IDENTIFYING TEXTUAL PERSONAL INFORMATION WITH ARTIFICIAL NEURAL NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEMDUH ÇAĞRI DEMIR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

AUGUST 2019

Approval of the thesis:

IDENTIFYING TEXTUAL PERSONAL INFORMATION WITH ARTIFICIAL NEURAL NETWORKS

submitted by **MEMDUH ÇAĞRI DEMIR** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** ______________

Assist. Prof. Dr. Şeyda Ertekin
Supervisor, **Computer Engineering, METU** ______________

Examining Committee Members:

Prof. Dr. Ferda Nur Alpaslan
Computer Engineering, METU ______________

Assist. Prof. Dr. Şeyda Ertekin
Computer Engineering, METU ______________

Prof. Dr. Suat Özdemir
Computer Engineering, Gazi University ______________

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Memduh Çağrı Demir

Signature :

ABSTRACT

IDENTIFYING TEXTUAL PERSONAL INFORMATION WITH ARTIFICIAL NEURAL NETWORKS

Demir, Memduh Çağrı
M.S., Department of Computer Engineering
Supervisor: Assist. Prof. Dr. Şeyda Ertekin

August 2019, 37 pages

In this thesis, a two-layered neural network model named Anonimatik is described. Anonimatik model is developed to automatically detect personal information holding words in plain text that does not contain any metadata. Unlike traditional automatic de-identification systems, Anonimatik neural network model is designed to be used without requiring external knowledge resources (e.g. name lists, medical term lists, common word lists etc.) with the aim of making the model applicable to different languages. Anonimatik model classifies words by processing the local context instead of relying on the dictionary definitions. Anonimatik takes nine-word sequences as input and outputs the classification stating whether the word in the middle of the sequence contains personal information or not. In the first layer of the proposed model, a bi-directional long short term memory (Bi-LSTM) network encodes the local context of the target word. Then in the second layer, a fully connected deep neural network classifies the target word. The model proposed in this thesis is evaluated on the dataset created for the automated de-identification challenge organized by Informatics for Integrating Biology & the Bedside (i2b2) in 2006 and it is shown that proposed model produces comparable results to other teams participated into the challenge. Anonimatik neural model is the only publicly available automated de-identification model that is designed to be used without requiring any external dictionaries.

Keywords: de-identification of plain texts, text classification, local contexts of words, long short term memory networks

ÖZ

METİN İÇİNDEKİ KİŞİSEL BİLGİLERİN YAPAY SİNİR AĞLARI KULLANILARAK TESPİTİ

Demir, Memduh Çağrı
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Dr. Öğr. Üyesi. Şeyda Ertekin

Ağustos 2019 , 37 sayfa

Bu tezde Anonimatik adı verilen çift katmanlı bir yapay sinir ağı modeli tarif edilmiştir. Anonimatik modeli serbest metin şeklinde hazırlanmış hiç üst bilgi içermeyen metinlerin içindeki kişisel bilgi içeren kelimeleri otomatik olarak tespit etmek için geliştirilmiştir. Geleneksel metin anonimleştirme yöntemlerinden farklı olarak, Anonimatik modeli anonimleştirme çalışmaları için hazırlanmış kelime listeleri (özel isim listeleri, tıbbi terim listeleri, genel kelimeler listeleri vb.) kullanmadan çalışmak üzere tasarlanmıştır. Bu yaklaşımla Anonimatik modelinin farklı dillere uygulanabilir olması amaçlanmıştır. Anonimatik modeli cümle içindeki kelimelerin sınıflandırmasını yaparken hedef kelimenin sözlük anlamını kullanmak yerine cümle içindeki anlamını çözümleyerek sınıflandırma yapmaktadır. Model girdi olarak dokuz kelimelik bir dizi alıp, çıktı olarak dizinin ortasındaki kelimenin kişisel bilgi içerip içermediği sınıflandırmasını yapmaktadır. Tasarlanan modelin ilk katmanında kullanılan çift yönlü uzun/kısa dönem bellek ağı (Bi-LSTM) hedef kelimenin cümle içindeki anlamını bir vektör uzayında kodlarken, ikinci katmanda kullanılan derin ileri beslemeli yapay sinir ağı kelimenin sınıflandırmasını yapmaktadır. Oluşturulan model 2006 yılında

Informatics for Integrating Biology & the Bedside (i2b2) tarafından düzenlenen otomatik anonimleştirme yarışması kapsamında oluşturulmuş derlem üzerinde denenmiş ve yarışmaya katılan diğer takımların elde ettiği sonuçlar ile kıyaslanabilir bir sonuç elde edilmiştir. Anonimatik modeli, hiçbir ekstra sözlük kullanmadan çalışmak üzere tasarlanan ve tüm araştırmacıların kullanabilmesi için açık kaynak kodlu olarak yayınlanan ilk ve tek otomatik metin anonimleştirme modelidir.

Anahtar Kelimeler: serbest metin içinde kişisel bilgi tespiti, metin sınıflandırması, kelimelerin yerel anlamı, uzun kısa dönem hafıza ağları

This thesis is dedicated to my family.

ACKNOWLEDGMENTS

Foremost, I would like to thank to my thesis advisor Assist. Prof. Dr. Şeyda Ertekin for her immense knowledge and support. She was always there to help me throughout this study and guided me to completion of this thesis. Besides my advisor, I would like to thank my examining committee members Prof. Dr. Ferda Nur Alpaslan and Prof. Dr. Suat Özdemir for their valuable feedback.

I would like to acknowledge Informatics for Integrating Biology & the Bedside (i2b2) for preparing the dataset used in this thesis. All annotated medical discharge summaries used in this research were provided by the i2b2 National Center for Biomedical Computing funded by U54LM008748 and were originally prepared for the Shared Tasks for Challenges in NLP for Clinical Data organized by Dr. Özlem Uzuner, i2b2 and SUNY.

I thank Erenay Dayanık for proofreading this thesis and for his valuable help on this study.

I am deeply grateful to my parents for their love and endless support throughout this study. None of this would be possible without their encouragement.

I kept the biggest "thank you" to the end. I would like to thank my beloved wife Gökçe a million times. Without the support and motivation she provided, I would never be able to complete this thesis.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

Bi-LSTM	Bi-directional Long Short-Term Memory
CRF	Conditional Random Fields
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
IP	Internet Protocol
KVKK	Kişisel Verileri Koruma Kanunu
LSTM	Long Short-Term Memory
PHI	Protected Health Information
RNN	Recurrent Neural Network
SVM	Support Vector Machine
URL	Uniform Resource Locator

CHAPTER 1

INTRODUCTION

In this thesis, I present Anonimatik, a neural network model that takes a sequence of words from the sentence with their corresponding part of speech (POST) tags and produces a binary result stating whether the word at the middle of the sequence contains personal information or not. In other words, assuming the input sequence contains nine words from w_1 to w_9, Anonimatik's result classifies w_5 as PHI word or non-PHI word. Once trained, Anonimatik model can be used without requiring any additional knowledge resources unlike most traditional automated de-identification methods. As explained in later chapters, many of the similar applications depend on predefined dictionaries of special words such as lists of cities or lists of names. In contrast, Anonimatik model presented in this thesis is designed to be used without using any dictionaries or knowledge resources.

