8-2021

Linking social media, medical literature, and clinical notes using deep learning.

Mohsen Asghari
University of Louisville

LINKING SOCIAL MEDIA, MEDICAL LITERATURE, AND CLINICAL NOTES USING DEEP LEARNING

By

Mohsen Asghari
B.S. Azad Tehran Central Branch University, 2008
M.S. Khaje Nasir University of Technology, 2012

A Dissertation Submitted to the
J.B. Speed School of Engineering the University of Louisville
In Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy
In Computer Science and Engineering

Department of Computer Science and Engineering
University of Louisville
Louisville, Kentucky

August 2021

LINKING SOCIAL MEDIA, MEDICAL LITERATURE, AND CLINICAL NOTES USING DEEP LEARNING

By

Mohsen Asghari
B.S. Azad Tehran Central Branch University, 2008
M.S. Khaje Nasir University of Technology, 2012

A Dissertation approved on

July 26, 2021

by the following Dissertation Committee:

Dr. Adel Elmaghraby

Dr. Monica Gentili

Dr. Hui Zhang

Dr. Juw Won Park

Dr. Daniel Sierra Sosa

DEDICATION

I dedicate this study to my beloved parents and my sister, Fatemeh Mosavi, Abdollah Asghari, and Maede Asghari. Who, despite being physically far from me, are a major source of inspiration and strength for me. Their continual moral, spiritual, and emotional support motivates me to persevere and never give up on my dreams. I would also like to dedicate this study to my lovely wife, Antigona Mehani-Asghari. She, too, provides constant support and encouragement through the challenges of graduation and life.

ACKNOWLEDGMENTS

First and foremost I am extremely grateful to my supervisors, Prof. Adel S. Elmaghraby and Dr. Daniel Sierra-Sosa for their invaluable advice, continuous support, and patience during my PhD study. Their immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life. I would also like to thank Dr. Monica Gentili for her support on my study. Also, I would like to thank Dr. Hui Zhang, and Dr. Juw Won Park for their great comments and supports. It is their kind help and support that have made my study and life in the University of Louisville a wonderful time. Finally, I would like to express my gratitude to my parents, my wife. Without their tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my study.

ABSTRACT

LINKING SOCIAL MEDIA, MEDICAL LITERATURE, AND CLINICAL NOTES USING DEEP LEARNING

Mohsen Asghari

July 26, 2021

Researchers analyze data, information, and knowledge through many sources, formats, and methods. The dominant data format includes text and images. In the healthcare industry, professionals generate a large quantity of unstructured data. The complexity of this data and the lack of computational power causes delays in analysis. However, with emerging deep learning algorithms and access to computational powers such as graphics processing unit (GPU) and tensor processing units (TPUs), processing text and images is becoming more accessible. Deep learning algorithms achieve remarkable results in natural language processing (NLP) and computer vision.

In this study, we focus on NLP in the healthcare industry and collect data not only from electronic medical records (EMRs) but also medical literature and social media. We propose a framework for linking social media, medical literature, and EMRs clinical notes using deep learning algorithms. Connecting data sources requires defining a link between them, and our key is finding concepts in the medical text. The National Library of Medicine (NLM) introduces a Unified Medical Language System (UMLS) and we use this system as the foundation of our own system.

We recognize social media's dynamic nature and apply supervised and semi-supervised methodologies to generate concepts. Named entity recognition (NER) allows efficient extraction of information, or entities, from medical literature, and we extend the model to process the EMRs' clinical notes via transfer learning. The results include an integrated, end-to-end, web-based system solution that unifies social media, literature, and clinical notes, and improves access to medical knowledge for the public and experts.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

The quality of hospital services significantly affects people's life. Hospital professionals capture every service, or clinical event, and evaluate ways to improve the quality of their services. Currently, massive storage devices and complex information management systems allow these professionals to store, process, and capture every clinical event. The data in these devices improves a hospitals' quality of services. However, connecting and transforming data in existing healthcare systems proves complicated, time-consuming, and expensive [1]. Having a meaningful and complete data transformation requires a deep understanding of the data process. The lack of data integrity seen in laboratories and intensive care units causes data reliability limitations. Electronic health records (EHR) and electronic medical records (EMR) were the first pioneer systems to gathering data promptly, efficiently, and overcame old data gathering methods restrictions.

1.1.1 Problem Definition

Electronic health records (HER) and electronic medical records (EMR) enhance accessibility, quality, and reliability of medical data, however, some challenges remain. The most crucial challenge in healthcare data collection involves the massive amounts of unstructured data. Research reveals 80% of healthcare data remain unstructured and untapped, such as images, text, audio, signals, and so forth. It is expensive for EHR and EMR to process and store this complex data. For example, all the clinical encounters, assessments, and clinical activities between physician and patient describe and dictate text format. A pathologist or radiologist's diagnosis will store an image and a text of their observations. Unfortunately, many of these healthcare professionals exclude this data from practical healthcare analytic systems.

Researchers apply conventional machine learning algorithms such as support vector machines **(**SVMs**),** decision trees, and logistic regression to predict diagnosis, find correlations, and other purposes for many years. These algorithms limit the methods for processing extensive data and finding nonlinear patterns [2]. Modern deep learning algorithms such as recurrent neural networks (RNN) and convolutional neural networks (CNN) can discover nonlinear ways in intricate structures and high-dimensional data. These methods apply in different domains such as science, healthcare, businesses, or governments [2].

A survey of recent publications shows a noticeable increased interest in using deep learning in EHR data analysis. Between 2012-2017 there were over 700 deep learning research publications. From 2016 to 2017, over 500 of these research reports were published in just two years [3].

We can categorize these researches into six major categories: 1. Phenotyping, 2. Outcomes prediction, 3. Readmission prediction, 4. Concept extraction, 5.Clinical support system, and 6. De-identification. Below are more in-depth definitions.

1. **Phenotyping:** Maps illness and other clinical data concepts into categories [4, 5].
2. **Outcomes Prediction**: Predicts the patient outcome such as heart failure [6], suicide risk stratification [7].
3. **Readmission Prediction:** Predicts the future risk from a patient. Readmission Prediction is an essential healthcare prediction, including readmission after discharge [8].
4. **Concept Extraction:** Finds a proper way to convert free text into a structured data format. Clinical data usually contains text such as clinical notes or descriptions; the fundamental task of unstructured data analysis is to transfer unstructured to structured layouts [9-11].
5. **Clinical Support System:** Helps reduce errors and improve treatment procedures' success. We can divide this system into the Basic and Advance Support system. Essential decision support includes patient distinctive medical information such as drug-allergy checking, necessary dosing guidance, formulary decision support, duplicate therapy checking, and drug-drug interaction checking. Advanced decision support includes treatment and diagnosis-related information such as dosing support, guidance for medication-related laboratory testing, drug-pregnancy checking, and drug-disease contraindication checking. [12]
6. **Patient De-identification:** Removes sensitive data from clinical information. Transferring information between facilities has to follow HIPPA regulations in the

USA. These strict policies cause inefficiency in accessing patient data and need to be addressed.

Electronic Health Records (EHRs) allows clinics and hospitals to collect data associated with patients in all formats, including demographic data, clinical notes, images, lab results. Analyzing EHRs data could reduce patients' costs, readmissions, triage, decompensation, adverse events, and treatment [13]. However, almost all EHR data remains unstructured, and it's difficult to merge and analyze. Figure 1 presents a descriptive schema for the different formats of frequent medical data [14]. We have more data types under the unstructured data section; however, we focus on text and expand discussion later. Text data in healthcare comes from the following sources — EHR, social media, journals, and publications.

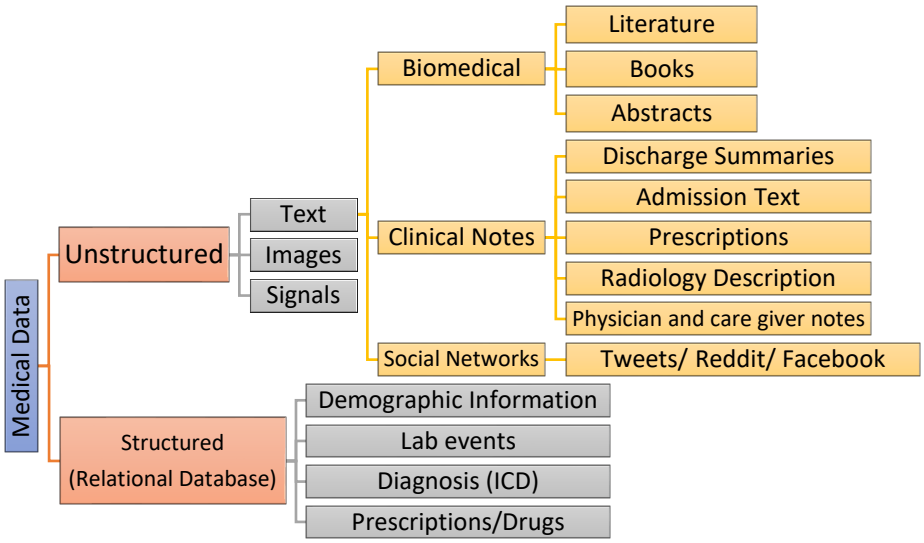

Figure 1 The two major categories of healthcare data are structured and unstructured. although we are focusing on text, unstructured data also includes images and signals

There are two goals to this study. The first goal is to introduce a unified framework that handles the mixture of text medical information shown in Figure 1. This framework will

link different information sources, ranging from social knowledge and beliefs to clinical and biomedical data. The second goal is to construct a platform to help doctors and the public find patterns and solutions, leading to link the gap between public health belief and biomedical knowledge.

1.1.2 Importance of Study

We categorize health care text into biomedical publications, clinical notes, and social comments. Biomedical publications gathers Medical Doctors (MDs) texts and incorporate them with other scientists. MDs, nurses, caregivers, radiologists, etc. generate various clinical notes and dictate clinicians' observations and practice. Finally, social media texts can be general ideas, advice, or individual observations about healthcare topics. Since the authors of social media texts often share publicly, it's less liable. Even if they are not experts in the field, the public's contributions are still important to research.

Social network sources represent public health beliefs and help understand many topics such as diagnosis, drugs, and claims. There are 140 potential healthcare uses of Twitter reported in the literature [15]; Here are the ten most common use applications for social media data:

1. Disaster alerting and response
2. Diabetes management (blood glucose tracking)
3. Drug safety alerts from the Food and Drug Administration
4. Biomedical device data capture and reporting
5. Shift-bidding for nurses and other healthcare professionals
6. Diagnostic brainstorming
7. Rare diseases tracking and resource connection

8. Providing smoking cessation assistance
9. Broadcasting infant care tips to new parents
10. Post-discharge patient consultations and follow-up care

We need to mention that part of social-network data is private such as consulting through websites where medical doctors answer questions. This data is hard to access because of HIPPA-compliance laws, therefore private social-network data is out of the scope of this thesis.

