.64 |

This architecture performs extremely well as compared to the systems on Table.7.1 that use the Adam optimizer, whereas most of use the Stochastic Gradient Descent (SGD) optimizer. In fact, Reimers et al. [101] show that the SGDoptimizer performed considerably worse than the Adam optimizer for different NLP tasks.

I should point out that testing outcome on multiple occasion produced by the organizers evaluation tools produced the results as shown in Table.7.5 and Table.7.6. Here below on Table.7.4 are the results that I was able to obtain in this study:

Table 7.4: The results I obtained in this study (strict, %).

| Methods | i2b2 2014 | | | 2016 N-GRID | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1-score | Precision | Recall | F1-score |
| **Bi-LSTM + CRF** | 99.08 | 97.19 | 96.57 | 92.50 | 88.04 | 90.27 |

#### 7.2.0.2 Further Evaluation

Every validation assessment was carried out in the entire test pairs with a screening tool supplied by NLP contest organizers. The tools provided outputs indicates that an instance of PHI is extracted correctly only if it matches exactly the same category and gold boundary. Depending on the token encountered the system the indicates that an instance of PHI is extracted correctly only when it overlaps (only two characters at the end of it mismatched), with a gold one of the same category, "HIPAA" takes into account only categories characterized by HIPAA as PHIs (refer to Figure.1.1 for the list), and "binary" considers the PHI boundary without consideration of the PHI categories. The "strict" is the main criterion of these ten(10).

Table 7.5: My de-identification system performance on major PHI classes ('strict', %).

| PHI Type | i2b2 2014 | | | 2016 N-GRID | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1-score | Precision | Recall | F1-score |
| NAME | 0.9642 | 0.9453 | 0.9672 | 0.9501 | 0.8990 | 0.9544 |
| PROFESSION | 0.9134 | 0.6640 | 0.7768 | 0.8233 | 0.7311 | 0.7575 |
| LOCATION | 0.9325 | 0.8769 | 0.9197 | 0.9245 | 0.8466 | 0.8492 |
| AGE | 0.9863 | 0.9642 | 0.9851 | 0.9366 | 0.9022 | 0.9589 |
| DATE | 0.9853 | 0.9311 | 0.9888 | 0.8997 | 0.9305 | 0.9421 |
| CONTACT | 0.9772 | 0.9579 | 0.9642 | 0.9167 | 0.8829 | 0.9504 |
| ID | 0.9440 | 0.9104 | 0.9311 | 0.8559 | 0.6347 | 0.7011 |

Table 7.6: Optimal research-produced outcomes currently by my methods.

| Methods | i2b2 2014 | | | 2016 N-GRID | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1-score | Precision | Recall | F1-score |
| Token | 0.9802 | 0.9693 | 0.9772 | 0.9360 | 0.8902 | 0.9111 |
| Strict | 0.9908 | 0.9719 | 0.9657 | 0.9250 | 0.8804 | 0.9027 |
| Relaxed | 0.9755 | 0.9509 | 0.9631 | 0.9398 | 0.9175 | 0.9602 |
| Binary Token | 0.9922 | 0.9647 | 0.9877 | 0.9784 | 0.9812 | 0.9293 |
| Binary Strict | 0.9803 | 0.9591 | 0.9778 | 0.9732 | 0.9497 | 0.9145 |
| HIPAA token | 0.9852 | 0.9759 | 0.9782 | 0.9319 | 0.9471 | 0.9367 |
| HIPAA strict | 0.9775 | 0.9599 | 0.9724 | 0.9577 | 0.9715 | 0.9366 |
| HIPAA relaxed | 0.9812 | 0.9591 | 0.9702 | 0.9418 | 0.9448 | 0.9152 |
| HIPAA binary token | 0.9872 | 0.9729 | 0.9805 | 0.9688 | 0.9525 | 0.9419 |
| HIPAA binary strict | 0.9756 | 0.9544 | 0.9651 | 0.9720 | 0.9218 | 0.9466 |

Liu et al. observed that LSTM subsystems are better than that of the CRF subsystems because LSTM subsystems have far greater recalls that LSTM models can determine more PHI instances than CRF. The subsystems are more effective than CRF. Also if BI-LSTM contributes a little handcraft functionality, then will be improvements in both accuracy and recalls. Hence they concluded that right functionality implementation is a useful de-identification complement to the LSTM [67].

## 7.3 Conclusion

In this thesis I was able to obtain slightly higher F1-score as compared to some previously conducted i2b2 2014 data study overall scores and lower F1-score as compared to overall scores on the 2016 N-GRID data. Refer to Table.7.2 and Table.7.4 respectively for details. However, I have shown that using the two commonly accessible gold standard de-identification datasets my Bi-LSTM_CRF architecture, which includes the recent advances in situational word embedding and NLP, actually accomplishes state-of-the-art efficiency and produce great results that meet (and exceed) the threshold requirements as outlined by the guidelines for the HIPAA Expert determination method the I briefly discussed and covered in Chapter 1 Section 1.0.2.

The amount of the medical identification numbers that my architecture forecast as a telephone number because of the presented number format in the medical records was notable however a use of simple regular expression can readily alleviate such mistakes. Also despite using a "combination of n-gram, morphological, orthographic and gazetteer features" by Dernoncourt et al. the best performing models on the i2b2 dataset by Dernoncourt et al. [22] achieved similarly low scores to mine on the Profession and ID categories. However, on ID PHI, my assessment demonstrates that some mistakes have been caused by tokenization problems.