1.1 Motivation and Problem Definition

De-identification is the task of clearing any information that may connect the data to the related people without compromising the integrity of the data. Data in question can be in any format, but in this thesis I focused on the textual data generated in the hospitals called medical discharge summaries. In medical domain, the information that needs to be removed from patient records is specified by the United States Department of Health & Human Services. According to the new regulations introduced in 1996 titled The Health Insurance Portability and Accountability Act (HIPAA) of 1996, words which contain information on any of the categories below considered as protected health information (PHI). [1]

- All kinds of names including people, hospitals etc.
- All geographic location identifiers more specific than a state, including city names, street names, post codes, municipality names and geographic coordinates.
- All kinds of dates excluding the year for the dates related to people, including the dates of birth, death, admission to hospital, discharge from hospital
- Phone numbers of people and institutions
- Vehicle plates, vehicle identifiers
- Fax numbers
- Device identifiers and serial numbers
- Email addresses
- Webpage links
- Social security identifiers
- IP addresses
- Medical record identifiers
- Biometric identifiers such as fingerprints
- Health insurance identifiers
- Photographs where people's face are visible and distinguishable and all similar sketches
- Account identifiers
- Any other unique identifying number, characteristic, or code
- Certificate/license identifiers

In plain text format, detecting numbers that complies to strict rules (such as social security numbers, IP addresses) is rather easy to identify since they do not contain ambiguity. As a result, many researchers used simple rule based methods and regular expressions to detect numerical personal health information as explained in the Chapter 2 of this thesis.

For an automated de-identification process, it is crucial to detect personal information holding words. This is a particularly difficult task for traditional algorithms since meaning of the word may differ in different contexts. An instance of this is shown in Table 1.1 which shows multiple different usages of the word "Hunter". As shown in the table, same word is used as a common noun and as a proper name. This ambiguity makes it difficult to differentiate words containing personal information from others. Failures in identification stage may result in either loss of necessary information (e.g. disease names) or infringement of personal privacy which defeats the purpose of de-identification.

Table 1.1: Different usages of an example word "Hunter"

Word	Usage	Example
hunter	Common Noun	She is a bargain hunter.
hunter	Common Noun	He is an avid hunter.
hunter	Proper Noun	Hunter S. Thompson
hunter	Proper Noun	Hunter Mountain Airport
hunter	Proper Noun	Hunter College, CUNY, 695 PA

The ambiguity shown in Table 1.1 must be handled in the deciding stage whether the word contains personal information or not. In order to deal with the ambiguity, automated de-identification systems must consider the neighbouring words around target word.

In this work, my primary motivation was to create an automated neural network model which is able to detect textual private health information in plain text data without using external dictionaries and language specific rules. The model I proposed in this thesis is composed of 2 neural networks. First neural component of my model is a special type of long-short term memory networks which has the ability to process the input in both ways hence named bi-directional network (Bi-LSTM) that is used to understand the local context of the word [2]. This network component is essentially two LSTM networks together in opposite directions [3]. Using this neural component, my automated de-identification system can eliminate the ambiguity mentioned at the beginning of this chapter. Second neural component of Anonimatik model is a fully-connected deep neural network which has 2 processing layers and a softmax activation at the end to output the probabilities of target word being a personal health information containing word. Whole model is described in detail in Chapter 3 of this document.

1.2 Proposed Methods and Models

There are several works and proposed methods for the automatic de-identification problem. Specifically those attended to 2006 de-identification challenge achieved good results using the dataset specifically prepared for the challenge. These models are described in detail in the Chapter 2 of my thesis.

In order to briefly summarize proposed methods, they mostly depend on pre-defined knowledge resources such as name lists, hospital lists, medical term lists and mostly not available for public use. [4]

1.3 Contributions and Novelties

My contributions to this field of study with the thesis are outlined below:

- Created a publicly available automatic de-identification tool for both research and public use.
- Demonstrated that it is possible to develop an automatic de-identification system without using any predefined knowledge resources.
- Developed an automatic de-identification model which is easily adaptable to different languages.
- Compared and commented on the performance of same model on two different data distributions.

1.4 Outline of the Thesis

Next chapters of this document are summarized below:

- Chapter 2 establishes the terminology on automatic de-identification and summarizes the literature on automated de-identification studies.
- Chapter 3 describes the details of Anonimatik model's input/output, explains the neural components that composes the model and lists the details of network parameters.
- Chapter 4 illustrates evaluation dataset in detail and shows the results Anonimatik model had on the evaluation dataset.
- Chapter 5 concludes my thesis by briefly summarizing my work.

CHAPTER 2

RELATED WORK

In this chapter, I establish terminology on de-identification in medical field and electronic health records, then summarize previous works on de-identification of medical discharge summaries.

2.1 De-identification in Medical Field

There are several regulations introduced to protect the personal health information of patients throughout the world. In United States any researcher or research team who wants to use medical information of patients must get the informed written consent of patient in addition to approval of the Internal Review Board according to the Health Insurance Portability and Accountability Act (HIPAA). But this obligation can be waived if the data is de-identified which enables research teams to use the data stored in hospitals. [1]

Similar to United States, European Union also introduced a law to protect its citizens data. The law titled General Data Protection Regulations (GDPR) is in effect since May 2018 and it applies to all types of information provided by European Union citizens including their medical records. Unlike HIPAA coverage that only applies to personal data stored in the United States itself, GDPR regulations apply to personal data stored inside and outside of the European Union [5].

Similar to those regulations above, Turkey has a similar law that is in effect since 2016. “6698 Sayılı Kişisel Verilerin Korunması Kanunu” states that patient’s data must be de-identified before using it for research purposes. [6] [7] [8]

Although these regulations are necessary to protect patients' privacy, it makes it difficult to use patients' data for research purposes since manual de-identification is a time consuming task.

Having the motivation to ease the burden of future researchers working with patients' data, researchers throughout the world tried to develop an automatic de-identification tool. Sharing the same concern, Partners Healthcare -a Boston-based non-profit hospital and physicians network- organized an event to challenge all automatic de-identification methods and published their results with their implementation details. All of the automatic de-identification studies focused on narrative text are done after this challenge. Table 2.1 shows the details of automatic de-identification systems that focus on the medical discharge summaries.

Table 2.1: Automatic de-identification systems with focus on medical discharge summaries

Primary Author	Availability/License	Knowledge resources
Aramaki [9]	Not open for public use	Name list, location list, important dates list
Guo [10]	Not open for public use	Location list, hospital list
Hara [11]	Not open for public use	None
Szarvas [12]	Not open for public use	Name list, location list, disease name list and general English words list
Uzuner [13]	Not open for public use	MeSH term list, name list, location list and hospital list
Wellner [14]	Open source (BSD)	US states list, month names list and popular words list

As outlined in the Table 2.1, five of the six automatic de-identification systems use predefined dictionaries such as lists of names, lists of hospitals or lists of common English words. These knowledge resources make the system language dependent and cause the system to depend on those lists. It is difficult to adapt these systems into different languages because of the knowledge resource dependency.

In addition to this drawback, only one of those systems is open sourced. Other systems are closed-source and it is impossible for other researchers to use those systems in their studies which defeats the purpose of automatic de-identification studies.