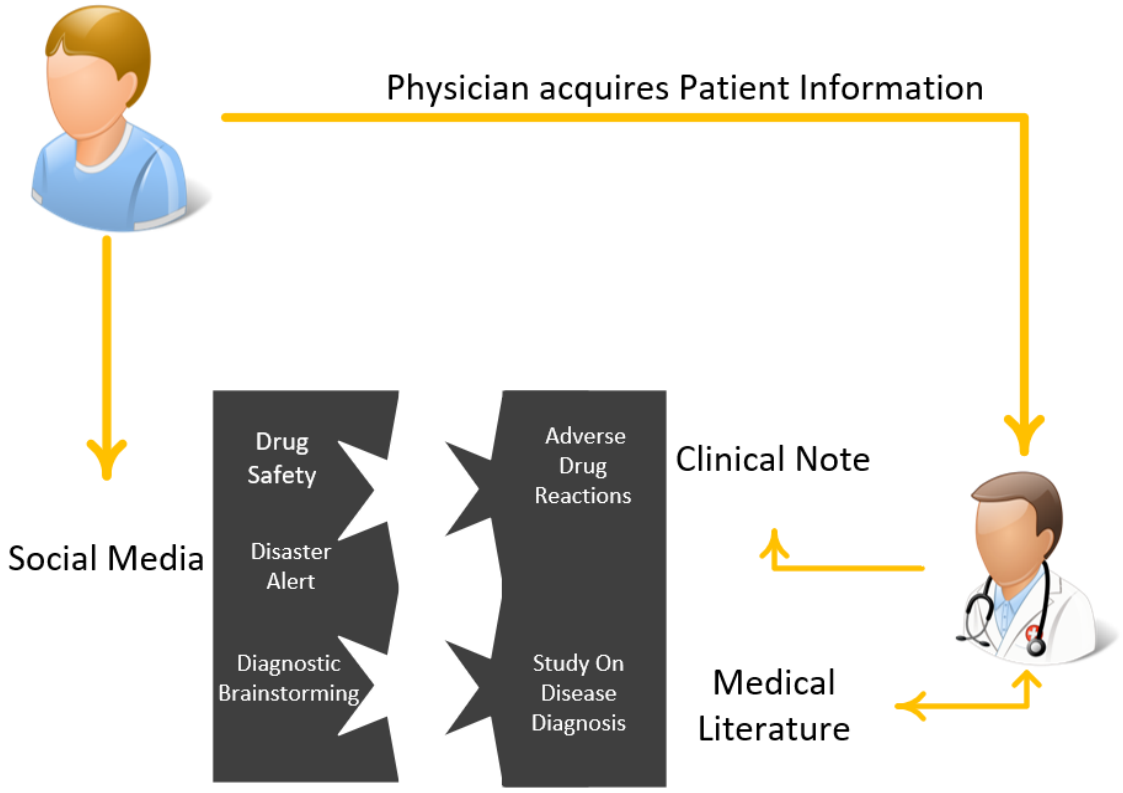

Figure 2 Explanation of gap between note created by physician in the format of clinical note and biomedical publications and patient in social media

These data sources have a weak connection. Figure 2 shows how Patients provide information to physicians and health practitioners; these data will convert to publications and clinical notes, radiology, pathology reports, and interviews. On the other hand, patients

share much information about their sentiment of a health service they received or share their symptoms or disease on social media. However, these data are somehow connected but link the gap between them could benefit the public and experts in healthcare.

1.1.3 Proposed Solution

The key challenge in analyzing a text in healthcare is unstructured data storage and the complex retrieval. To mitigate the complexity of retrieving the clinical or biomedical text information, we need to find four types of features in the text.

First, negation, sentences or phrases that mention symptoms that the patient does not have or reports a diagnosis that did not work. For example, "treatment does not reduce the pain" [15, 16].

Second, certainty, sentences or phrases that mention the possibility of symptoms; these kinds of sentences are common when nurses write the notes. For example, "unlikely pneumonia" will not negate the situation, but the writer also was unsure, so it is just a possibility.

Third, temporality, sentences or phrases that note symptoms that might change over time during the treatment. The temporality separates into two parts: First, the historical components, when the patient had symptoms of at least 14 days' precedent to the examination date. Second, the hypothetical details, when the physicians assume that the patient might have certain symptoms for the subsequent examination, for example, "the patient should return if she develops a fever."

Studies suggest that distinguishing temporal statements complicates identifying recent problems because there is a high possibility of a transition between historical and

contemporary and hypothetical to current. Another reason for this complication lies in the fact the algorithm has little data to train for these transitions [17, 18].

Fourth, family history, sentences or phrases that note patient's family's medical history. "The patient's father has a history of CHF (Congestive Heart Failure)." After annotating the text by these four features, we need to handle the matching challenge. According to Amelie Gyrard et al., many ontologies created do not have any comments and labels; this makes it hard to develop a matching structure [19].

The Named Entity Recognition (NER) can handle this challenge by recognizing the entity and matching them together. These entities would help us find the four features in medical text and connect the health-related social media domain to publication and clinical notes.

We define ten name entities below:

(1) **Disease**: Name of the condition that is stated explicitly in the text.

(2) **Severity**: (a severe form of disease entity): It is a disease entity of etiological origin from a relatively mild disease entity.

(3) **Trigger:** refers to the cause (entity/substance/environmental condition) of the disease. For example, pollen, weather, cough can cause asthma.

(4) **Location**: It refers to the place affected in human anatomy—for instance, bones, muscles, nose, lungs, etc.

(5) **Control**: It is a dichotomous concept whose value is "yes" when the tweet talks about disease control, reduction in severity, or reduced frequency of asthmatic attacks

(6) **Treatment/Device**: These entities define a treatment or device used by the patient or clinician to treat the disease entity stated in the text. For example, an inhaler is a device to treat asthma.

(7) **Time**: As we discussed, we need to know the procedure or test and doctor examination time to find the temporality.

(8) **People**: related to a patient, and we want to understand the Experiencer and their histories

(9) **Laboratories Test/ Procedures**: It related to testing names that doctors did on a patient and reported them—for example, Blood Tests and the results.

(10) **Drug**: Related To name of Drugs

These entities will mainly connect publication and clinical notes. We have a good deal of research and database focused on Named entity recognition on the journal and clinical notes. Social media's intensive noisy environment, dynamic changes, misspelling, abbreviations, and emojis complicates identifying healthcare data.

We categorize these challenges into three major groups:

1. Intensive noisy environment

Social media such as Facebook, Twitter, and Reddit include various topics from politics, sports to advertisement and job vacancies. Processing all of them would be impractical. Thus, we need an intelligent way to filter and reduce the noise for more quality data.

2. Dynamicity of social media

The dynamicity of social media makes it challenging to keep the machine learning models accurate. For example, in a presidential election, they relate most comments to

support or disapproval of a candidate. After that, the topic will cool down, and we can see a topic shift. This phenomenon can happen with many topics, including healthcare and related topics. The dynamicity is not only limited to time, but location will also be a major factor, and that leads us to spatio-temporal research.

3. Incompleteness and inconsistency

Data in social media has a variety of characters that makes it challenging to use. Length of comments is one of the critical factors. For example, Twitter has a 280-character limits, whereas comments on New York Times articles do not have character limits, making the data inconsistent. Spelling the data or expressing emotion with emojis will bring the data incomplete as well. We must remove many words and characters that are meaning full for human readers but meaningless and incomplete for machines.

In Figure 3, we explain the road map of a proposal to address all the challenges in-depth in five chapters. Starting from Chapter 2, we study the background and literature review of related research. Chapter 3 introduces our Extract Transfer and Load system [20]. Chapter 4 present a framework named real-time automatic social media topic detection (REAT).

This system will help the health care communities follow the trend of topics over time and location. This kind of system would allow the hospital; research institution follows issues based on time and place [21-23]. Chapter 5 proposes a low-cost biomedical named entity recognition (BINER) to make named entity recognition more available to everybody, in contrast with state-of-arts techniques that required high-end hardware with intensive GUP powers.

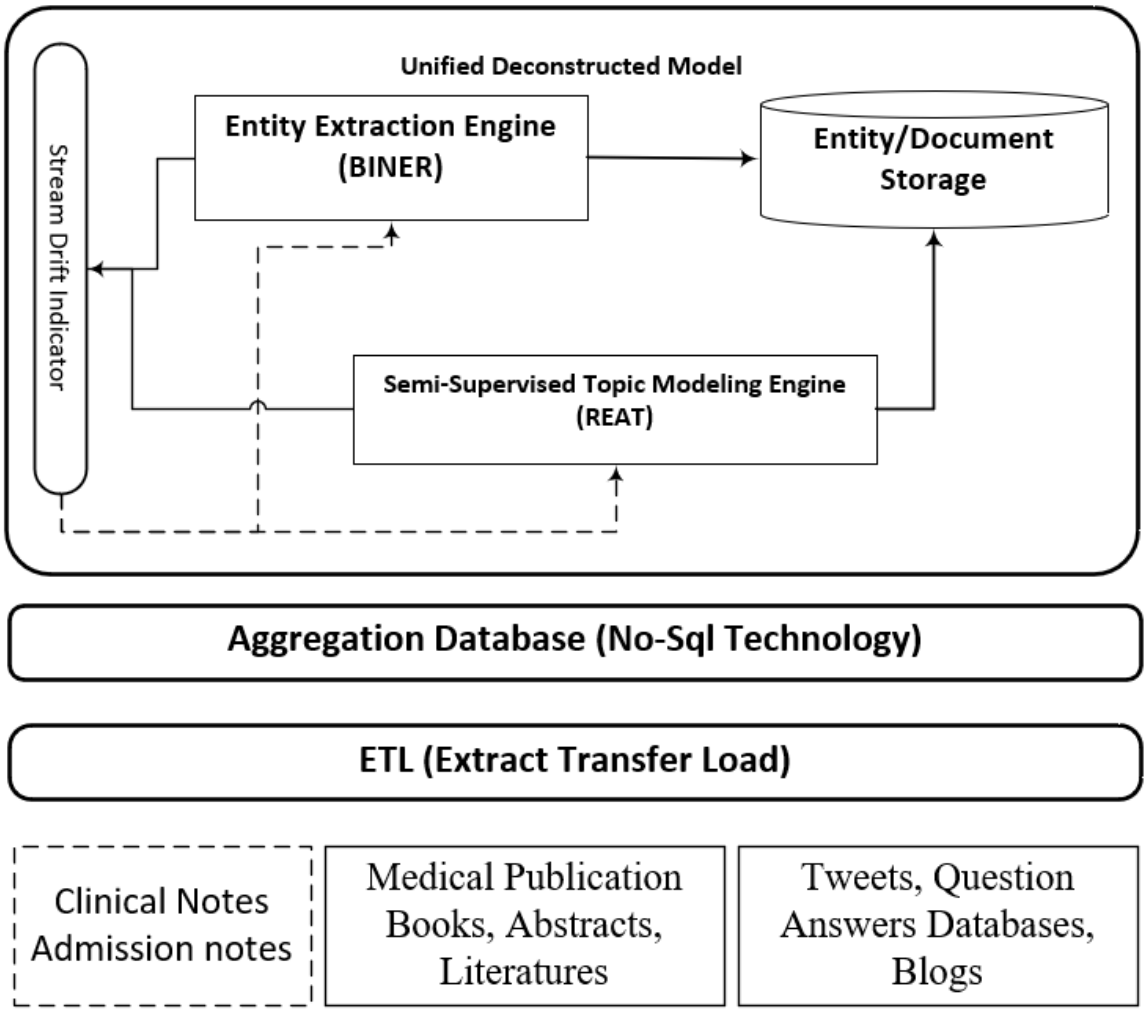

Figure 3 Defines unified deconstruction model from bottom to top

1.1.4 Hypothesis and Objective

Our main goal is as follow:

> Create a unified model to link social media, literature, and clinical notes to improve access to medical knowledge for the public and experts.