Machine Learning can be a strategic advantage for any corporation whether it's a top player or a start - up, as tomorrow tasks are handcrafted by computers. The revolution in machine learning will remain long here and the foreseeable future of machine learning will continue. Within the next few years the confidence shortfall in the' technological solutions capabilities' will disappear. We will see a shift from limited distrust and cynicism to total reliance on AI and other sophisticated technology sooner than later. The majority of AI-propelled implementations face consumers, which is yet another reason that popular users over time try to overcome the confidence barrier. More exposure and more access for the customer in their daily business to practical solutions pave the way for a globe propelled by new technologies.

Lastly, there is a need to develop a mechanism to address shared representations at sub-word level. In the current setup there is no shared representations at sub-word levels and this seems to be another challenge for the Word Embeddings since every word is represented as an independent vector. Also new embedding matrices are created every time a new language is involved which leads to new embedding matrices and does not permit parameter sharing, meaning cross-lingual use of the same model isn't an option and neither is scaling.

# Chapter 8

# Future Work

Since the goal of this entire de-identification process is to ensure patients privacy and unauthorized disclosure of healthcare records with PHIs, therefore, I genuinely believe it is imperative for centralized access controlled de-identification system to be successfully implemented around confidentiality, integrity and availability paradigm to address many of the privacy issues but also allow authorized access —this could equally been a reason why many of the earlier de-identification infrastructures are predicated on data center systems.

Secondly, an interface for easy and automatic transferring of approval or waiver into the de-identification system could play a crucial role. A manual system is time-consuming, inconvenient, and prone to errors. Moreover, high accuracy is attainable and necessary to prevent data falling into wrong hands but also a breach caused by hacking or carelessness of data recipient could happen if re-identification is possible from the previously released de-identified data. Re-identification to protect patient's privacy using the de-identification system.

Third, Since words are the features for textual de-identification then the issue of better methods to develop word embeddings needs to be devised and put in place. The inability to handle unknown or OOV words is another challenge with word embeddings since one is forced to use a random vector every time the model hasn't encountered unknown or out-of-vocabulary (OOV) words due to the fact that there hasn't been a logical way on how to build a vector for these OOV words.

## 8.1 Future Work

This section presents an overview of some possible future work extending the contributions of this thesis.

### 8.1.1 Semi-Supervised Learning

What if the clinical notes de-identification process could be extended into Self-training? Self-training [77, 100, 103, 112] consists of using the model being trained to label the unlabeled training examples. This is achieved in multiple steps. First, the labeled data is used to learn parameters. The resulting model is used to label the unlabeled training examples. Those examples are used to update the model's parameters.

Large amount of unlabeled written clinical text could be obtained through the Healthcare provider repositories. However, such data cannot be used for the supervised learning algorithm used as described herein, but could be utilized by semi-supervised learning algorithms that use a mix of labeled and unlabeled training examples to train models. The process to pre-train the hidden layer of a BiLSTM_CRF could be similar to what was used to pre-train the word representation through utilization

of the unlabeled data to train a deep NN model, whose parameters would be used to initialize the hidden layers of a BiLSTM_CRF. The process can be repeated, depending on the algorithm. This approach requires enough data to train a good initial model, and can be computationally expansive, due to its iterative nature.

Finally, the algorithm presented in [58] could be adapted for BiLSTM_CRF. This algorithm minimizes a loss function composed of a discriminative model and a generative model, both based on NNs. In [58], this algorithm is used to train image classifiers, where the class is scalar.

### 8.1.2 Medical Solutions through Self-Attention Mechanisms

Attention Mechanisms were used primarily in the field of visual imaging, beginning in about the 1990s. The essence of Attention Mechanisms is that they are an imitation of the human sight mechanism. When the human sight mechanism detects an item, it will typically not scan the entire scene end to end; rather it will always focus on a specific portion according to the person's needs. When a person notices that the object they want to pay attention to typically appears in a particular part of a scene, they will learn that in the future, the object will look in that portion and tend to focus their attention on that area.

Attention Mechanisms have found broad application in all kinds of natural language processing (NLP) tasks based on deep learning. In NLP work, the key and value are frequently the same, therefore key=value.
The Multi-Headed Attention Mechanism method uses Multi-Headed self-attention heavily in the encoder and decoder.

**Layer Complexity, Run-ability, Distant dependency learning:**

It is known that if the input sequence n is smaller than the representative dimension d, then Self-Attention is advantageous regarding the time complexity of each layer. When n is larger, In NLP, may be the author provides a solution in Self-Attention (restricted), where not every word undergoes Attention calculation, instead only r words undergo the calculation. Regarding concurrency, Multi-Head Attention and CNN both depend on calculations from the previous instance and features better concurrency than RNN.

It will be interesting to develop a solution that could provide many of the Medical Solutions (from drug development to automated prescription drug dispensing) through Self-Attention Mechanisms that go well beyond de-identification of clinician textual notes? What I am proposing here is not only development of Self-Attention Mechanisms for the tagging of textual notes de-identification tasks but a multi-layered multi-headed solution with information extraction ability to further utilize the notes to provide the timely patient care and solutions.