First system designed by Aramaki et al. is one of the top performers of the 2006 de-identification challenge organized by institute of Informatics for Integrating Biology to Bedside (i2b2). This system approaches de-identification problem by processing global features and local features together. The system uses lexical features such as capitalization, word length, sentence position in the document and sentence length in addition to part-of-speech tags of the surrounding words and dictionary terms. The system uses conditional random fields (CRF) to correctly label words as PHI or non-PHI. The system uses two passes of CRF. First pass is used to identify local and global features except label consistency. The idea behind label consistency is, if a personal information holding word or phrase occurs more than one time inside the document, it probably shares the same label with the previous ones. Second pass is used to achieve label consistency. In the 2006 de-identification competition, this system performed better than the average score. For overall personal information holding word detection, this model placed third and scored greater than 94% in precision, recall and F-measure scores. [9]

Another system participated into de-identification challenge is designed by Guo et al. This system approaches de-identification as a named-entity recognition (NER) task and classifies named entities into PHI categories. It uses capitalizations, prefixes/-suffixes, word lengths, regular expressions, part of speech tags and lists of doctors, hospitals and locations. They use Support Vector Machines (SVM) and GATE open source test processing tool with ANNIE natural language processing framework. ANNIE itself is capable of identifying entity types to words and phrases such as person names and dates. According to authors ANNIE's entity recognition had to modified since entity types defined in ANNIE does not map to PHI categories. For overall PHI detection, this system performed below average in i2b2 de-identification challenge and scored better results than 86% for precision, recall and F-measure scores. [10]

Unlike these systems explained in this section, the model designed by Hara and others. does not use any knowledge resources. Their system uses capitalizations, regular expressions, part of speech tags of words and sections headings. This system uses Support Vector Machines (SVM) and uses 4 steps to detect personal information containing words. In the first phase, system applies patterns matching to identify headings in the reports. Secondly, system uses regular expression rules to find patterns like dates and phone numbers. In the third phase, a sentence detector that detects sentences containing personal information holding words. In the last stage, a SVM based text-chunker that identifies words with PHI information. The results showed this system achieved average results in the challenge. For overall personal information holding words detection this system scored better results than 92% for precision, recall and F-measure scores. [11]

Forth system to mention here is Svarvas's de-identification system. This system uses word lengths, capitalizations, regular expressions, term frequencies and part of speech tags. Like majority of these systems, this system uses knowledge resources in addition to these features. System uses lists of person names, states of United States, countries around the world, popular cities and general terms. The general approach of this system is to detect named entities in discharge records and use a continuous learning approach in conjunction with decision trees on top of the result. Svarvas's approach also utilizes regular expression based rules to detect common patterns of all PHI categories. In the de-identification competition, Svarvas's system scored better results than 96% for precision, recall and F-measure scores for overall personal information holding words detection. [12]

Another automatic de-identification system is Stat De-id and designed by Uzuner and others. This system uses capitalization, punctuation, word lengths, part of speech tags and lists of names, US states, hospital names as knowledge resources. This system approaches de-identification as a multi-class labelling task. It tries to determine PHI holding words by processing local context of the word. The system examines the surrounding ±2 words of the target word. It uses support vector machines (SVM) and Link Grammar Parser. On the same challenge dataset, Stat De-id scored 98% F-measure score with a precision score of 99% and a recall score of 97%. [13]

The last system participated into i2b2 de-identification challenge is designed by Wellner and others. It uses capitalizations, prefixes/suffixes, regular expressions and knowledge resources for US states and common English words. This system approached automatic de-identification problem as if it is a sequence-classification problem where classes are assigned to words indicating if the any word is the start of a personal information holding phrase, end of a personal information holding phrase or contained in a personal information holding phrase. The system had the best results in the i2b2 de-identification challenge and scored greater results than 96% as precision, recall and F-measure scores. [14]

The literature review for automatic de-identification of medical discharge records shows that all of these studies use word capitalization as a feature, a great majority of these systems use external knowledge resources and most of them are not publicly available. In fact, there is no publicly available system that does not use external resources to identify personal health information containing words. This is our main motivation for developing the automatic de-identification system we named 'Anonimatik'. Unlike these systems mentioned above, our proposed system does not use external knowledge resources and it is completely open source.

CHAPTER 3

MODEL

In this chapter of my thesis, I demonstrate the layout of my model. First I briefly explain long short term memory networks for the sake of completeness and explain why I preferred to use bi-directional long short term memory networks. Then, I describe input/output structure with examples. Next, I describe the details of neural components that composes Anonimatik model. Lastly, I explain how I assemble those neural components to build the complete model for automated de-identification.

3.1 Long Short Term Memory (LSTM) Networks

Long short-term memory network (LSTM) is a special type of artificial recurrent neural network architecture used in the field of deep learning [3]. Unlike traditional feed-forward neural network models, LSTM networks contain feedback connections which makes them comparable to Turing machines since they both can make same computations [15]. The feedback connection acts as a memory for this type of networks. Thanks to this artificial memory, LSTM networks can process sequences of data points unlike feed forward neural networks which can only process single data points. This feature of LSTM networks makes them perfect for our de-identification task. In general, LSTM networks are suitable for various tasks as Bloomberg Business Week wrote: "These powers make LSTM arguably the most commercial AI achievement, used for everything from predicting diseases to composing music." [16]

A common LSTM unit is composed of four parts. First one is the cell which includes a hidden state (31) (32), second part is the input connection (33), third part is the output connection (34) and last one is a forget gate (35). The cell in first part of

LSTM unit stores values over and rest of the unit including three gates mentioned above regulate the information flow through the unit. The workflow is summarized with the equations below.

$$c_s = f_s \cdot c_{s-1} + i_s + \sigma_c\big(W_c x_s + U_c h_{s-1} + b_c\big) \tag{31}$$

$$h_s = o_s \cdot \sigma_h\big(c_s\big) \tag{32}$$

$$i_s = \sigma_g\big(W_i x_s + U_i h_{s-1} + i_f\big) \tag{33}$$

$$o_s = \sigma_g\big(W_o x_s + U_o h_{s-1} + o_f\big) \tag{34}$$

$$f_s = \sigma_g\big(W_f x_s + U_f h_{s-1} + b_f\big) \tag{35}$$

In the above equations matrix W_q contains the weights of the input and matrix U_q contains the weights of the recurrent connections. Initial values of c_0 and h_0 are equal to zero.

Before creating the model, I checked possible alternative neural network types for sequence labelling problems. In addition to bi-directional LSTM networks, other most promising alternatives were unidirectional LSTM networks, bi-directional recurrent neural networks (RNN) and uni-directional RNNs. After searching the literature, I decided that bi-directional long short term memory network is the best possible solution since this model needs to process the context of the words in both directions and needs to store the state information for a long time to process the long sentences. [17] [18]

Bi-LSTM network I have used in thesis is slightly modified version of the LSTM networks. The difference is, Bi-LSTM networks computes outputs in two directions both forward and backward rather than computing only for forward direction.

3.2 Input/Output Representation

My model takes vectors of nine word sequence S = $[w_1, \ldots, w_9]$ as an input, which has target word in the middle of the input. Each vector is composed of one-hot representation of the word, one-hot encoded part of speech tag of the token and length of the token.