To achieve our goal, we define the following milestones:

1. Introduce a robust decision-making procedure for selecting a model to explain the active topics.
2. Use transfer learning techniques to enhance the framework's ability to detect unrelated messages over Twitter data streams.
3. Create an automatic deep cleaning method to enhance data quality to perform better classification in a noisy environment.
4. Create a concept drift indicator to enhance the topic modeling and classification models.
5. Introduce low-cost Named Entity Recognition in Biomedical using deep learning techniques.
6. Use Transfer learning to create a multi-source Named Entity Recognition model to handle biomedical text and Clinical text.
7. Unified all the parts and created a framework to extract biomedical, clinical text, and social media knowledge.

There are three challenges to achieving these milestones:

(1) Linking biomedical, clinical, and social data sources appropriately

(2) Merging public health beliefs and knowledge in healthcare

(3) Reducing the complexity of clinical text analysis

To address these challenges, we contribute to the named entity recognition (NER) method and create a knowledge base that can provide and connect public knowledge and beliefs to medical and biomedical text.

We will contribute to health care in the listed ways:

A. Extract embedded knowledge and make it available to research communities.

B. Create a framework to evaluate the public beliefs regarding different treatments and healthcare concepts.
C. Make reliable clinical concepts available to the public.

1.2 Relevant Methodologies

1.2.1 Latent Semantic Analysis

The probability of finding a set of words with the same meaning is more likely in a set of documents with the same topic. Latent semantic analysis (LSA) finds a group of words that can create clusters of documents. Scott Deerwester [24] proposes a model to find these words between documents. Creating an LSA model requires three steps.

The first step is to generate a matrix that summarizes documents and words. In this matrix, the rows represent unique words, and the columns represent the documents and will grow with the number of documents. [25] explains the research that cumulating frequencies in a sublinear fashion would improve the result. The second step uses singular-value decomposition (SVD) to reduce the dimensionality. Finally, we use the SVD generated matrix to calculate the similarity between entities. Since both the document and terms are in the same space vectors, we can define a record by some words known as a topic.

1.2.2 Latent Dirichlet Allocation – LDA MALLET

Latent Dirichlet Allocation (LDA) is a three-level hierarchical Bayesian model [26]. LDA aims to reduce the dimensionality of data and create a representation of documents by topics where a group of words defines each topic [26].

LDA following generative process for nth word w_n in a document as follows: $z_n \sim Multinomial\,\gamma$ where z_n And γ represent the sample a topic and document topic

probability matrix respectively, and $w_n \sim Multinomial(\varphi)$ where w_n And φ represent sample a word, topic word probability matrix respectively. The purpose of following this process is to create a sequence of words describing a topic. Recently, a team of researchers at UMASS AMHERST created a MALLET toolkit [27]. The toolkit is public and contains multiple helpful topic modeling techniques, such as the sample-based implementation of LDA.

1.2.3 Biterm Topic Modeling

This model assesses the problem of the sparsity of data in brief texts. [28] proposes the Biterm model, which bases word connectivity on pattern rather than by document. The Biterm model creates terms, such as “breast cancer” and “digital healthcare” by searching the connectivity between the words over the corpus. In this model, each topic is z, word Dirichlet distribution in each topic is ϕ_z and topic distribution is θ. Each Biterm contains two words (w_i, w_j) then they calculate the joint probability of these words as the likelihood of the whole corpus as follows:

$$P(B) = \prod_{(i,j)} \sum_{z} \theta \, \phi_{(i|j)} \, \phi_{(j|z)} \tag{1}$$

1.2.4 Linear Support Vector Classification (SVC)

The Linear SVC is one of the more simple classification methods, which creates a line of "best fit." Each line forms a decision boundary in the data, with two given classes falling on either side of the line, as illustrated in Figure 4. This method is foundational to many other classification methods.

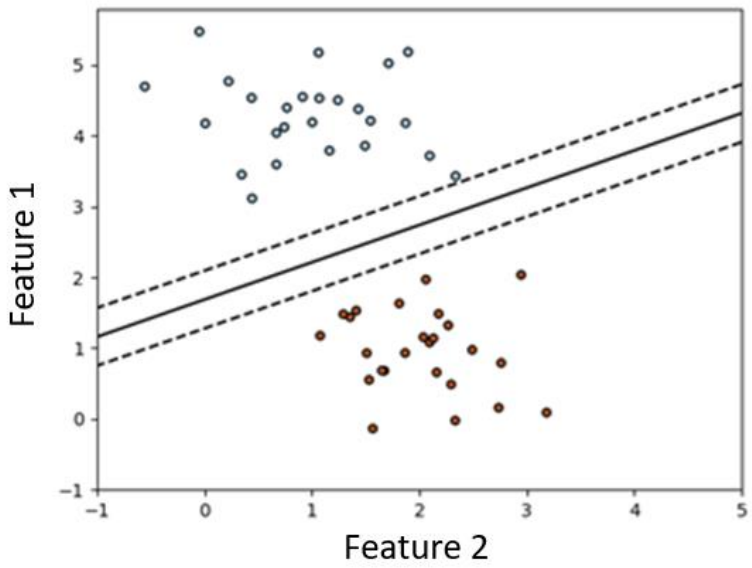

Figure 4 Illustrates the Decision Boundary for Classification

1.2.5 Multinomial Naive Bayes

The multinomial Naive Bayesian classifier extends the traditional Naive Bayesian classifier, and we show the formulation and simplified representation in Equation (2) and (3), respectively.

$$p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)} \tag{2}$$

$$posterior = \frac{prior * likelihood}{evidence} \tag{3}$$

A vital feature of any Naive Bayesian probability is its feature independence assumption (hence Naive). For example, we may consider a fruit an apple if red, round, and about 10 cm in diameter. A Naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any correlations between the color, roundness, and diameter features.

A multinomial distribution is one such that for n independent trials, each of which leads to success for precisely one of the k categories. With each category having a given fixed success probability, there is a probability of any combination of numbers of successes for the various types. This means that it predicts the probability that, given a sample of 100 comments, classify X of them "Yes," and organize the rest as "No." This is distinct from determining the probability of *whether* we will classify a comment as "Yes/No."

1.2.6 Text Embedding

In structured data, such as relational datasets, we have columns and rows to define the data. Each row describes a case, and columns represent features that explain the rows. For example, if we have an orders dataset, each row represents an order case, and columns define the orders such as the price, provider, location, vendor, and so forth. The features are continuous or discreet and easy to transform for mathematical models. Nonstructural data such as text as features cannot define preparing the data for mathematical data as a challenge.

Different techniques such as Hash Vectorization, Term Frequency, Inverse Document Frequency (TF-IDF), GLOVE Stanford, and Google Word2Vec introduce and develop transforming text. Suppose that each Document defines as D and features that define the D is v. The elements can vary from characters, words, or phrases. A sentence defines as a sequence of elements in a document. The location of these features is essential, so we can conclude that a D represent by a vector of features $\vec{V}$. This vector needs to transform into a vector of numbers representing the Document. Thus, the purpose of all the techniques is to describe a text by a vector of numbers.

The vectorization techniques separate into two categories, lazy and attentive. We label GLOVE and Word2Vec as lazy techniques because of these techniques' nature. They train massive datasets over billions of documents, and as a result, each word defines by a vector. These approaches use a dictionary that helps change our document to vectors. They use a good deal of research and show promising results.

The attentive techniques need more effort, such as Hashing Vectorization and TF-IDF. We need to create our dictionary based on the data we gathered in these approaches. There are some pros and cons in these two categories. In the lazy approach, we can transform the data faster. We take advantage of a model that trains on a petabyte of data; however, we might miss many particular words in a specific domain, such as healthcare. The results would not be satisfying enough. The attentive approach, we lose generality and become more specific, and can improve the outcome impressively sometimes.

CHAPTER 2

LITERATURE REVIEW

2.1 Social Media

Social network sites such as Twitter or Facebook provide a communication platform. Recent studies show that Twitter users share advice on health-related information [29, 30]. These sources contain public beliefs from the public and expand the understanding of diagnoses, medicines, and claims. There are almost 140 potential healthcare uses of Twitter [31]. The most common uses are: disaster alerting and response, diabetes management, drug safety alerts from the Food and Drug Administration, biomedical devices data capture and reporting, shift bidding for nurses, and other healthcare professionals, diagnostic brainstorming, rare diseases tracking, and resource connection, smoking cessation assistance, infant care tips to new parents and post-discharge patient consultations and follow-up care [15].

Another example of how social networks portray medical information, even when the safety and effectiveness of the human papillomavirus (HPV) vaccine proves, social network trends report low efficacy in some countries, including the United States. However, the news, celebrities, and trend-setters promote these negative opinions and information and impact public trust on this topic [32]. Because of the impact of social networks on healthcare, there is a growing interest in model development and its analysis. Prieto et al. (2014) study uses machine learning techniques to categorize tweets. They use regular expression in Spain and Portugal for data gathering the tweets into four categories: "Pregnancy," "Depression," "Flu," and "Eating Disorder" and apply two traditional machine learning methods K-Nearest Neighbors (KNN), Support Vector Machine (SVM) [6]. Prier et al. (2011) proposes a model based on the LDA model and set the model to generate 250 topics; they select "Tobacco" as a topic to validate the model [7]. Two other studies, one in the U.K. and the other in the U.S. finds the correlation between twitter's sentiment analysis and the quality of healthcare services [8,9].

2.2 Spatio-Temporal Analysis

Researchers use twitter data for many purposes, such as detecting opinions, risk analysis, and event detection [33-35] [22]. One of the valuable comments in social media is eyewitnesses generated text [36]. Researchers designed a model to detect eyewitness tweets to extract helpful and accurate information for analysis. This kind of tweet applies to different practices, such as management, recommendation, and monitoring systems in various areas, including natural disasters, city traffic, and healthcare. Another research uses location base social network (LBSN) data to detect anomalies and visualization [37]. This system applies and synthesizes information from an earthquake in the US in 2011, the

London riots, and hurricane Irene. They leverage a visual analysis method that assesses spatio-temporal anomalies and achieves more dynamic results than when performing clustering on Twitter data streams.

Some practical use of Twitter data is early disaster detection, covering the news, and understanding peoples' perceptions. Dictionary-based text analysis and Latent Dirichlet Allocation (LDA) is an unsupervised topic modeling to give journalists better insight [38]. These techniques help in conducting empirical research for the 2012 presidential election. They concluded both methods generated valuable results, such as detecting which users on Twitter discussed both Barack Obama and foreign affairs and a tax policy problem related to the presidential candidate Mitt Romney.