### 8.1.3 Context-Aware Neural Model for Information Extraction and Privacy Preserving

LSTM-variant NNs have trainable gates to address the vanishing and exploding gradient problem (Hochreiter and Schmidhuber, 1997). At each time step, the NN chooses what to memorize and forget, so patterns over arbitrary time intervals can be recognized. However, the memory in LSTM is still short-term. No matter how long the cell states keep certain information, once it is forgotten, it gets lost forever. Such a mechanism suffices for modeling contiguous sequences. However, when the sequences are not contiguous, as in temporal and other discourse-scale relations, LSTM models do not have inherent

capability to look for input pieces across sequences.

It will be interesting to develop and/or to have a model with a uniform architecture for event-event, event-timex and timex-timex pairs that is robust and context-aware to deal with the NLP challenges in a highly technical and specialized environment like medical field with many layers as inspired by Neural Turing Machine(NTM)— An NTM-like architecture with an external memory and attention mechanisms or a system with long-term memory as well as attention mechanisms to process information.

## 8.2 Conclusion

The field of natural linguistic processing includes a number of instruments which are very helpful for obtaining, labeling and prediction information from raw text data. This collection of methods is used primarily to recognize feelings, text labeling, chat-bots and vocal assistants. Machine learning and artificial intelligence technology in healthcare have become increasingly important in recent years. Deep learning is the recent development in artificial intelligence technology that allows computers to imitate human intelligence in more advanced and autonomous ways.

For EHR the ML component for processing structured data (images, EP data, genetics) and the NLP component for the mining of unstructured texts must have great algorithms in order to be successful AI systems. The advanced algorithms must then be taught through health data before the systems are able to help doctors diagnose diseases and make recommendations for therapy. Early healthcare systems used artificial intelligence systems largely to train computers by encoding medical insight into the logical rules for clinical situations in particular. More sophisticated machine learning systems learn such rules by detecting and evaluating data-related characteristics such as pixels in medical imagery or raw information from electronic health records (EHRs).

The bulk of health information and patient information is currently stored as unstructured clinical document including doctor's notes, medications, transcripts for audio interviews, and reports on pathologies and radiology. Today's identification of such data is a manual, long procedure, requiring either data entry by qualified professionals or developer teams that write custom code and guidelines to automatically retrieve the content. This undifferentiated necessary heavy lifting, in most instances, depletes economic resources from attempts to enhance patient results through technology.

De-identification of medical notes has crucial role to play in unearthing the wealth of clinical information in order to make an enormous leap forward in the areas of ML-aided disease detection, ML-driven triage and avoidance of disorders, in the clustering of illness sub-types, in defect identification, and in an ML-augmented physician deployment.

Through the use of some of the methods that I have outlined in this chapter, I truly believe that the future is very bright and could create many industries of tomorrow that will leverage the provisions of POPIA and HIPAA in order to create the healthcare system that delivers optimum care and advanced medicine for its customers.

# Appendix A

# Python scripts used in the Research

Here is the implementation in python

```
"""
Class for the BiLSTM_CRF model: derived from the BiLSTM class
"""
def __init__(self, vocabs_size, tags_to_ix, embeddings_dim, hiddens_dim):
    super(BiLSTM_CRF, self).__init__()
    self.embeddings_dim = embeddings_dim
    self.hiddens_dim = hiddens_dim
    self.vocabs_size = vocabs_size
    self.tags_to_ix = tags_to_ix
    self.tagsets_size = len(tags_to_ix)

    self.word_embeds = nn.Embedding(vocabs_size, embeddings_dim)
    #Bidirectional has twice the amount of hidden variables so if we
    #want to keep the final output size the same dimension then we
    # have to divide the hidden_dim by 2.
    self.lstm = nn.LSTM(embeddings_dim, hiddens_dim // 2,
                        num_layers=1, bidirectional=True)

    # Graphs the LSTM results to the labels space.
    self.hidden2tag = nn.Linear(hiddens_dim, self.tagsets_size)

    # Matrix of transition parameters.  Entry i,j is the score of
    # transitioning *to* i *from* j.
    self.transitions = nn.Parameter(
        torch.randn(self.tagsets_size, self.tagsets_size))

    # Such two declarations impose the restriction that we never switch
    # to the start tag and we never transfer from the stop tag
    self.transitions.data[tags_to_ix[START_TAG], :] = -1000000
    self.transitions.data[:, tags_to_ix[STOP_TAG]] = -1000000

    self.hidden = self.init_hidden()
```

# Appendix B

# Glossary

A selected few terms and concepts used in this proposal are defined here below.

***Anonymity:*** 'Circumstance or means for identifying an entity as distinctly different, with little or no identifying evidence to make a reference to a documented identity' (ISO/IEC 24760-1:2011)

***Anonymization:*** 'Mechanism which eliminates the link connecting data collection to the information subject matter' (ISO/TS 25237:2008)

***Anonymized data:*** ' Information wherein the data beneficiary could not identify the subject.' (ISO/TS 25237:2008)

***Anonymous identifier:*** 'Identifying attribute of a person who does not permit a biological individual to be unambiguously identified.' (ISO/TS 25237:2008)

***Attacker:*** 'An individual who want to compromise a mechanism through its inherent potential weaknesses'

***Attribute:*** "characteristic or property of an entity that can be used to describe its state, appearance, or other aspect (ISO/IEC 24760-1:2011) [1]. In statistical databases, of which tabulated data is made up of, are field names or columns. Various types of attributes are taken into consideration when dealing with data privacy as pointed out by Ciriani et al. [13]