Input representation is visualized in Figure 3.1.

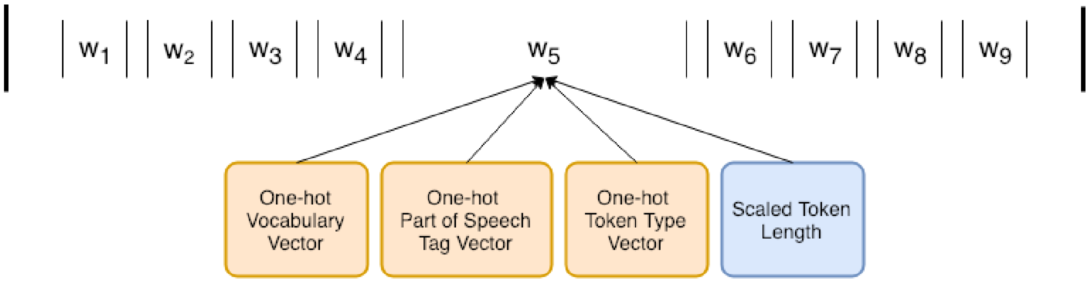
w1
w2
w3
w4
w5
w6
w7
w8
w9
One-hot Vocabulary Vector
One-hot Part of Speech Tag Vector
One-hot Token Type Vector
Scaled Token Length

Figure 3.1: Input representation.

The output of each sequence is the probabilities of output classes which indicate whether the target word in the middle is a PHI word or a non-PHI word. Output is generated by using a softmax function at the end of fully connected neural network.

3.3 Neural Components

In this section of my thesis, I define the neural layers of my model separately. Then, I present whole model in detail and explain the network parameters in detail.

3.3.1 Context Encoder

In our model, context encoder is the first layer and it is a bi-directional LSTM network. Inputs for context encoder is generated by the pre-process step and contains sequences of nine word vectors S = (w1, . . . , w9) as described in the earlier section of this chapter. Context encoder in the first layer for my model processes the inputs in both directions (forward and backward) and generates a particular embedding vector for each token in the sequence. Then, we use the unique embeddings generated after the target word in both directions since this model predicts only for the word in the middle. For a word vector $\vec{w}_t$, context embedding $\vec{c}_i$ is created by concatenating embeddings created on forward and backward passes.

$$\vec{c}_i^f = \mathbf{LSTM}_f(w_i, \vec{c}_{(i-1)}^f) \tag{36}$$

$$\vec{c}_i^b = \mathbf{LSTM}_b(w_i, \vec{c}_{(i+1)}^b) \tag{37}$$

$$\vec{c}_0^f = \vec{c}_9^b = 0 \tag{38}$$

$$\vec{c}_i = [\vec{c}_i^f; \vec{c}_i^b] \tag{39}$$

Since words can be used in different meanings in different contexts, this is the most important component of my model. Figure 3.1 shows an example sentence and visualizes how context encoder explained in this section works.

3.3.2 Decoder

Decoder of my model is designed as a one layer fully connected neural network which calculates the probabilities for whether the word in question is a PHI word or a non-PHI word. Concatenated output of context encoder is used as an input for decoder and softmax activation is applied to the output of decoder layer so that final result can be used as probabilities directly.

$$\vec{y}_i = \vec{c}_i * \vec{w}^{512} + \vec{b} \tag{310}$$

$$\vec{y} = softmax(\vec{y}_i) \tag{311}$$

3.4 Unified Model

Our model is based on the hypothesis that understanding the context of a word is more important than knowing the meaning of actual word. In order to process the context target word is used in, I have used a context encoder. In the first step, a sequence composed of nine words is processed and feature vectors get created. Then, using these 9-vector sequences context encoder creates a unique representation of the context that target word used in. In the final step, a decoder neural network at the end of neural pipeline processes this representation and outputs probabilities of target word being a PHI or non-PHI word.

$$L(t) = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{312}$$

In the training stage, calculated probabilities are used in cross entropy loss to calculate the loss after each batch. Then using an adaptive learning rate with Adam optimizer network updates its weights to improve the accuracy.

Figure 3.2 below demonstrates the execution of Anonimatik model with a sample sentence.

3.5 Tuning Network Parameters

As explained in previous sections, it is crucial to detect the context of the target word for de-identification tasks. Anonimatik model depends on the window size parameter to process the context of the target word and hence it is the most important part of this network. In simple terms, window size parameter is the number of words sent into the model. With the window size of n, the network would be able to process $(n-1)/2$ words before the target word and $(n-1)/2$ words after the target word. Therefore, window size must always be an odd number.

During the development of my model, I experimented with different window sizes ranging from 5 to 13. Performance metrics of my network got better until 9 word-window and started to decrease after that. The performance metrics showed that

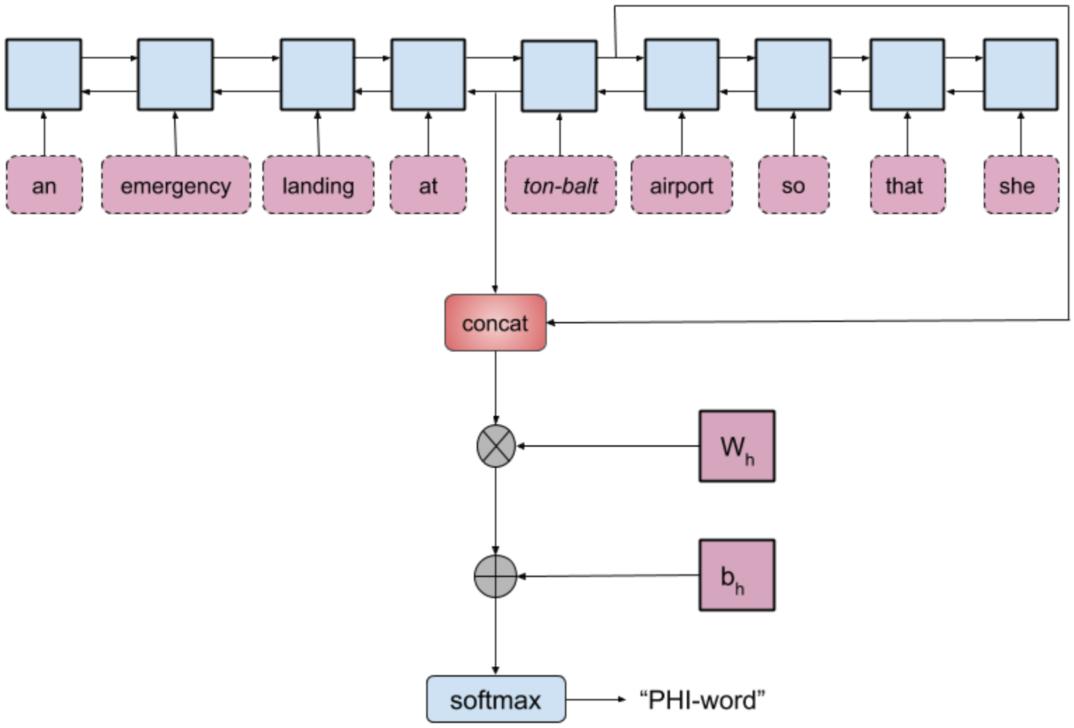

Figure 3.2: Complete neural network with an example input/output.

after 9 words, my network tends to overfit and obtains worse results than 9-words window size. This observation shows that local context of words are encoded in 9 word sequences in English language.