Since velocity is an inevitable part of the LBSN datasets, we require a data management system. Hwang et al. propose scalable pipeline infrastructure and a database schema [33] to collect and analyze data. They present a case study for flu risk surveillance and gather and process their data using twitter stream data and pipeline infrastructure to detect a flu epidemic in the United States. Rathore et al. proposes a Hadoop-based framework to perform real-time analysis on social media data [39]. They made this framework general and detect various events ranging from natural disasters to healthcare information. Tagging recommendations [40] and identified scientific topics in documents [41] are other Twitter data usage.

Recently a real-time social media processing framework named SENTINEL, focuses on disease surveillance introduce [41]. This research contains 9,353 twitters that they manually annotate and apply classification techniques. Although this dataset has tremendous value, they did not propose any solution for scalability in case of a new event

appearing in the system. In other similar research, Koylu et al. proposed a framework for prepossessing, topic extraction, and pattern exploration [42]. Topic extraction uses Latent Dirichlet Allocation (LDA) techniques and cosine similarity to calculate the pair-wise similarity between topics.

2.3 Named Entity Recognition

BioBERT conducts research related to our work applying named entity recognition to deep learning [43]. Its design for the multi-task named entity recognition, relation extraction, and question answering. BioBERT reports 0.62% improvement for medical NER. They benchmark their proposal over nine annotated databases and the lowest F1-Score is 71.11% in the JNLPBA database for detecting Gene/Protein, and 93.44% the highest F1-Score for BC5CDR when seeing Drug/Chem names. We compare our proposed architectures with the BioBert result over five standard databases for the NER task.

Wang Et al. studies the multi-task learning for Named Entity Recognition on a biomedical text by combining the character level and word [44]. They propose three architectures, Multi-Task Model Character (MTM-C), Multi-Task Model Word (MTM-W), and Multi-Task Model Character-Word (MTM-CW). MTM-C uses character level and MTM-W uses word-level embedding. The MTM-CW merges Character-Level embedding and Word-Level embedding. All these models will end up in a CRF layer to learn the sequence and identify them in a sentence.

In contrast with our research, we focus on sharing the learning by keeping the Embedding and BiLSTM fixed. We concentrate on multi-task learning on the CRF layer and learn the word segmentation and entities and combine the learning values for better outcomes. We

study the effect of different embedding layers on different BINER architectures and compare our results over four other databases.

The other related work by Marcove Et al. [45] focuses on multi-task learning (MTL) on Opinion Role Labeling (ORL). Their research approach is like the proposed architecture in our research. However, we define the loss function and tune the models differently for different databases. The science they focus on in one stage and followed achievements do not directly affect our results.

Ma & Hovey At Carnegie Mellon University [46] research the same neural network proposed by Chiu & Nichols [47] with an additional Conditional Random Fields (CRF) layer. They offer a model without feature engineering and reduced prepossessing cost. They claim this model is helpful for an extensive range of sequence labeling. This model evaluates two tasks, part of speech tagging and NER with 97.55% and 91.21% F1-Score. Their research focuses on the general, non-domain-specific named entity recognition. They combine the conditional random fields with BiLSTM layers, making it a relevant case to compare with our work. We train their network over the standard biomedical databases and report their F1-Scores to compare with our results.

The other aspect of our research focuses on different Embedding systems over Biomedical text sources. Several research studies show that studying word distribution is not enough for NLP tasks, such as POS and NER. As an example, at IBM, Santos & Zadrozeny use Convolutional Neural Networks (CNN) at character level to improve the POS tagging problem [48]. They develop a language-dependent model and experiment with it in English and Portuguese with the reported accuracy of 97%. Chui & Nicholas at British Columbia present a NER model [47] using bidirectional long short-term Memory

(BiLSTM) on top of a character-level CNN; their model obtains an F1-Score of 91.62%. In this research, we compare four different embedding layers: RNN-Character level, CNN-Character level, LookUp, and Self-attended Encoder, to compare the effect of these layers on biomedical text sources.

2.4 Deep Learning

2.4.1 Bidirectional Long Short Term Memory

Detecting the entity of a word or a phrase depends on the entity we use. To find a correlation between words in a sentence using a traditional neural network would be ineffective because of the complexity of a sentence. We define a sentence as a sequence of words. The recurrent neural network (RNN) architectures design is an excellent choice for learning sentence sequences and analyzing. RNN is used in language modeling [49, 50]; however, one drawback of using a simple RNN is the difficulty in capturing long-distance correlations in a sequence. We know this problem as the vanishing gradient problem.

Thus, they introduce the long short-term memory (LSTM) model to overcome the RNN limitation. In the LSTM network, it can transfer processed sequences to a further stage if needed, providing a time frame representing the word lines' word locations. An entity is a set of related words. A relevant word or words will identify each entity in a sentence. We use the BiLSTM model to bring the information in two directions.

2.4.2 Embedding Layer

The embedding layer creates a mapping between words and dense numerical matrices in natural language processing (NLP) to generate proper inputs to an artificial neural network (ANN). We study this layer of words and characters. First, word-level embedding,

representing each word as a vector, requires a vocabulary of words gathered from all the documents, named as V with size $|V|$, and D defined as embedding size, representing the size of a vector describing a word. Therefore, we have a matrix with size $|V| * D$, where each row refers to an individual word behaving like a lookup table. At this level, we apply a pre-defined word-level embedding such as Stanford's GLOVE or Google's Word2Vector [51, 52]. Kitaev & Klein's research on the effectiveness of the encoder on performing the parser and presented Self-Attentive Encoder (SAE) [53] proves valuable to Peters et al. when they introduce a model that enhanced the SAE and the Elmo (Embedding from Language Models) created [54].

Researchers show that studying morphological features of words such as word suffix and prefix would add value to the analysis [46, 48]. We know this type of text representation as character level. This research uses two types of character-level embedding: recurrent neural network (RNN) and the second convolution neural network (CNN).

For some problems, such as online learning, the need for creating a dictionary before training can be a nuisance. Researchers often solve this need with feature hashing, where a hash function is used to assign each token w ∈ T to one of a fixed set of "buckets" $1, 2, \ldots B$, each of which has its embedding vector [55]. A hash function is a function which maps data of arbitrary size to fixed-size values. For example, in a set of three documents containing 100, 200, and 500 words, the hash function will map the data from these documents into strings of 50 characters each, corresponding to the feature or "bucket."

The aim of hashing is to reduce the dimensionality of the token space T; it is common to have several buckets that are significantly smaller than the number of unique tokens. Bucketing inevitably results in many tokens "colliding" with each other because they

assign to the same bucket (an event more likely to occur with the 500-word document, for example). When multiple tokens collide, they will get the exact vector representation, preventing the model from distinguishing between them. A hash embedding using as an interpolation between a standard word embedding and a word embedding created using a random hash function (the hashing trick).

2.4.3 Conditional Random Field Layer

The Conditional Random Field (CRF) [56] uses segmenting and labeling sequence data. Researches show that the CRF technique at the sentence level achieves better results compare to individual word analysis such as Maximum Entropy Markov Models (MEMMs) and Hidden Markov Models (HMMs) [57, 58]. This CRF characteristic draws the researcher's attention to using this technique for NER tasks. CRF combined with Bidirectional Long Short Term Memory (BiLSTM) could improve sequence analysis outcomes [59]. Figure 5 explains how these two layers can contribute to each other. The figure illustrates how a sentence at the bottom of the figure goes through the LSTM blocks, and then the CRF layer predicts the output, which is the entity type of each word.

We define the CRF loss function as a matrix of scores, where A is a sentence formed as a sequence defined by $[[x]_1^T]$, then we have a function to calculate the scores $f_\theta([x]_1^T)$. In the scoring matrix, we defined each element by $[f_\theta]_{i,t}$ Where it is indexing tag outputs and t addresses each word, this matrix represents a position-independent scoring system. Equation (4) presents the scoring definition [59].

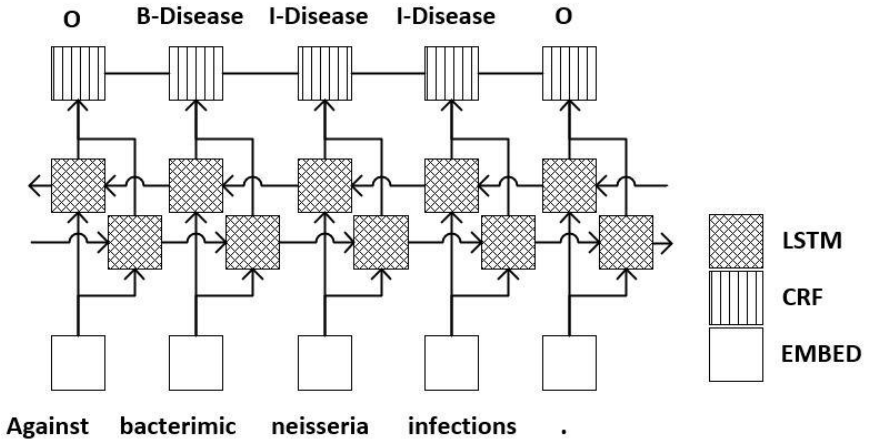

Figure 5 Shows the Transformation of a Sentence to output by connecting embed, bidirectional Lstm, and CRF extensions

$$s([x]_1^T, [i]_1^T, \theta) = \sum_{t=1}^{T} ([A]_{[i]_{t-1},[i]_t} + [f_\theta]_{[i]_t,t}) \quad (4)$$

CHAPTER 3

DATA SOURCES AND PREPARATION METHODOLOGY

3.1 Data Source Collection

3.1.1 Publication Data Set

There are several publicly available standard datasets to train and evaluate NER tasks. Table 1 shows a list of datasets used in this research with some of their features. We select five different databases in the biomedical domain besides the LINNAEUS dataset, which focuses on species to challenge our approaches and evaluate them for cross-domain adaptation. We use the BIO format (Begin, Inside, Outside) to prepare the datasets and tag entities. We preprocess and separate all the datasets into three files. The train includes 70%, validation contains 10%, and the test comprises 20% of the dataset.

JNLPBA [60] dataset is driven from GENIA corpus and annotated to be used as ground truth to detect *Protein, RNA, Cell Line, Cell Type,* and *DNA*. BioCreative II Gene Mention (BC2GM) [61] is a benchmark dataset to train for NER tasks and has been used by several researchers to detect gene names such as BANNER, GLIMI, and BIOBERT created by [43, 62, 63] respectively. Pathway Curation was the main task in BioNLP 2013 [64];

this dataset was designed to tackle the event extraction in a medical text to support curation. Before conducting any event extraction, we identify entities, and use this dataset to detect *Gene or Gene Product*, *Complex*, *Cellular-Component*, *Simple-Chemical*. LINNAEUS [65] dataset normalizes and recognizes species' names mentioned in a medical text, the version in this research contains 100 random full-text documents converted to standoff format.