***Brute force attack:*** in cryptography, an attack that involves trying all possible combinations to find a match

***Coded:*** 1. identifying information (such as name or social security number) that would enable the investigator to readily ascertain the identity of the individual to whom the private information or specimens pertain has been replaced with a number, letter, symbol, or combination thereof (i.e., the code); and 2. a key to decipher the code exists, enabling linkage of the identifying information to the private information or specimens. [94]

***Confidential attributes*** are attributes not in the PII and quasi-attributes category but contain sensitive information, such as disease, diagnosis, salary, HIV status, etc. *Non confidential attributes* are attributes that individuals or entities do not consider sensitive and whose publication does not cause

[1] ISO/IEC 24760-1:2011, Information technology – Security techniques – A framework for identity management – Part 1:Terminology and concepts

disclosure. However, studies have shown that none confidential attributes can still be used to re-identify an individual given auxiliary data, thus making the explicit description of what PII is and is not even more challenging[Narayanan and Shmatikov, 2010].

***Control:*** measure that is modifying risk. Note: controls include any process, policy, device,practice, or other actions which modify risk. (ISO/IEC 27000:2014)

***Covered entity:*** under HIPAA, a health plan, a health care clearinghouse, or a health care provider that electronically transmits protected health information (HIPAA Privacy Rule)

***Data linking:*** matching and combining data from multiple databases (ISO/TS 25237:2008)

***Data privacy*** is the protection of an individual's or an entity's data against unauthorized disclosure [12].

***Data security:*** the safety of data from illegitimate access, use, and alteration, focusing on access control while data privacy focuses on disclosure control [21]

***Data subjects:*** persons to whom data refer (ISO/TS 25237:2008)

***Data use agreement:*** executed agreement between a data provider and a data recipient that specifies the terms under which the data can be used.

***De-identification:*** is a process in which PII attributes are removed or transformed to such an extent that when the data is published, an individual's identity cannot be reconstructed [Ganta et al., 2008; Oganian and Domingoferrer, 2001]. Also defined as general term for any process of removing the association between a set of identifying data and the data subject (ISO/TS 25237-2008)

***De-identified information:*** records that have had enough PII removed or obscured such that the remaining information does not identify an individual and there is no reasonable basis to believe that the information can be used to identify an individual (SP800-122)
***De-personalization***is the process of completely removing any subject-related information from an image, including clinical trial identifiers.

***Direct identifiers:*** see direct identifying data

***Direct identifying data:*** data that directly identifies a single individual (ISO/TS 25237:2008)

***Directly identifying variables:*** a category of data that contains direct identifiers

***Disclosure:*** divulging of, or provision of access to, data (ISO/TS 25237:2008)

***Distinguishable information:*** information that can be used to identify an individual (SP800-122)

***Effectiveness:*** extent to which planned activities are realized and planned results achieved(ISO/IEC 27000:2014)

***Entity:*** item inside or outside an information and communication technology system, such as a person, an organization, a device, a subsystem, or a group of such items that has recognizably distinct existence (ISO/IEC24760-1:2011)

***Family and Educational Records Privacy Act (FERPA):*** the primary law in the United States that governs the privacy of student educational records

***Fair Information Practice Principles (FIPP):*** a set of principles that have been developed world-wide since the 1970s that provide guidance to organizations in the handling of personal data

***Genomic information:*** information based on an individual's genome, such as a sequence of DNA or the results of genetic testing

***Harm:*** any adverse effects that would be experienced by an individual (i.e., that may be socially, physically, or financially damaging) or an organization if the confidentiality of PII were breached (SP800-122)

***Health Insurance Portability and Accountability Act of 1996 (HIPAA):*** the primary law in the United States that governs the privacy of healthcare information

***Healthcare identifier:*** identifier of a person for exclusive use by a healthcare system (ISO/TS 25237:2008)

***HIPAA:*** see Health Insurance Portability and Accountability Act of 1996

***HIPAA Privacy Rule:*** establishes national standards to protect individuals' medical records and other personal health information and applies to health plans, health care clearinghouses, and those health care providers that conduct certain health care transactions electronically (HIPAA Privacy Rule, 45 CFR 160, 162, 164)

***Health Information Technology for Economic and Clinical Health Act (HITECH Act):*** a 2009 law designed to stimulate the adoption of electronic health records (HER) in the United States

***Identifiable person:*** one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity (ISO/TS 25237:2008)

***Identification:*** process of using claimed or observed attributes of an entity to single out the entity among other entities in a set of identities (ISO/TS 25237:2008)

***identified information:*** information that explicitly identifies an individual

***Identifier:*** information used to claim an identity, before a potential corroboration by a corresponding authenticator (ISO/TS 25237:2008)

***Indirect identifier:*** information that can be used to identify an individual through association with other information

***Inference:*** refers to the ability to deduce the identity of a person associated with a set of data through clues contained in that information. This analysis permits determination of the individual's identity based on a combination of facts associated with that person even though specific identifiers have

been removed, like name and social security number (ASTM E1869 [2]

***Inference and reconstruction attacks***are methods of attack in which separate pieces of data are used to derive a conclusion about a subject. The attacker will use a combination of publicly published data and other sources of separate information to reconstruct an individual's identity[Brewster, 1996].