Another network parameter to tune is the number of hidden nodes on the first and on the second layer of the network. For my model, I experimented with different numbers of hidden nodes and obtained the optimal results on 256 hidden nodes. Considering that hidden nodes in the second layer are connected to the hidden nodes in the first layer, increasing number of hidden nodes causes the network to perform slower than before. There were no improvements on the accuracy metrics after reaching to convergence for more than 256 nodes but performance difference was noticeable. As a result, 256 hidden nodes is preferred for this network.

Finally, after experimenting with a wide range of learning rates using classical stochastic gradient descent, I got the best results using a adaptive learning rate of Adam optimizer starting with 0.001 as learning rate. [19]

The unified model explained in this chapter is implemented using Tensorflow open-source machine learning library [20] and visualized using Tensorboard open-source visualization library [21]. Tensorflow provides the methods to create network graphs, activation functions, loss functions, training functions and lot of mathematical operators on the matrices used in this thesis and TensorBoard provides tools to visualize the metrics of the Tensorflow network. In order to make reproducibility easier for future works, all pipeline is implemented using Docker containers. [22]

CHAPTER 4

EXPERIMENTS AND RESULTS

This chapter of my thesis outlines the dataset I used and explains the results of experiments with Anonimatik model. First, I explain the dataset I used to create and evaluate my model in detail. Then, I explain the network training process with some charts to give some insights. Finally, I show my model's results on the evaluation dataset and comment on the results.

4.1 De-identification Challenge

In 2006, a de-identification challenge is organised by Informatics for Integrating Biology and the Bedside (i2b2) which is a national institute of Health-funded National Center for Biomedical Computing based within Partners Healthcare. [23]

The i2b2 Foundation is a Boston-based non-profit organization aimed to create an open-data community to achieve better collaboration for precision medicine. For that purpose, i2b2 shares their resources and analysis of all data from medical and academic institutions.

The automatic de-identification challenge is aimed at finding the best algorithm to automatically de-identify medical discharge records. Several teams participated into the challenge. Although the best scoring teams achieved considerably good results using rule based methods, identifying ambiguous PHI proved challenging for all teams. [24] Those submissions are reviewed in detail in the Chapter 2 of this document.

4.2 Dataset

Dataset used in the challenge is prepared by Partners Healthcare and it consists of only medical discharge summaries written in plain text format. Partners Healthcare researchers changed personal information holding words (PHI-words) to realistic surrogates and annotated those surrogates. In addition to replacing original PHI-words with their realistic surrogates, annotators also added extra ambiguity to corpus by replacing surrogate names with medical terms. As a result, additional ambiguities are created among PHI tokens and non-PHI tokens within the corpus.

Table 4.1: Number of instances in the corpus grouped by their PHI-type.

PHI Categories	Complete Corpus	Training Set	Test Set
Non-PHI	444.127	310.504	133.623
Patient Name	1.737	1.335	402
Doctor Name	7.697	5.600	2.097
Location Name	518	302	216
Hospital Name	5.204	3.602	1.602
Date	7.651	5.490	2.161
Identifier	5.110	3.912	1.198
Phone Number	271	201	70
Age	16	13	3

Following the annotation stage, dataset is divided into two different sets. Training set contains 669 medical discharge summaries and test set contains 220 medical discharge summaries. In total, resulting dataset has 889 annotated medical discharge summaries. Table 4.1 shows number of tokens grouped by PHI categories in each different set of the corpus.

The table clearly shows count of non-PHI words in the dataset is considerably greater than the count of PHI words considering number of non-PHI tokens is equal to 444,127 whereas number of PHI tokens is only 28,204. This difference introduces another difficulty for automated de-identification systems and causes the systems to be skewed over non-PHI classification while decreasing their recall scores greatly.

On top of the dataset I explained in this section, I have also created another dataset by arranging training set in the scope of this work. Although the nature of de-identification problem includes unbalanced data since there are much more non-identifying information than personal information, I wanted to experiment training same network with a balanced training set. In order to do this, I have carefully removed some tokens from training set without breaking the sentence integrity. This way, one is able to see the training and evaluation metrics of the same network with a similar balanced dataset.

Table 4.2: Example sentences from dataset

Example sentences within dataset
Discharge Summary name: *GIVEN, TIN R*
DISCHARGE NOTIFICATION *LONGSWALD , NAER*
Shalyo Kaysplass is a 36-year-old male, with metastatic malignant fibrous histiocytoma of bone...
Dictated By: *NALA BEST*, M.D.
The patient was admitted to the *Oaksgekesser Memorial Hospital* on *7-28*-94 for placement of an infusaid...
ADMISSION DATE: 2004-*08-31*
This is to notify you that your patient , *NERSLET , BERISS* arrived in the Emergency Department...

In order to explain the corpus better, Table 4.2 shows example sentences from the corpus. In the table above, labelled words are highlighted with italic font. These examples might give an insight regarding the ambiguity of corpus and show three different PHI categories.

4.3 Training

In order to train the model, first I pre-processed the data. Pre-process step includes tokenization of plain text records, tagging tokens with their part-of-speech tags and then creating 9-word sequences as explained in Chapter 3 of this document. For tokenization and part-of-speech tagging processes, I have used Python's NLTK module since it is a commonly used tool that provides natural language processing methods and it is possible to tokenize a bulk of text while tagging the tokens with their part-of-speech tags at the same time with NLTK library. [25]

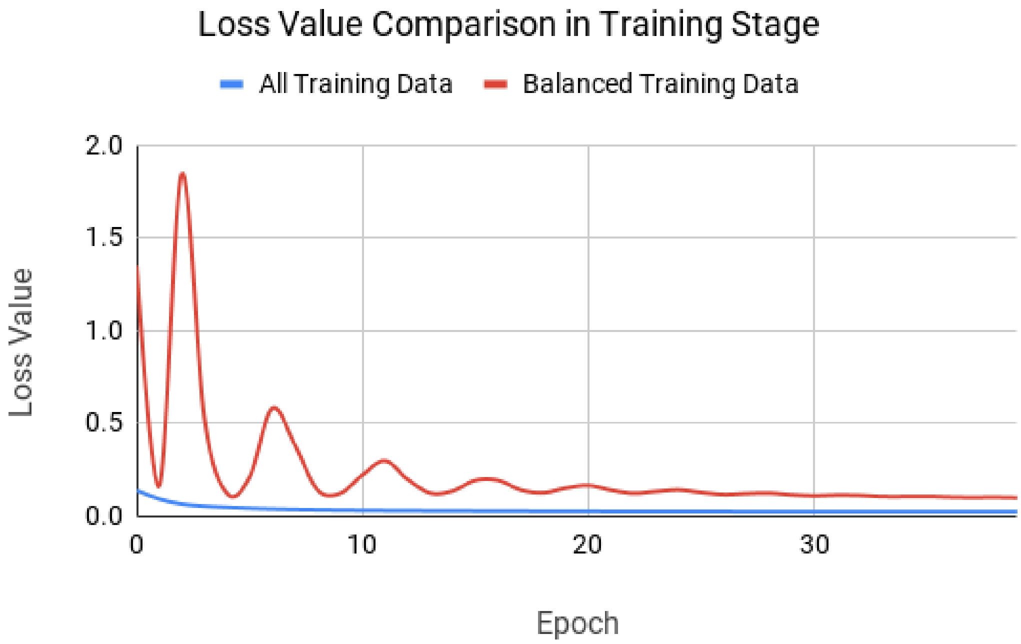

Figure 4.1: Loss value after each epoch in training stage.