One of the leading entities in the medical text is disease names. We select NCBI-disease [66] test to evaluate our models on this task. This dataset contains 793 PubMed abstracts and includes 790 unique disease names. The BioCreative V Chemical Disease Relation (BC5CDR) [67] dataset combines two entities (chemical and disease) and extracts humans from 1,500 PubMed articles. More databases are available, but we select these six databases to cover different types of entities and evaluate our neural network architectures in other areas. The summarization of all the datasets is shown in

Table 1.

Table 1 Type and Frequency of Entities in Each Dataset

Dataset	**Type**	**Entity Frequency**
JNLPBA	Gene/Protein	DNA: 8,392 protein:27,032 cell type: 6,177 cell line: 3,380 RNA: 837
BIONLP13PC	Gene	gene: 5,399 complex: 719 cellular: 464 chemical: 1,155
BC2GM	Gene	gene: 15,047
LINNAEUS	Species	species: 2,103
NCBI-DISEASE	Disease	disease: 5,118
BC5CDR	Disease/Chem.	chemical: 5,185 disease: 4,098

3.1.2 Social media Dataset

This section introduces two databases. The first public database captures spatio-temporal health care trends detection [21, 23]. The second one collects specific live kidney donations; this database helps researchers study the motivations for people who want to be live kidney donor.

3.1.3 Social media Spatio-Temporal Dataset

A stream collection process runs for nine months, from October 2018 until May 2019, resulting in 765,715 collected tweets. Table 2 reports the distribution of the tweets for two of the higher and lower states over the observation period. Florida was the state with the most significant number of tweets, and Wyoming had the lowest tweets.

Table 2 Most Representative States in Terms of the Total Number of Tweets Over the Observation Period

State	**2018**	**2018**	**2018/**	**2018/**	**2018/**	**2019**	**2019**	**2019**	**2019**	**Tota**
Florida	1,24	8,86	997	7,946	4,838	7,69	1,29	24,7	15,2	92,8
Californi	1,75	9,99	1,332	7,570	4,496	7,56	8,33	4,10	3,66	8,81
South	4	52	4	106	57	126	256	301	122	1,02
Wyoming	19	82	8	44	25	73	127	135	116	629

After collecting the data on a server, the framework cleaning module will automatically clean and filter the twitter message. Then we and a team of graduate students annotate a portion of the database (1,275 messages) and use them as ground truth for evaluation metrics. Labeled data was required to calculate the homogeneity, completeness, and v-measure. The statistics of the eight most common topics in the annotated database represented in Table 3.

Table 3 The top 8 most common generated topics in 1,275 tweets annotated by graduate students

Topic	Count	Percent
not related	251	20%
innovation, health tech, health it, tech, iot, medtech, mhealth, cybersecurity, bigdata, machine learning	126	10%
cancer, bcsm, week, day, share, love, risk, childhood cancer, read	76	6%
industry, medicine, future, check, digital, pharma, learn, blog, big, global	71	6%
Digital health, data, technology, health tech, innovation, blockchain, tech, industry, digital, iot	65	5%
pregnancy, pharma, migraine, women, pregnant, love, baby, biotech, pain, fda	63	5%
diabetes, news, research, education, free, women, pregnancy, information, study, food	61	5%
patients, hospital, access, services, providers, quality, hospitals, home, telehealth, doctor	56	4%

3.1.4 Clinical Note Dataset

Medical Information Mart for Intensive Care (MIMIC) Version 3 is an extensive database of electronic patient health records. Data gather in Beth Israel Deaconess Medical Center between 2001 and 2012. This extensive database, which it is freely accessible, encompasses forty thousand patient electronic health records. The MIMIC information includes laboratory test results, vital sign measurements made at the bedside, procedures: medications, caregiver notes, imaging reports, and mortality.

ShARe/CLEF eHealth generates a data set in three tasks to help better information retrieval in natural language processing with an approach to clinical care. Tasks 1 and 2 relate to annotating the clinical notes, and the third one relates to web pages based on queries generated when reading the clinical reports. Both Task 1 and 2 individually have 200

annotated clinical words for training and 100 clinical information for the test. Task 1 includes annotations of disorders, and Task 2 includes acronyms/abbreviations. Task 3 has a collection of medically related web documents, five development queries, and the result set of web documents and 50 test queries. [68, 69]

3.1.5 Meta Data Sources

We use two datasets to evaluate the effectiveness of the results obtained by following the methods:

(1) U.S. National Library of Medicine creates a free access database named Unified Medical Language System (UMLS). This database integrated 2 million names for some 900,000 concepts from over 60 families of biomedical vocabularies and 12 million relations among these concepts. UMLS connects multiple domains such as Genome annotations, Biomedical Literature, Genetic knowledge bases, Clinical repositories, and other sub-domains [70].

(2) Medical WordNet (MWN) is a lexical database. This is part of an extensive project named WordNet (W.N.). This database combines a dictionary and thesaurus, so we have some definitions of English and related words. The MWN as a part of W.N. consists of the medical term used by experts and non-expert. For instance, we search "chickenpox" on online access to this database, and we get the result as below:

> **chickenpox**, varicella (an acute contagious disease caused by herpes varicella zoster virus; causes a rash of vesicles on the face and body)

we can see "varicella" as a scientific world of chickenpox and a short explanation of the disease [71]. These datasets are widely used in the literature and provide a way to compare the result more easily.

3.2 Data Cleaning Module

Figure 6 illustrates the tweet cleaning and tokenization process. In this research, we focus on healthcare in the USA. We set English in API's language properties and assign health-related hashtags for the first iteration of the cleaning process. We target the USA and use API's Location properties and Census information. The other cleaning aspect is retweeted messages because they do not add value to analysis, we remove all those duplicated messages.

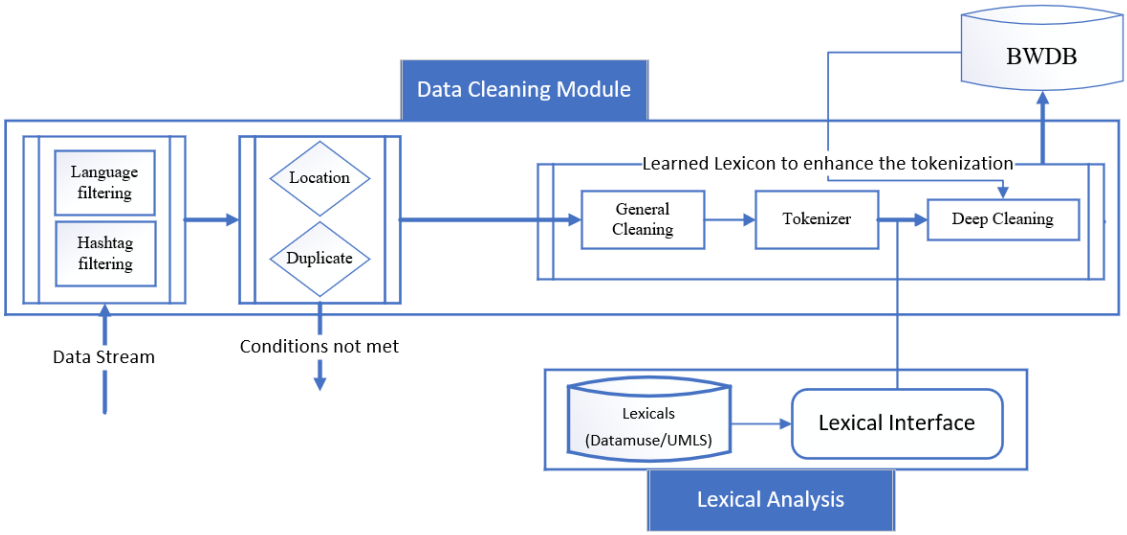

Figure 6 Data Cleaning Module – Input Collected Tweets and Return List of Words Cleaned in Two Layers of General and Deep

We use regular expressions to perform general cleaning. We create several expressions to remove IPs, URLs, HTML tags, and repeated characters, for example, “hellllllooo” which

corresponds to “hello.” We also apply extra cleaning steps like stemming, lemmatization, and tokenization to enhance the data quality. Implementing these steps requires metadata and expert knowledge. We use trustworthy lexicons, such as Datamuse [72] and UMLS [70], provided by NIH. Deep cleaning module, use the Datamuse lexicon for spell checking. Also, the UMLS lexicon helps us solve the joined and separated words in the healthcare domain, which we called this problem the "*Connected Words*" problem. These affect the bag of words and the model’s reliability; for example, “breast cancer” should transform to “breastcancer.”

Finally, we store the words in the *DWDB* and use them as an internal lexical, which gives feedback to the next iteration of the tokenization process. This database would enhance and speed up deep cleaning by using previously generated knowledge of words.

CHAPTER 4

HEALTH TREND DETECTION AND SOCIAL MEDIA SPATIO-TEMPORAL ANALYSIS

4.1 Introduction

Topics related to healthcare have grown in interest on social networks. On Twitter, plenty of healthcare-related comments are generating. These posts contain public opinions on health, and help us understand popular perception on medical diagnosis, medicines, facilities, and claims. Real-time processing and learning of conflicting data, proves challenging and tackling this challenge receives a lot of attention, especially considering this data comes from different perspectives, locations, and times, in a dynamic environment.

This section presents an adaptive system combination of unsupervised and supervised algorithms to track the trends of health social media to extend the system for real-life trend detection. We introduce a framework for managing, processing, analyzing, detecting, and tracking topics in streaming data.

We propose a topic tracking neural architecture that presents the accuracy of 83.34%, the precision of 83%, recall 84%, and F-Score of 83.8%. Our results compare with two state-of-the-art techniques demonstrating an advantage. [73]

We introduce a model selector procedure with a hybrid indicator to tackle online topic detection. In this framework, we built an automatic data processing pipeline with two levels of cleaning. Regular and deep cleaning are applied using multiple sources of meta knowledge to enhance data quality. Deep learning and transfer learning techniques classify health-related tweets with high accuracy and improved F1-Score. In this system, we use visualization to understand trending topics better. To demonstrate the validity of this framework, we implement and apply it to health-related twitter data from users originating in the USA over nine months. This implementation shows that this framework could detect and track the topics comparable to manual annotation. The result will display graphically on the United States map to better explain the emerging and changing issues in various locations over time.

4.2 Social media topic modeling tracking

The proposed model builds five layers. We collect data using a python tool called Teewpy [12] on the first layer. The second layer contains the cleaning and preprocessing methods described, converting the tweets to vectors that can be processed. The third layer is a Word2Vec method that creates a matrix based on the vector received by the last layer and uses that matrix to initialize a neural network to predict the labeled tweet. A CNN classifier is the fourth layer, where unseen tweets coming from the Word2Vec are labeled. Usually, sequence modeling relates to recurrent neural networks (RNN). However, the results indicate a different perspective. The convolutional neural network (CNN) provides notable results in NLP [13]. Yih et al. 2011 applies CNN to semantic parsing [14], Shen 2014 uses it for query retrieval [15], KalchBanner 2014 uses it for sentence modeling [16], and Yoon

Kim 2014 connects the Word2Vec model with CNN [13]. As part of this research project, we explore classification models that we can use for an adaptive system. Feature selection is one challenge, and the Convolutional Neural Networks (CNN) are appropriate, as they do not require a priori feature selection. A limitation of CNNs is that they need a fixed input size; as the tweets limit to 280 characters, we can use padding for shorter tweets preserving the fixed input sizes. Our architecture comprises three convolutional layers with a 128, 64 and 32 kernel sizes, respectively. The system has a drop out of 0.5. We iterate it by 200 epochs using batches of 100 records with a learning rate of 10^{-5}. We obtain the weights using Adam-Optimizer. Some tweets may not fit into the selected topics on this layer, and these are flagged as unlabeled data and stored in an independent dataset. If the tweets do into the selected topics, they are labeled accordingly.