***k-anonymity:*** a technique to release person-specific data such that the ability to link to other information using the quasi-identifier is limited. [120] k-anonymity achieves this through suppression of identifiers and output perturbation.

***l-diversity:*** a refinement to the k-anonymity approach which assures that groups of records specified by the same identifiers have sufficient diversity to prevent inferential disclosure [71]

***Likelihood:*** chance of something happening (ISO/IEC 27000:2014)

***Limited dataset:*** A limited data set is protected health information that excludes the following direct identifiers of the individual or of relatives, employers, or household members of the individual: (i) Names; (ii) Postal address information, other than town or city, State, and zip code; (iii) Telephone numbers; (iv) Fax numbers; (v) Electronic mail addresses; (vi) Social security numbers; (vii) Medical record numbers; (viii) Health plan beneficiary numbers; (ix) Account numbers; (x) Certificate/license numbers; (xi) Vehicle identifiers and serial numbers, including license plate numbers; (xii) Device identifiers and serial numbers; (xiii) Web Universal Resource Locators (URLs); (xiv) Internet Protocol (IP) address numbers; (xv) Biometric identifiers, including finger and voice prints; and (xvi) Full face photographic images and any comparable images. (under §164.514 HIPAA Privacy Rule specifically 45 CFR 164.514 (e) (2)). Limited data sets can contain complete dates, age to the nearest hour, city, state, and complete ZIP code.

***Linkable information:*** information about or related to an individual for which there is a possibility of logical association with other information about the individual (SP800-122)

***Linked information:*** information about or related to an individual that is logically associated with other information about the individual (SP800-122)

***Masking:*** the process of systematically removing a field or replacing it with a value in a way that does not preserve the analytic utility of the value [25], such as replacing a phone number with asterisks or a randomly generated pseudonym [25]
Active ***Moiety*** is a molecule or ion, excluding those appended portions of the molecule that cause the drug to be an ester, salt (including a salt with hydrogen or coordination bonds), or other noncovalent derivative (such as a complex, chelate, or clathrate) of the molecule, responsible for the physiological or pharmacological action of the drug substance.

***New chemical entity (NCE)*** is, according to the U.S. Food and Drug Administration, a drug that contains no active moiety that has been approved by the FDA in any other application submitted under section 505(b) of the Federal Food, Drug, and Cosmetic Act.

***New molecular entity (NME)*** is a drug that contains an active moiety that has never been approved by the FDA or marketed in the US.

[2] ASTM E1869-04 (Reapproved 2014), Standard Guide for Confidentiality, Privacy, Access, and Data Security Principles for Health Information Including Electronic Health Records, ASTM International.

***Non-deterministic noise:*** a random value that cannot be predicted

***Obscured data:*** data that has been distorted by cryptographic or other means to hide information. It is also referred to as being masked or obfuscated. (SP800-122)

***Personal identifier:*** information with the purpose of uniquely identifying a person within a given context (ISO/TS 25237:2008)

***Personal data:*** any information relating to an identified or identifiable natural person (data subject) (ISO/TS 25237:2008)

***Personally Identifiable Information (PII):*** is any information about an individual that could be used to construct the full identity of that individual with or without their authorization; these are attributes that explicitly identify an individual. [U.S. Department of Homeland Security, 2008; Mccallister and Scarfone, 2010]. But can further be defined as Any information about an individual maintained by an agency, including (1) any information that can be used to distinguish or trace an individual's identity, such as name, social security number, date and place of birth, mothers maiden name, or biometric records; and (2) any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information." [3](SP800-122)

***PII attributes***are properties that uniquely identify an individual. Examples include SSN, and combination of first and last name. None explicitly identifying attributes or *quasi-attributes* are attributes not in the PII category but can be used to reconstruct an individual's identity in conjunction with external data.

***POPI Act or POPIA:*** see Protection of Personal Information Act.

***Potentially Identifiable Personal Information (PI 2):*** 'A recent data classification method suggested by Robert Gellman to de-identify but re-identify information. "Any personally identifiable information, without open identifiers, is personal information [38].'

***Privacy:*** 'Free of interference in the personal or person's life if this infringement is the result of unarranted or illegal collection and use of information relating to that person.' (ISO/IEC 2382-8:1998, definition 08-01-23)

***Process:*** 'a series of interdependent practices that converts inputs into outputs.'(ISO/IEC 27000:2014)

***Protection of Personal Information Act:*** The law governing the disclosure of protected health information in South Africa. The Protection of Personal Information Act No. 4 of 2013 encourages the protection and prevent unauthorized disclosure by public and private institutions of personally identifiable information. The South African government ratified the POPIA Act on 19 November and released Notice 37067 on 26 November 2013 in the Government Gazette.

***Pseudonymization:*** ' Specific type of anonymisation, both removing the affiliation with a data subject and associating one or more values with a specific set of traits concerning that data subject.' Unlike pseudonymization, ***anonymization*** The provision does not provide means through which data

[3]GAO Report 08-536, Privacy: Alternatives Exist for Enhancing Protection of Personally Identifiable Information, May 2008, http://www.gao.gov/new.items/d08536.pdf

records or information systems could be linked with the same person. Therefore it is impossible to reidentify anonymised data.(ISO/TS 25237:2008) Pseudonymization is generally carried out by having to replace a pseudonym for a direct identifier, such as a randomized value.