In Anonimatik model, context encoder module has 256 hidden units. The dimensions of context embedding vectors are set to 512 and used in the decoder module. Initial values of weights and biases are sampled from a normal distribution with mean of 0 and standard deviation of 1. Cross entropy loss is chosen as loss function. The network is trained by using back-propagation through time [26] with Adam optimizer [27].

Both datasets are used as training data but only original testing set is used for testing. The difference in training data makes a huge difference in same network. This difference is clearly visible in the following figures.

Figure 4.1 shows changes in loss value for two different runs with the same network during training stage. In the first run which is shown with a red line, I used balanced training data explained in the previous section of this document. On the other hand, second run which is shown with a blue line uses whole training data.

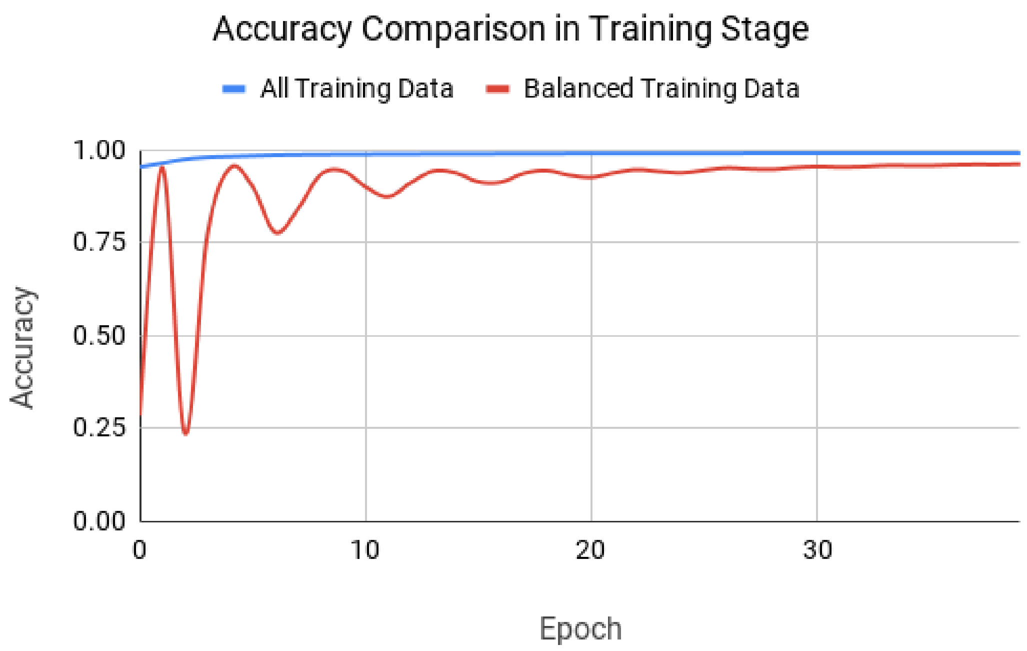

Figure 4.2: Accuracy value after each epoch in training stage.

The difference between the two loss values in Figure 4.1 shows that number of negative examples in training set affects weight updates and prevents network to change its weights as much as compared to balanced training set. Oscillation of balanced dataset with smaller wave amplitudes in following epochs shows network slowly converges optimized point.

As expected, accuracy percentage chart shown in Figure 4.2 looks similar to loss value chart. They are almost identical to their mirrored version over horizontal axis. For the second run -blue coloured line- accuracy value stays close to one which is expected since most examples in training data are annotated as negative. However, it is clear that first run -red coloured line- fluctuates considerably more than the blue coloured second run showing network trained on the balanced dataset is more prune to false positives.

In training stage, I have calculated the number of true positives, true negatives, false positives and false negatives after each epoch. Figure 4.3 and Figure 4.4 shows those metrics below.

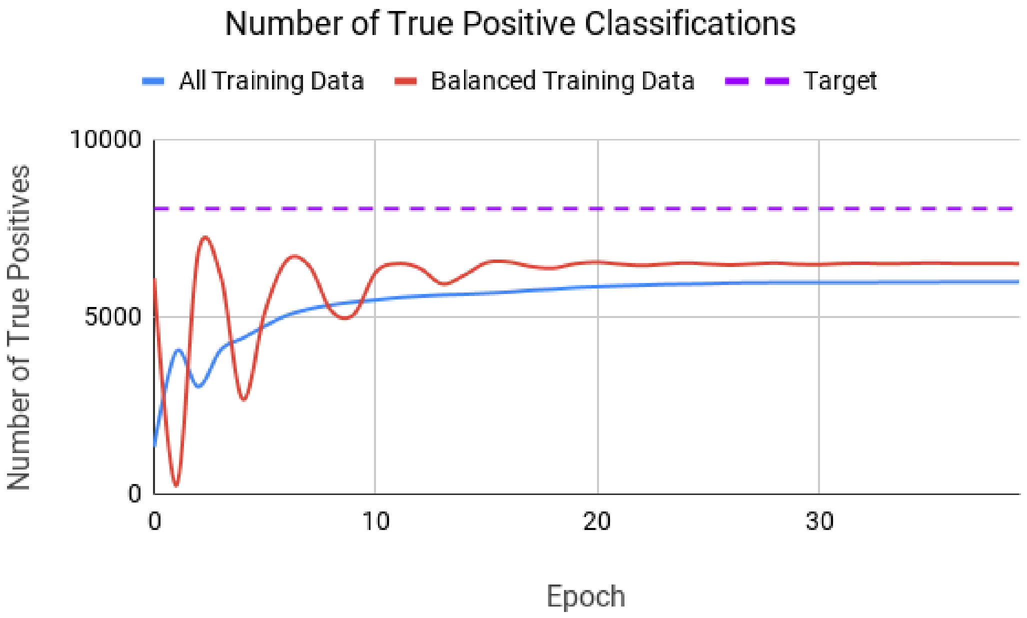

Figure 4.3: Number of true positives after each epoch.

Figure 4.3 shows number of true positives after certain epochs. Similar to charts shown previously in this chapter, this chart also shows two separate runs with different training sets. In addition to those two lines, this chart contains a purple dashed line which shows the target value for number of true positive classifications. True positives chart shows that network trained with balanced dataset tends to change classifications more frequently than the network trained with whole training dataset. This might be the result of loss function penalizing false positives and false negatives equally in balanced dataset. Whereas, in the case with whole training data, false positives are penalized much more than false negatives because of their frequency in the training set.

As shown in Figure 4.3, both runs converged to final state after approximately same epoch. The resulting number of true positive classifications values show balanced training run is more aware of positive samples which results in having a higher recall value. Although there is a huge difference between datasets as explained in the previous section of this chapter, the difference in the result is less than 10% which suggests training with all training data might the better choice. Despite the fact that balanced data gives smaller F_1 score, having greater recall values might be preferred in some cases.