In the fifth layer, we feed unlabeled data to an LDA model and creates new topics. The CNN model will train again, updating both the issues and the Word2Vec model. Figure 7 shows the system's architecture. This system provides active learning of topics and reinforces the CNN model classification.

We compare our system with SVM and CNN techniques independently to predict and label new tweets. However, the prediction capacities from both methods were limited because of the unbalanced datasets. The results using the prediction metrics shown in Table II.

4.3 Proposed Topic modeling Framework

The dynamic behavior from Twitter data, given the velocity characteristic at different times and locations, makes the analysis challenging. Designing a supervised methodology would require lots of resources and time for annotating the data. Thus, we need an unsupervised learning approach for tracking the topics. Figure 7 illustrates the proposed Twitter stream

analysis framework for automatic preprocessing, unsupervised topic modeling, knowledge representation, and visualization.

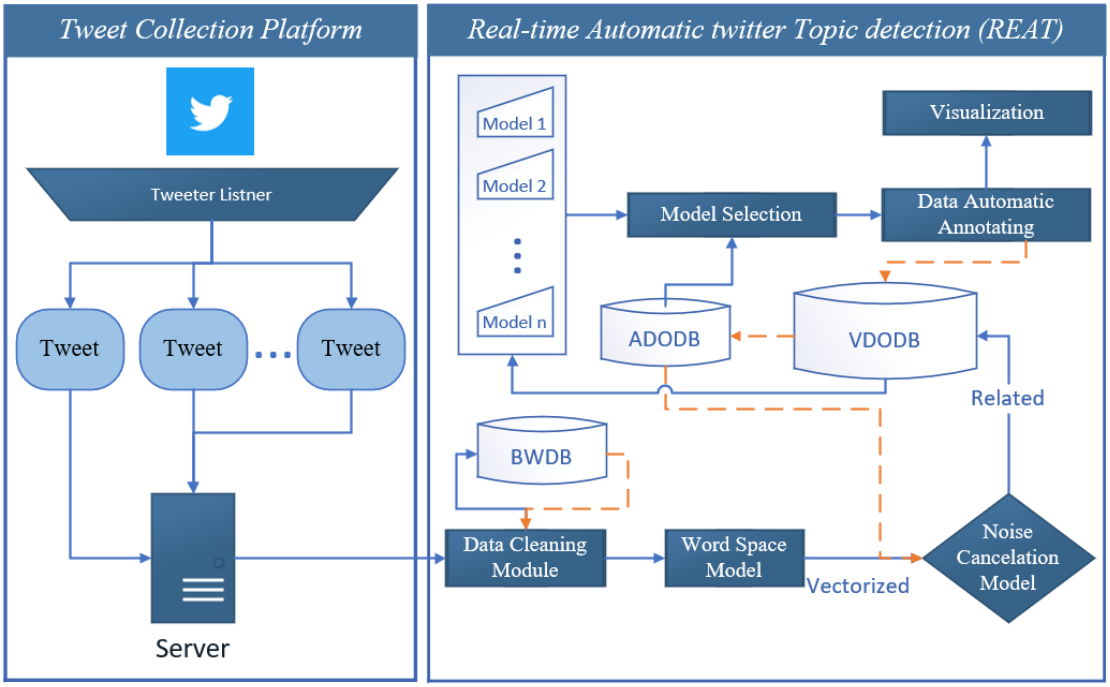

Figure 7 Real-Time Twitter Topic Modeling Architecture and Software Modules

This system forms two significant modules, *"Tweet Collection Platform"* and *"Real time Automatic Twitter Topic detection"* (*REAT*). *Tweet Collection Platform* implements a listener using Tweepy API, collecting comments and storing them on a server to analyze by the REAT module.

The *REAT* consists of five sub-modules, including data cleaning, word space, related tweets, model selector, and visualization; each module will explain each in detail.

The *REAT* defines three databases. First is the "*Bag of Word DataBase*" (*BWDB*) to store the tokens. The *BWDB* will grow incrementally, helping to improve the tokenizer procedure, which is implemented in the Data cleaning module. The second is the "*Vectorized DOcuments DataBase*" (*VDODB*); this database will store the data after

transforming and removing the noises, using the "*Word Space Model*" and *Noise Cancellation Model*." The third is the *Annotated DOcument DataBase*" (*ADODB*) to boost the topic modeling selector. This database stores annotated twitters. *ADODB* will help the framework calculate homogeneity, completeness, and V-measure; these metrics will discuss later. To generate the ground truth, we annotated a random sample of collected data by assigning a graduate student team and storing it in *ADODB.*

We create a bridge between vectorized Twitter database and an annotated database to respond to the model selector and provide a confidence level threshold to decide which part of the data sent to the *ADODB.* We recommend having an expert interfere in this level to increase the quality of *ADODB* by checking on annotated messages.

4.4 Noise Cancelation Model

A word-space model is a method that converts text to a vector. There are several ways to implement this model. One solution is using one-hot encoded vectors, but this representation will not be satisfactory to our purpose. Since each vector needs an array size of created vocabulary (BWDB), this technique is not practical relative to the size of the vocabulary and the dynamic of the environment.

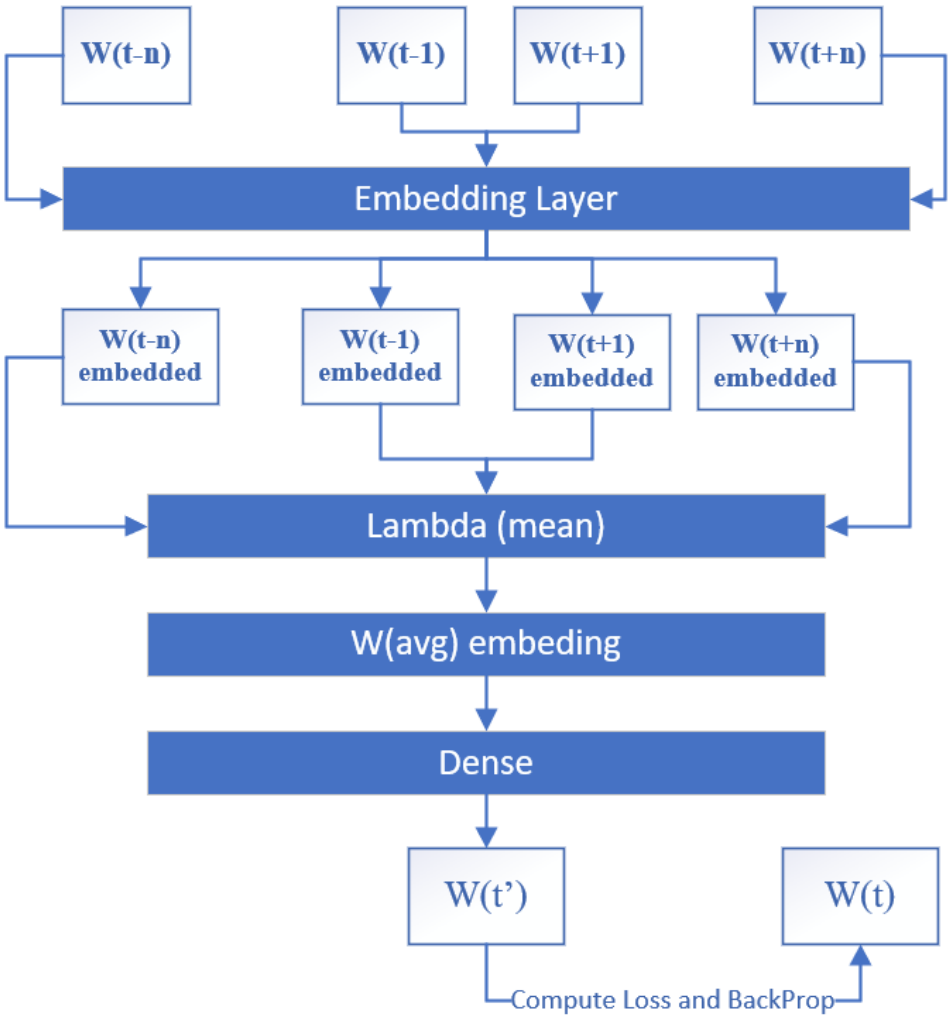

Figure 8 CBOW Deep Learning Layers

The continuous Bag of Words (CBOW)[52] technique can solve the vocabulary size problem by defining the vector dimensionality. Figure 8 represents the CBOW Deep learning model structure used as Word2Vec in this study. N represents window size, and W is a word, $W(t)$ will be a word inside a document in position t, and create a sequence of $W(t-n)\dots W(t-1)W(t+1)W(t+n)$. We assume V is the size of Vocabulary and D is the dimensionality and create a matrix size of $V * D$. This matrix is known as Embedding Layer.

To create the Embedding layer, defining a loss function is essential. If we assume the predicted target word is $W(t')$ and the actual word is $W(t)$, we can define the loss function. We implement a Dense layer (size of Vocabulary) with a Softmax activation to predict the target $W(t')$. Table 3 presents an example of input and target sample by window size 2, representing two words before and after the target word. The last update of the weight in the Dense layer represents each word as a vector length of D.

Table 2 Sample Input and Output Target in the CBOW Model

Input	***Target***
'opposite', 'think', 'cancer', 'replace'	roll
'think', 'roll', 'replace', 'control'	cancer
'roll', 'cancer', 'control', 'month'	replace
'cancer', 'replace', 'month', 'provide'	control
'replace', 'control', 'provide', 'let'	month
'control', 'month', 'let', 'cell'	provide

4.5 Model Selection

To detect a topic, we extract the main ideas discussed in a text. These ideas represent themes, subjects for discussion, or conversations. At this stage, we process data in different granularities, such as the topic of a sentence, a paragraph, or an article. Twitter's messages are like a sentence; therefore, we analyze the data with sentence granularity in this research. We select Latent Semantic Analysis (LSA) as a classic algorithm, Latent Dirichlet Allocation (LDA) as the most used technique, LDA-MALLET as an enhanced version of LDA, and Biterm Topic Modeling technique as a specific targeted short message topic modeling, to evaluate the proposed framework for selecting a proper model for topic

modeling. This section discusses each of these techniques individually, and then we explain the evaluation metrics.

4.6 Spatio-Temporal Visualization

After calculations and decision-making to create topics, we need to represent and visualize the trend over the tweets to make the framework represent trends track over location and time. The trend detection process is better described visually to perceive the topic trend over time and across areas. We use a color-coding structure, which will represent these factors for a given topic. RGB channels are utilized for color-coding and data structure, as presented in Figure 9. Each row represents the location change over time. In addition, the RGB color mapping is defines three channels per row.