***Pseudonym:*** 'Individual identification which differs from the usual personal identification.'(ISO/TS 25237:2008)

***Recipient:*** 'natural or lawful individual, public authorities, agencies or any other body to which information is communicated' (ISO/TS 25237:2008)

***Re-identification:*** General terminology for any procedure which restores the relationships between identification of data and risk of re-identification of the individual: the risk of reconstituting the identity in the de-identified records. Re - identification risk is generally shown as the propotion of records in a re-identifiable dataset.

***Risk:*** 'Uncertainty effect on targets.Risks are often articulated in the combining the implications of an occurrence and the associated probability of occurrence.' (ISO/IEC 27000:2014)

***Synthetic data generation:*** 'a process wherein the input information is used to artificially generate data with certain statistical features as the informational input seed.'

***Threat:*** 'a possible cause whence emanate the repercussion that could lead to damage to a scheme, mechanism, or institution.' (ISO/IEC 27000:2014)

***Trusted data recipient:*** 'a company with restricted access to the information it obtains because it is obligated by an operational control, such as a pact, a legislation, or rules on information use.'

# Appendix C

# Standards

**ASTM E1869-04(2014)** Medical information including electronic health records modern standard guidebook for patient confidentiality, authorized disclosure mechanisms, and information security practices.

**ISO/IEC 27000:2014** The general standard that encompasses the areas of IT, security methodologies, IT monitoring or surveillance systems, and data security domain overview and vocabulary.

**ISO/IEC 24760-1:2011** The general standard that encompasses the areas of IT, security methods, an identity management mechanisms, and in Part 1: The terminology and paradigms.

**ISO/TS 25237:2008(E)** This policy document by ISO located in Geneva, Switzerland was published in 2008. The policy encompasses the areas of computer health, pseudonymisation. It characterizes how confidentiality confidential information can be de-identified by means of a "pseudonymization service," which substitutes specific identification with aliases. It offers a number of terms and concepts which this ISO technical policy document considers authoritative.

# Bibliography


[1] 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.

[2] Ehsaneddin Asgari and Mohammad RK Mofrad. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS one*, 10(11):e0141287, 2015.

[3] Leonard E Baum and John Alonzo Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73(3):360–363, 1967.

[4] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

[5] Bruce A Beckwith, Rajeshwarri Mahaadevan, Ulysses J Balis, and Frank Kuo. Development and evaluation of an open source software tool for deidentification of pathology reports. *BMC medical informatics and decision making*, 6(1):12, 2006.

[6] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[7] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[8] Kathleen Benitez and Bradley Malin. Evaluating re-identification risks with respect to the hipaa privacy rule. *Journal of the American Medical Informatics Association*, 17(2):169–177, 2010.

[9] David M Blei, Michael I Jordan, et al. Variational inference for dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143, 2006.

[10] 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.

[11] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*, 2015.

[12] Valentina Ciriani, Sabrina De Capitani Di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Fragmentation and encryption to enforce privacy in data storage. In *European Symposium on Research in Computer Security*, pages 171–186. Springer, 2007.

[13] Valentina Ciriani, Sabrina De Capitani Di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Combining fragmentation and encryption to protect privacy in data storage. *ACM Transactions on Information and System Security (TISSEC)*, 13(3):22, 2010.

[14] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[15] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[17] Hamish Cunningham. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254, 2002.

[18] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[19] Azad Dehghan, Aleksandar Kovacevic, George Karystianis, John A Keane, and Goran Nenadic. Combining knowledge-and data-driven methods for de-identification of clinical narratives. *Journal of biomedical informatics*, 58:S53–S59, 2015.

[20] Louise Deleger, Katalin Molnar, Guergana Savova, Fei Xia, Todd Lingren, Qi Li, Keith Marsolo, Anil Jegga, Megan Kaiser, Laura Stoutenborough, et al. Large-scale evaluation of automated clinical note de-identification and its impact on information extraction. *Journal of the American Medical Informatics Association*, 20(1):84–94, 2013.

[21] Dorothy E Denning and Peter J Denning. Data security. *ACM Computing Surveys (CSUR)*, 11(3):227–249, 1979.

[22] Franck Dernoncourt, Ji Young Lee, Özlem Uzuner, and Peter Szolovits. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association: JAMIA*, 24(3):596–606, 2017.

[23] Noa P Cruz Díaz and Manuel Maña López. An analysis of biomedical tokenization: Problems and strategies. In *Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis*, pages 40–49, 2015.

[24] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[25] Khaled El Emam and Luk Arbuckle. *Anonymizing health data: case studies and methods to get you started*. " O'Reilly Media, Inc.", 2013.

[26] Khaled El Emam, Elizabeth Jonker, Luk Arbuckle, and Bradley Malin. A systematic review of re-identification attacks on health data. *PloS one*, 6(12):e28071, 2011.

[27] JD Ferguson. Hidden markov analysis: an introduction. *Hidden Markov Models for Speech*, 1980.

[28] Oscar Ferrández, Brett R South, Shuying Shen, F Jeffrey Friedlin, Matthew H Samore, and Stéphane M Meystre. Bob, a best-of-breed automated text de-identification system for vha clinical documents. *Journal of the American Medical Informatics Association*, 20(1):77–83, 2012.

[29] Gernot A Fink. *Markov models for pattern recognition: from theory to applications*. Springer Science & Business Media, 2014.

[30] Ronald A Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222(594-604):309–368, 1922.

[31] Office for Civil Rights. *Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule*, 2015 (accessed February 14, 2019).