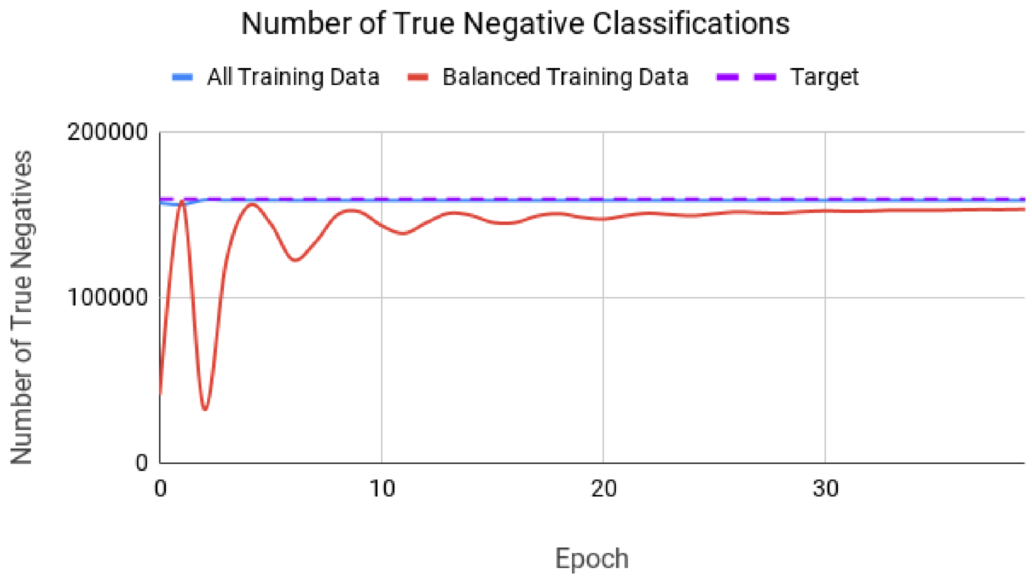

Figure 4.4: Number of true negatives after each epoch.

Figure 4.4 shows another chart visualizing the number of true negative classifications after certain epochs. In contrary to convergence epoch similarity between two runs in previous figures, number of true negative classifications chart shows a huge difference between those in Figure 4.4. While both runs converge after 30th epoch in previous charts, convergence epochs differs slightly in true negatives chart. Obtained results are differs approximately 5% in the end.

Similar to true positives chart, fluctuations in the run with balanced data are considerably greater compared to the run with whole train set. This graph shows the probable precision outcome of the run with whole training data would be better than the balanced run.

Both training runs show my model learn the contexts as it improves after epochs. Although it is easier to track the learning on the balanced set, network updates are visible in both runs.

Having run the same model multiple times with same data, one can conclude that Anonimatik model does not depend on random initialization and it is capable of learning the context and makes better predictions in later epochs.

After checking the results of both runs, network trained with unbalanced original data seems to be most feasible choice. Therefore, next section will be focused on the results obtained with first run.

For real world de-identification systems, recall score might be more important than precision score. In the case of lower precision, one might remove important information from discharge summary. Although it is not wanted, it might be preferred to not removing personal information holding words in the real world usage. The experiment done in this section shows it is possible to use the same network to create alternative de-identification system just by partitioning training data.

4.4 Results

In this chapter of my thesis, I present and comment the results of my model on i2b2 challenge dataset. I have calculated the performance of my model based on 2 main metrics, precision and recall.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{41}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{42}$$

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{43}$$

In our automatic de-identification problem, precision score (also known as positive predictive value) is the fraction of personal information holding words among the positively classified words, while recall score (also called sensitivity) is the fraction of personal information holding words that have been marked correctly over the total amount of personal information holding words. Both precision and recall scores are therefore a measurement of relevancy.

In the most basic terms, greater precision in automatic de-identification means that the system identified substantially more personal information holding words than irrelevant words, while greater recall means that the model identified most of the personal information holding words in the whole text.

In the scope of this work, the precision score is calculated by dividing the total count of true positive examples (i.e. the number of personal information holding words correctly marked as PHI-word) by the sum of true positive examples and false positive examples (i.e. total number of words labelled as personal information holding word) as shown in equation (41). Recall in the de-identification context is defined as the number of true positives divided by the sum of true positives and false negatives (i.e. total number of words in the texts that contains personal information) as shown in equation (42). Likewise, $F_1 score$ is another metric that takes into consideration both precision and recall I used to compare the performance of my model. Equation (43) shows the formula of $F_1 score$ which I used to calculate.

Obtained results on dataset with comparison to others are shown in Table 4.3.

Model	F-Score	Precision	Recall
Aramaki	0.94	0.95	0.93
Guo	0.81	0.83	0.80
Hara	0.90	0.91	0.90
Szarvas	0.97	0.98	0.96
Uzuner	0.98	0.99	0.98
Wellner	0.96	0.97	0.96
Anonimatik[2]	0.91	0.95	0.88
Anonimatik[1]	0.68	0.53	0.95

Table 4.3: Comparison of automatic de-identification systems

As explained previously in this chapter, I present the results of both runs with different training sets.

Results listed in Table 4.3 shows when trained with whole training dataset my model becomes skewed on negative prediction. This skewness decreases the recall score of model from 0.95 to 0.53. Along with the drastic 44% decrease in recall score, my model achieves a better precision score increasing precision score from 0.88 to 0.95. However when we evaluate the scores all together, the increase in the precision score does not supersede the decrease in recall score. Having lower F_1 score in balanced run shows it is better to use the whole training data to train the network.

On the other hand, the reason behind having better F_1 score is having a test set that is skewed as much as training set. When trained with the balanced dataset, the skewness of model is eliminated therefore Anonimatik model can not achieve good results in that case.