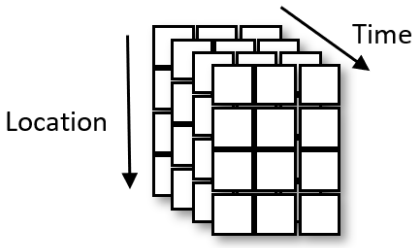

Figure 9 Tensor of Data Representation

Processing the data and running the topic modeling detect trends. These topics will change based on location and time frequency. We assume N as the total number of generated topics and n as the topic, location is l, and time is t, then the topic frequency in a specific location and time is $(n)_l^t$. To calculate the influence of a state, we use Equation (5). In our color-coding, the Red channel is dedicated to the State Influence $SI(n)_l^t$. Detecting the red gradient on the map infer that a state has more influence on that topic.

$$SI(n)_l^t = \frac{f(n)_t^l}{\sum_{topic=1}^{N} f(topics)_t^l} \tag{5}$$

The next level of representation will be the influence of a trend over a country. To calculate this, we need to assess how many tweets are posted about that topic at a specific time. If we assume the total number of locations at L. $f(l)_n^t$ represents the frequency of tweets in a specific location (l) followed by a topic (n) and time (t). Equation (6) presents the country influence calculation.

$$CI(n)_t^n = \frac{f(n)_t^n}{\sum_{location=1}^{N} f(location)_t^n} \quad (6)$$

The green channel represents the states that are trendsetters for the entire country. Similarly, the yellow color means the topic affects the state and affects the country; however, they are not a trendsetter.

4.7 Evaluation Metrics

In this section, we explain two types of evaluation metrics. The first one measures the coverage of selected words as a topic over documents, such as coherence measurement. Second, involve the expert opinion in topic evaluation. [74] used V-measured to evaluate K-means' performance over document clustering. They show that the V-measure can be performed transparently over a distinct dataset. Approaching this technique needs manual labeling. This paper uses homogeneity, completeness, and v-measure to evaluate topic modeling techniques.

4.7.1 Coherence

Measuring coherence among members of a set or subset of words is based on probabilistic analysis. [75] defines a score for describing the quality of document groups generated by a model. A set of words represents a topic, and the proposed score is based on comparing these sets. We assume the pair for comparison defined by $S = (w', w^*)$ which w', w^* represents two sets of words where $w' \cap w^* = \phi$. Equation (7) defines log-likelihood, and Equation (8) defines log-ratio; these are basic elements of coherence calculation.

$$ll = \log \frac{p(w'|w^*) + \epsilon}{p(w'|\neg w^*) + \epsilon} \tag{7}$$

$$lr = \log \frac{p(w'|w^*) + \epsilon}{p(w') * p(w^*)} \tag{8}$$

In the Figure 10. and the outcome ranges from 0 to 1. However, in our research, we did not rely on unsupervised metrics. We also used supervised metrics for evaluating generated topics to select the robust model.

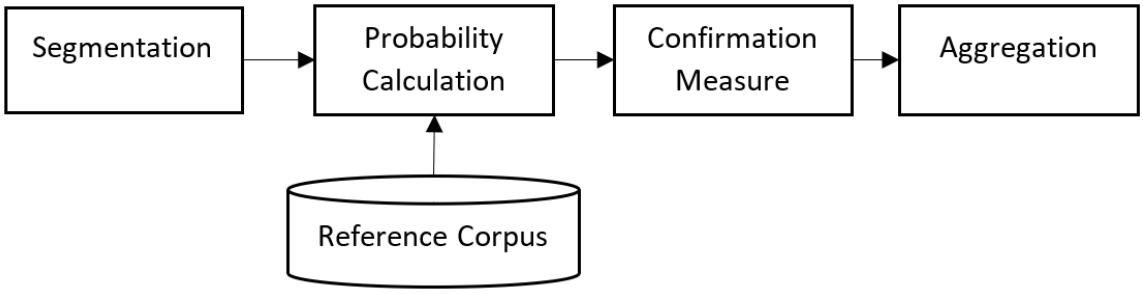

Figure 10 Coherence Algorithm Steps From Segmentation Through Aggregation

4.7.2 Homogeneity, completeness, and V-measure

Homogeneity explains the number of true-labeled elements in a single class. The goal is to have each cluster contain members of a single class. Completeness supplements uniformity by defining how many of the members of a single type are assigned to the same group. We define two metrics in a range from 0 to 1, which is the optimal performance indicator. The harmonic mean of these two measurements gives V-measure. [74]

These metrics determine how close a cluster is to its ideal definition by examining the conditional entropy of class distribution given the proposed clustering. Annotation of these functions is represented in Table 5:

Table 3 Annotations Definition in Homogeneity and Completeness Metrics

Description	**Annotation**
C as a Set of Classes	$C_{i_{1..n}}$
K as a Set of Cluster	$K_{j_{1..m}}$
Represent a member of class "c," which is an element of cluster "k"	a_{ck}
Number of Datapoint	N

$$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{a_{ck}}{N}\log\frac{a_{ck}}{\sum_{c=1}^{|C|}a_{ck}} \tag{9}$$

$$H(C) = -\sum_{c=1}^{|C|}\frac{a_{ck}}{N}\log\frac{\sum_{k=1}^{|K|}a_{ck}}{n} \tag{10}$$

Based on these formulas, we can calculate homogeneity with the following:

$$P_{GL} = \begin{cases} if\ H(C|K) = 0 \quad Then \quad h = 1 \\ else \qquad\qquad\quad h = 1 - \dfrac{H(C|K)}{H(C)} \end{cases} \tag{11}$$

$$P_{GL} = \begin{cases} if\ H(K|C) = 0 \quad Then \quad c = 1 \\ else \qquad\qquad\quad h = 1 - \dfrac{H(K|C)}{H(K)} \end{cases} \tag{12}$$

And as we mentioned before, we calculate V-measure with the harmonic-mean between homogeneity and completeness, so the calculation is:

$$V - Measure\ = \frac{(1+\beta) * h * c}{(\beta * h)\ +\ c} \tag{13}$$

We assume that β is equal to 1 then the calculation can be done in this way:

$$V - Measure\ = \frac{2 * h * c}{h\ +\ c} \tag{14}$$

We propose α as a dynamic decision-making metric to select the proper model among several models for topic detection. This measurement is the multiplication of V-measure and Coherence between [0,1]. In clustering, sometimes the probability of overlapping a set of words is high, Although they need to be captured as different topics. This situation happens in subtopics, for example, women as a big topic and sub-topic such as diabetes in women or pregnancy. The v-measure as a supervised evaluation can contribute to the metric to sharpener the decision boundaries. Therefore, besides V-measure, Coherence creates a combination to handle the dynamic of Twitter in topics detection. Equation (15) cr refers to coherence, and h refers to homogeneity; also, c represents completeness.

$$\alpha\ =\ cr\ * \frac{2 * h * c}{h\ +\ c} \tag{15}$$

4.8 Case Study

4.8.1 Noise Cancelation

We apply the CBOW deep learning model with a window size of 2 and a dimension of 100, leading to a matrix size of $169{,}394 * 100$, representing each word in a vocabulary of 169,394 words by vector length of 100. Then we apply the Noise cancellation model to filter unrelated messages. To evaluate the noise cancellation model, we use the annotated database (ADODB) to classify "related" and "not related" tweets, then we compare regular training on News20 [76] dataset with proposed transfer learning. This dataset contains a wide range of topics, and among all the topics, there is a group related to healthcare named “sci.med”, which has 990 documents. This topic was removed before we could train CBOW on News20 Dataset.

Using the transfer knowledge approach to select unrelated healthcare databases shows better results than the regular one. Classification results show that almost 10% improvement on F1-Score is achieved by transfer knowledge from other resources. The complete results of this classification over 20% Test dataset reported in Table 4.

Table 4 Regression Classification Results With and Without Transfer Learning

Class Label	Precision	Recall	F1-Score
Result in the Classification with Transfer Learning			
Not Related	0.82	0.86	0.84
Related	0.82	0.86	0.84
Result in the Classification without Transfer Learning			
Not Related	0.68	0.76	0.72
Related	0.81	0.74	0.77

4.8.2 Topic Modeling Selection

The next step is selecting the proper model to assign topics to the messages. To experiment with the model selector module, we repeat the experiment by a different number of issues t (in the range of 2 to 30). We report coherence in Figure 11 for each model based on another t. In our database, 30 topics using LDA-MALLET are the best. The coherence selects 30 as the optimal number of topics. However, the annotated dataset contains 28 individual topics and implies evaluating and making decisions based on coherence might not be accurate enough, and we need more evaluation and analysis over selecting the practical number of topics.

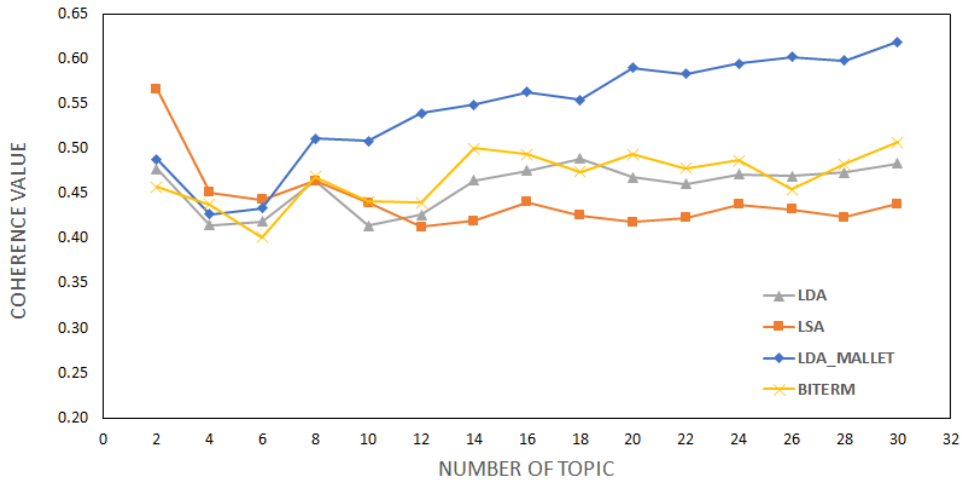

Figure 11 Coherence Measurement for LDA, LSA, LDA-MALLET, BTM

Our framework uses homogeneity, completeness, v-measure, and an annotated database as ground truth to evaluate the selected models. Based on these metrics, we observe two groups of models. The first is LDA and BITERM, which show the approximately same behavior. The second is LDA-MALLET and LSA.

We show the result of Homogeneity, Completeness, and V-Measure metrics in Figure 12, Figure 13, Figure 14.