[32] HHS(The Office for Civil Rights (OCR)). Guidance regarding methods for de-identification of protected health information in accordance with the health insurance portability and accountability act (hipaa) privacy rule, Nov 2012.

[33] Dayne Freitag. Machine learning for information extraction in informal domains. *Machine learning*, 39(2-3):169–202, 2000.

[34] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[35] F Jeff Friedlin and Clement J McDonald. A software tool for removing patient identifying information from clinical documents. *Journal of the American Medical Informatics Association*, 15(5):601–610, 2008.

[36] James Gardner and Li Xiong. Hide: an integrated system for health information de-identification. In *Computer-Based Medical Systems, 2008. CBMS'08. 21st IEEE International Symposium on*, pages 254–259. IEEE, 2008.

[37] Roger Garside. The large-scale production of syntactically analysed corpora. *Literary and Linguistic Computing*, 8(1):39–46, 1993.

[38] Robert Gellman. The deidentification dilemma: a legislative and contractual proposal. *Fordham Intell. Prop. Media & Ent. LJ*, 21:33, 2010.

[39] A Golab. Social security data puts 1.3 mil. voters at risk: suit. *Chicago Sun-Times*, page 13, Jan 2007.

[40] Philippe Golle. Revisiting the uniqueness of simple demographics in the us population. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 77–80. ACM, 2006.

[41] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[42] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2009.

[43] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[44] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[45] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, volume 1, 1996.

[46] Yikun Guo, Robert Gaizauskas, Ian Roberts, George Demetriou, Mark Hepple, et al. Identifying personal health information using support vector machines. In *i2b2 workshop on challenges in natural language processing for clinical data*, pages 10–11, 2006.

[47] Kazuo Hara et al. Applying a svm based chunker and a text classifier to the deid challenge. In *i2b2 Workshop on challenges in natural language processing for clinical data*, pages 10–11, 2006.

[48] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[49] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*, 2016.

[50] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14, 2012.

[51] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[52] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[53] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[54] CHEO Research Institute. *What de-identification software tools are there?*, 2018 (accessed December 15, 2018).

[55] Daniel Jurafsky and James H Martin. Automatic speech recognition. *Speech and Language Processing: An Introduction to natural language processing, computational linguistics, and speech recognition*, 2007.

[56] M Kayaalp. *NLM Scrubber: NLM's Software Application to De-identify Clinical Text Documents - Mehmet Kayaalp.*

[57] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[58] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[59] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[60] Stefan Kombrink, Tomáš Mikolov, Martin Karafiát, and Lukáš Burget. Recurrent neural network based language modeling in meeting recognition. In *Twelfth annual conference of the international speech communication association*, 2011.

[61] Taku Kudo and Yuji Matsumoto. A boosting algorithm for classification of semi-structured text. In *EMNLP*, volume 4, pages 301–308, 2004.

[62] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387, 2016.

[63] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Conference Paper University of Pennsylvania*, 2001.

[64] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[65] K Ming Leung. Backpropagation in multilayer perceptrons. *Polytechnic University Department of Computer Science/Finance and Risk Engineering*, 2008.

[66] Liyuan Liu, Jingbo Shang, Frank Xu, Xiang Ren, Huan Gui, Jian Peng, and Jiawei Han. Empower sequence labeling with task-aware neural language model. *arXiv preprint arXiv:1709.04109*, 2017.

[67] Zengjian Liu, Buzhou Tang, Xiaolong Wang, and Qingcai Chen. De-identification of clinical notes via recurrent neural network and conditional random field. *Journal of biomedical informatics*, 75:S34–S42, 2017.

[68] Denise Love, Emily Sullivan, Robert Davis, Alan Prysunka, and Barbara Rudolph. The collection and release practices of physician identifiers in statewide hospital discharge data reporting. 2010.

[69] James O Luke, Mark L Fischer, and Sue-Anne M Sweeney. Method for automatically identifying, matching, and near-matching buyers and sellers in electronic market transactions, October 10 2000. US Patent 6,131,087.

[70] Xuezhe Ma and Eduard Hovy. Efficient inner-to-outer greedy algorithm for higher-order labeled dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1322–1328, 2015.

[71] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[72] Christopher D Manning. Prabhakar raghavan, and hinrich schutze. *Introduction to information retrieval*, 2008.

[73] Christopher D Manning, Christopher D Manning, and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[74] Ian Marshall. Choice of grammatical word-class without global syntactic analysis: tagging words in the lob corpus. *Computers and the Humanities*, 17(3):139–150, 1983.

[75] A Maynord. New details reveal numerous mistakes prior to election commission break-in. *Nashville City Paper*, Jan 2008.

[76] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. *http://mallet.cs.umass.edu*, 2002.

[77] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics*, pages 152–159. Association for Computational Linguistics, 2006.

[78] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2391–2400. JMLR. org, 2017.

[79] Stéphane M Meystre, Óscar Ferrández, F Jeffrey Friedlin, Brett R South, Shuying Shen, and Matthew H Samore. Text de-identification for privacy protection: A study of its impact on clinical text information content. *Journal of biomedical informatics*, 50:142–150, 2014.

[80] Stephane M Meystre, F Jeffrey Friedlin, Brett R South, Shuying Shen, and Matthew H Samore. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC medical research methodology*, 10(1):70, 2010.