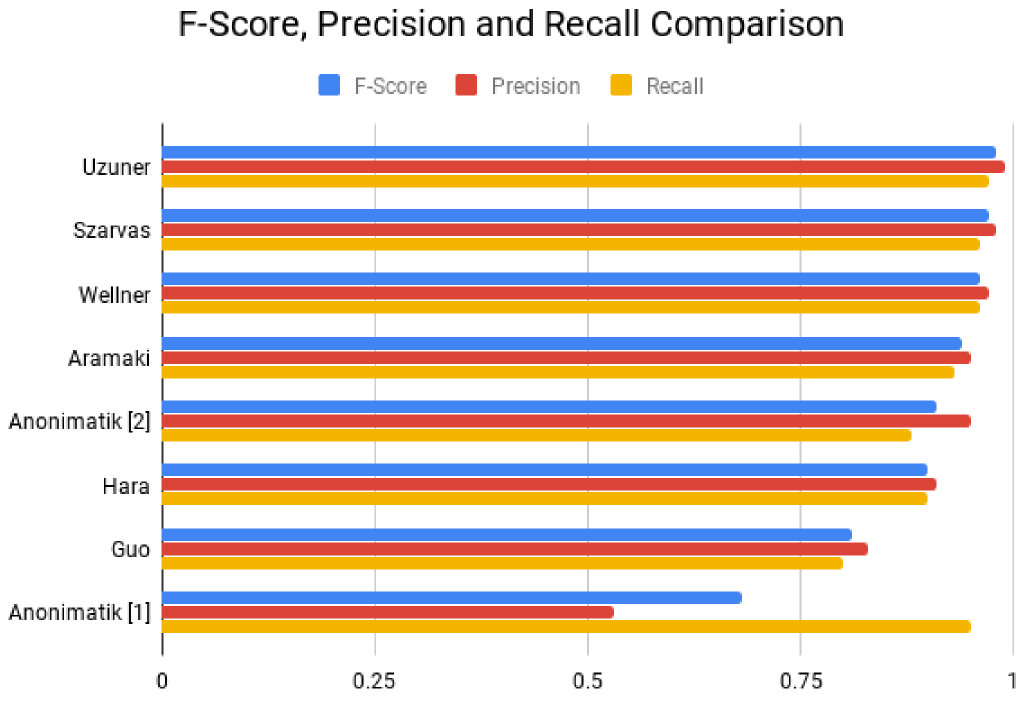

Figure 4.5: Comparison with other de-identification models.

Figure 4.5 shows visualized comparison with other automatic de-identification models that focus on the medical discharge summaries. It is ordered by F_1-score and both models (first one trained with balanced dataset, second one is trained with whole training dataset) are shown in the figure. Change in precision and recall scores on both runs is clear in the figure.

In spite of having more successful models as comparison, Anonimatik model trained on the whole training set achieves a comparable performance without using any knowledge resource.

When comparing my model to other model that does not use knowledge resources, Anonimatik model trained on the whole training set achieves a better F_1-score and better precision score.

CHAPTER 5

CONCLUSIONS

In the scope of this thesis, I created a two-layered neural network model to automatically identify personal information holding words in plain text data. My model uses a bi-directional long short term memory network to encode local contexts of words and a fully connected neural network to produce the probabilities stating whether target word is a personal information holding word or not. Unlike other approaches, my model does not require using external knowledge resources (e.g. name lists, medical term lists, common word lists etc.) making it easier to train and use with different languages. I have trained and evaluated the performance of my model on the corpus created for the automated de-identification challenge organized by Informatics for Integrating Biology & the Bedside (i2b2) in 2006. My model achieved 0.91 F-score on the test set with 0.88 recall and 0.95 precision scores. Results show that my model performs similar to existing models which depend on external knowledge resources and performs better than the similar model that does not use external knowledge resources. Over the course of this study, I showed that an automatic de-identification model can disambiguate contexts and identify personal information holding words without depending on any external dictionaries. Furthermore, I experimented my model with balanced and unbalanced training sets and compared the differences between the results. Differences between two runs show that when trained with balanced dataset my model becomes skewed which results in greater recall scores. Although it is not ideal for F-score, higher recall score might be preferable in some situations. The model I have created in the scope of this thesis is the only publicly available automatic de-identification model that is designed to be used without requiring any external dictionaries.

REFERENCES

[1] Centers for Medicare & Medicaid Services, "The Health Insurance Portability and Accountability Act of 1996 (HIPAA)." Online at http://www.cms.hhs.gov/hipaa/, 1996.

[2] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] S. M. Meystre, F. J. Friedlin, B. R. South, S. Shen, and M. H. Samore, "Automatic de-identification of textual documents in the electronic health record: a review of recent research," *BMC medical research methodology*, vol. 10, pp. 70–70, 08 2010.

[5] P. Voigt and A. v. d. Bussche, *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer Publishing Company, Incorporated, 1st ed., 2017.

[6] G. N. A. of Turkey, "KİŞİSEL VERİLERİN KORUNMASI KANUNU." `http://www.mevzuat.gov.tr/MevzuatMetin/1.5.6698.pdf`, 2016. [Online; accessed 26-November-2018].

[7] K. V. K. Kurumu, "KVKK KİŞİSEL VERİ SAKLAMA ve İMHA POLİTİKASI." `https://www.kvkk.gov.tr/SharedFolderServer/CNSFiles/e95s5382-23bd-4v38-8114-0522234f5887.pdf`, 2018. [Online; accessed 14-December-2018].

[8] K. V. K. Kurumu, "Data Protection in Turkey." `https://www.kvkk.gov.tr/SharedFolderServer/CNSFiles/5c82cb8c-7cc0-4f60-b0a7-85cd90399df8.pdf`, 2018. [Online; accessed 20-January-2019].

[9] E. Aramaki, P. D, T. Imai, and P. D, “Automatic deidentification by using sentence features and label consistency,” 2006.

[10] Y. Guo, R. Gaizauskas, I. Roberts, and G. Demetriou, “Identifying personal health information using support vector machines,” in *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data*, 2006.

[11] H. K, “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*, 2006.

[12] G. Szarvas, R. Farkas, and R. Busa-Fekete, “State-of-the-art anonymization of medical records using an iterative machine learning framework,” *Journal of the American Medical Informatics Association : JAMIA*, vol. 14, pp. 574–580, Sep-Oct 2007.

[13] O. Uzuner, T. C. Sibanda, Y. Luo, and P. Szolovits, “A de-identifier for medical discharge summaries,” *Artificial intelligence in medicine*, vol. 42, pp. 13–35, 01 2008.

[14] B. Wellner, M. Huyck, S. Mardis, J. Aberdeen, A. Morgan, L. Peshkin, A. Yeh, J. Hitzeman, and L. Hirschman, “Rapidly retargetable approaches to de-identification in medical records,” *Journal of the American Medical Informatics Association : JAMIA*, vol. 14, pp. 564–573, Sep-Oct 2007.

[15] H. T. Siegelmann and E. D. Sontag, “On the computational power of neural nets,” in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT ’92, (New York, NY, USA), pp. 440–449, ACM, 1992.

[16] A. Vance, “This man is the godfather the ai community wants to forget,” May 2018.

[17] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional lstm and other neural network architectures,” *Neural Networks*, vol. 18, no. 5, pp. 602 – 610, 2005. IJCNN 2005.

[18] E. Kiperwasser and Y. Goldberg, “Simple and accurate dependency parsing using bidirectional lstm feature representations,” *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 313–327, 2016.

[19] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014.

[20] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

[21] D. Mané *et al.*, “Tensorboard: Tensorflow’s visualization toolkit, 2015.”

[22] C. Boettiger, “An introduction to docker for reproducible research,” *ACM SIGOPS Operating Systems Review*, vol. 49, no. 1, pp. 71–79, 2015.

[23] P. HealthCare, “Informatics for Integrating Biology and the Bedside.” `https://www.partners.org/Services/General/Research/Research-Technology/Informatics-for-Integrating-Biology-and-the-Bedside.aspx`, 2019. [Online; accessed 12-January-2019].

[24] O. Uzuner, Y. Luo, and P. Szolovits, “Evaluating the state-of-the-art in automatic de-identification,” *Journal of the American Medical Informatics Association : JAMIA*, vol. 14, pp. 550–563, Sep-Oct 2007.

[25] E. Loper and S. Bird, “Nltk: the natural language toolkit,” *arXiv preprint cs/0205028*, 2002.

[26] P. J. Werbos *et al.*, “Backpropagation through time: what it does and how to do it,” *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[27] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.