[81] Stéphane M Meystre, Guergana K Savova, Karin C Kipper-Schuler, John F Hurdle, et al. Extracting information from textual documents in the electronic health record: a review of recent research. *Yearb Med Inform*, 35(128):44, 2008.

[82] D Michie, M Nuñez, V Podgorelec, P Kokol, B Stiglic, and I Rozman. Quinlan, jr c4. 5: Programs for machine learning, 1993.

[83] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[84] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[85] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[86] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013.

[87] Michael C Mozer. A focused backpropagation algorithm for temporal. *Backpropagation: Theory, architectures, and applications*, 137, 1995.

[88] Daša Munková, Michal Munk, and Martin Vozár. Data pre-processing evaluation for text mining: transaction/sequence model. *Procedia Computer Science*, 18:1198–1207, 2013.

[89] Harvey J Murff, Fern FitzHenry, Michael E Matheny, Nancy Gentry, Kristen L Kotter, Kimberly Crimin, Robert S Dittus, Amy K Rosen, Peter L Elkin, Steven H Brown, et al. Automated identification of postoperative complications within an electronic medical record using natural language processing. *Jama*, 306(8):848–855, 2011.

[90] Ishna Neamatullah, Margaret M Douglass, H Lehman Li-wei, 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):32, 2008.

[91] Patrick Ng. dna2vec: Consistent vector representations of variable-length k-mers. *arXiv preprint arXiv:1701.06279*, 2017.

[92] Office of the Federal Register (OFR). *45 C.F.R §164.514*. The US Department of Health and Human Services(HHS), 2002.

[93] Philip V Ogren. Knowtator: a protégé plug-in for annotated corpus construction. In *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume: demonstrations*, pages 273–275. Association for Computational Linguistics, 2006.

[94] US OHRP. Guidance on research involving coded private information or biological specimens [z]. *Washington, DC: HHS*, 2008.

[95] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

[96] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[97] Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108*, 2017.

[98] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[99] Paul Rayson and Roger Garside. Comparing corpora using frequency profiling. In *Proceedings of the workshop on Comparing Corpora*, pages 1–6. Association for Computational Linguistics, 2000.

[100] Roi Reichart and Ari Rappoport. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 616–623, 2007.

[101] Nils Reimers and Iryna Gurevych. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799*, 2017.

[102] Raúl Rojas. *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.

[103] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *WACV/MOTION*, pages 29–36, 2005.

[104] Sebastian Ruder. On word embeddings - part 1, Oct 2018.

[105] Haşim Sak, Andrew Senior, Kanishka Rao, Ozan Irsoy, Alex Graves, Françoise Beaufays, and Johan Schalkwyk. Learning acoustic frame labeling for speech recognition with recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4280–4284. IEEE, 2015.

[106] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.

[107] Fabrizio Sebastiani. Classification of text, automatic. *The encyclopedia of language and linguistics*, 14:457–462, 2006.

[108] Richard Socher, John Bauer, Christopher D Manning, et al. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 455–465, 2013.

[109] South African Parliament. No. 4 of 2013: Protection of personal information act., 2013. https://www.gov.za/sites/default/files/gcis_document/201409/3706726-11act4of2013protectionofpersonalinforcorrect.pdf,.

[110] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[111] Stanford.edu. Convolutional neural networks for visual recognition course(cs231n), Oct 2018.

[112] Mark Steedman, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. Bootstrapping statistical parsers from small datasets. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 331–338. Association for Computational Linguistics, 2003.

[113] Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. Fast and accurate entity recognition with iterated dilated convolutions. *arXiv preprint arXiv:1702.02098*, 2017.

[114] Amber Stubbs, Michele Filannino, and Özlem Uzuner. De-identification of psychiatric intake records: Overview of 2016 cegs n-grid shared tasks track 1. *Journal of biomedical informatics*, 75:S4–S18, 2017.

[115] 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.

[116] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

[117] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[118] Charles Sutton and Andrew McCallum. *An introduction to conditional random fields for relational learning*, volume 2. Introduction to statistical relational learning. MIT Press, 2006.

[119] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[120] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[121] György Szarvas, Richárd Farkas, and Róbert Busa-Fekete. State-of-the-art anonymization of medical records using an iterative machine learning framework. *Journal of the American Medical Informatics Association*, 14(5):574–580, 2007.

[122] Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 41–47. ACM, 2003.

[123] TensorFlow. Vector representations of words — tensorflow.

[124] Dozat Timothy. Incorporating nesterov momentum into adam, 2015.

[125] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[126] Özlem Uzuner, Yuan Luo, and Peter Szolovits. Evaluating the state-of-the-art in automatic de-identification. *Journal of the American Medical Informatics Association*, 14(5):550–563, 2007.

[127] Ben Wellner, Matt Huyck, Scott Mardis, John Aberdeen, Alex Morgan, Leonid Peshkin, Alex Yeh, Janet Hitzeman, and Lynette Hirschman. Rapidly retargetable approaches to de-identification in medical records. *Journal of the American Medical Informatics Association*, 14(5):564–573, 2007.

[128] Leon Willenborg and Ton De Waal. *Elements of statistical disclosure control*, volume 155. Springer Science & Business Media, 2012.

[129] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

[130] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[131] Donghuo Zeng, Chengjie Sun, Lei Lin, and Bingquan Liu. Lstm-crf for drug-named entity recognition. *Entropy*, 19(6):283, 2017.