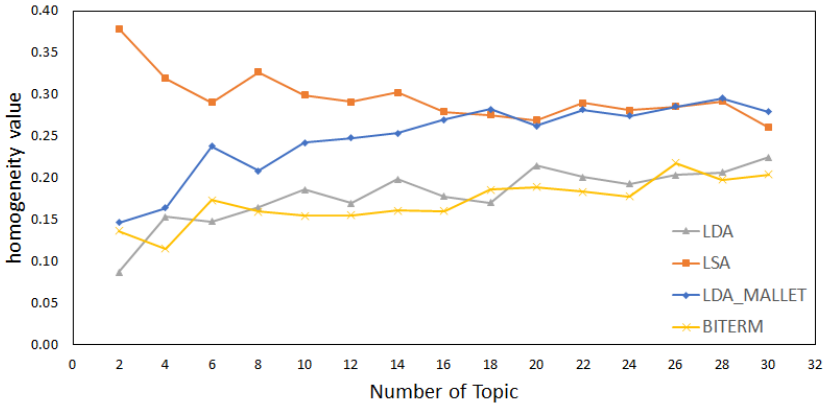

Figure 12 Homogeneity Over Annotated Database Based on the Model Created Over Vectorized Twitter Database

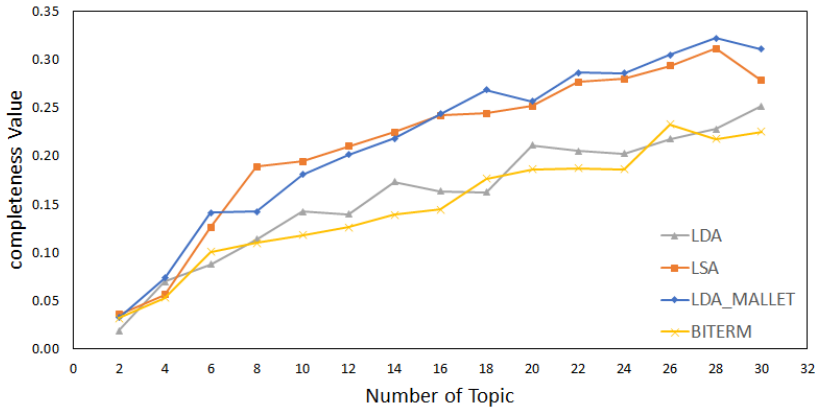

Figure 13 Completeness Over Annotated Database Based on the Model Created Over Vectorized Twitter Database

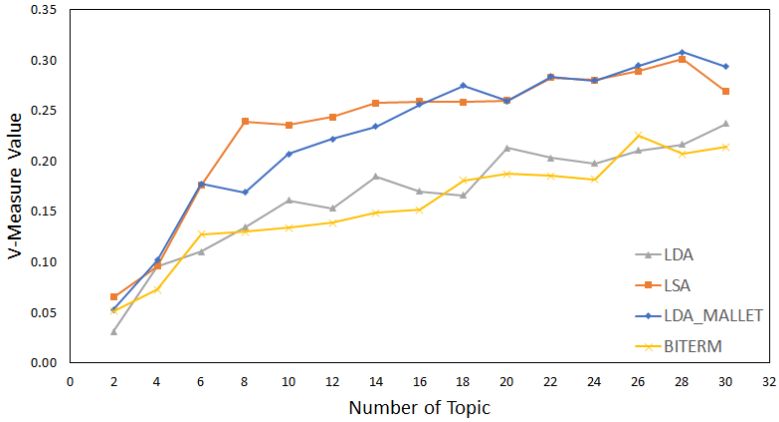

Figure 14 V-Measure Over Annotated Database Based on the Model Created Over Vectorized Twitter Database

This experiment not only will help us to choose several topics but also will represent which model performed more efficiently. The Alpha Value summarizes four metrics to let the model have a metric to rank the models. In Figure 15, we conclude that the LDA-MALLET shows better results compared to other models; also, we can see that 28 is the best number of topics. This metric helps us to understand the behavior of different models. Having this robust measurement will improve the quality and reliability of reports generated by this framework.

Although, part of this experiment needs experts to interfere and annotate the database. Experts could contribute by evaluating partially labeled messages. This can decrease the cost of manually annotated and help the system to do the self-correction in model selector by using expert feedback besides getting advantages from unsupervised properties.

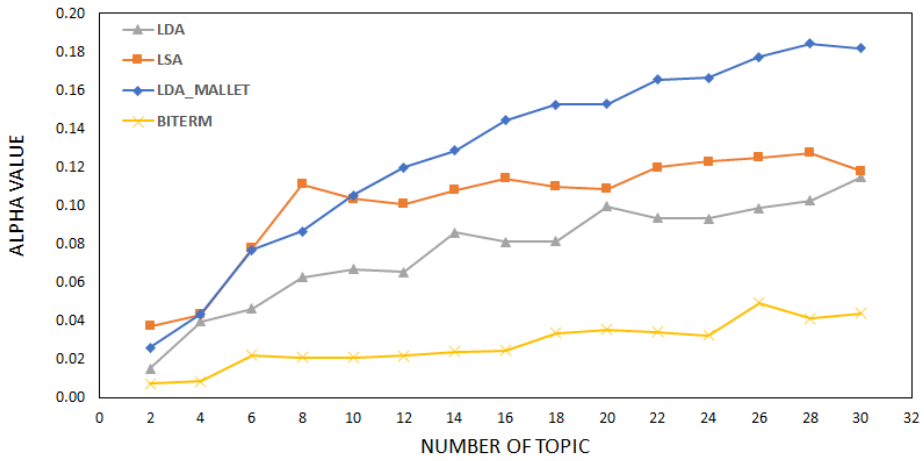

Figure 15 Calculated Alpha Over Annotated database based on the model created over vectorized Twitter database

We select trending topic related to people with diabetes that appear during this research study. In addition, media outlets such as US News discussed the insulin crisis, and there were general interest and research studies related to this concern as the one presented in [77]. Therefore, we found the "Cost of Insulin" as a demonstrative case study. We use color

to highlight the intensity of the trending topic over space and time distributed on the U.S map to visualize the results. To serve as an example, Figure 16 shows the topic interest change as time-lapse with monthly granularity.

We observe low saturation color (both red and green channel) over Alaska; this implies that this topic was not of concern for this state initially. As we move through time, we can observe that color migrates to red, meaning the topic interest grew. Florida and California from the beginning had green color, meaning they were trendsetters among all the states. We can also observe, Washington and Texas are colored yellow, meaning their influence is at the national level, but they were not trendsetters. If we put these frames in an animation, we can observe a movement of the topic from October 2018 to April 2019 for seven months. This movement started from Florida and California and spread up through the country until it reached Alaska. New York, Washington DC, and Chicago were already interested in this topic.

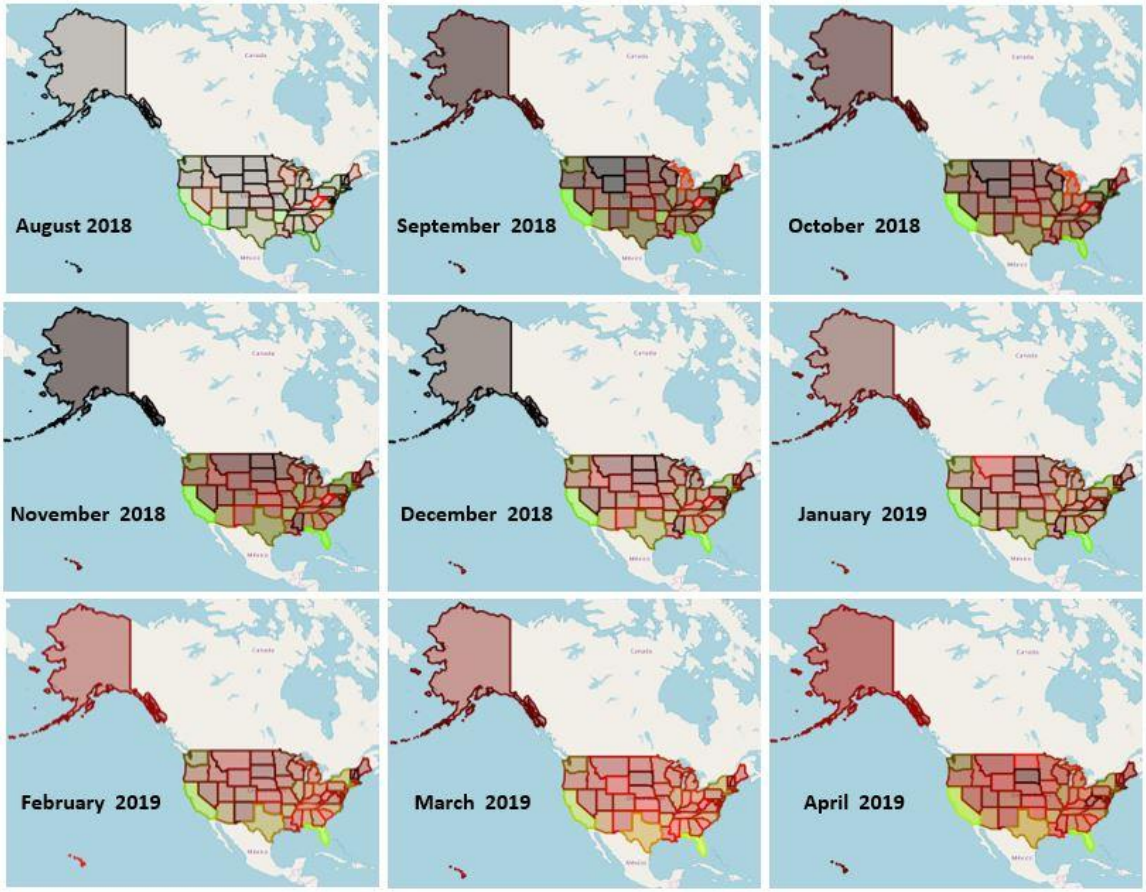

Figure 16 Tracking "Cost of Insulin" Topic From August 2018 Until April 2019

4.9 Conclusion

Topic detection has many applications, and its especially helpful in grouping scientific papers, understanding customers' concerns about a product, and detecting and tracking social media trends. These groupings can be helpful for many institutions, such as governments or news agencies. This paper proposes a general framework to perform a fully automatic text collection and cleaning, alongside a semi-automatic topic detection technique, by using hybrid evaluation metrics that assess the result regardless of selected models. We address two main issues, first automatic noise detection by using deep transfer learning techniques; second, we address the high cost of manual labeling and

annotation. We decrease the cost of labeling by creating an annotated database using entropy-based clustering measurements.

In this experiment, we focus on health-related tweets and represent the results to show the utility of our proposed framework. We measure our products using v-Measure, homogeneity, completeness, and coherence to select the number of topics and models. Besides, we visualize a sample of trend tracing in the United States, which shows the value of the analysis using this framework. This hybrid approach of using feedback from experts with our results improves performance in dynamic environments. This continuing learning approach addresses the domain shift problem in time and location as in COVID-19 pandemic analysis, as follows up on this research.

CHAPTER 5

LOW-COST BIOMEDICAL NAMED ENTITY RECOGNITION

5.1 Introduction

A natural language understanding (NLU) system needs to process the structure of a sentence for many purposes, such as question answering [78], topic modeling [21, 23], and automatic summarization [79]. Before we could get to these tasks, we need to solve and improve techniques such as part-of-speech (POS) detection and named entity recognition (NER) and allow the machines to learn more from text. Some of the traditional techniques, such as conditional random field and hidden markov models, require human intervention, including exact word spelling features and external resources generation like dictionaries and lexicons [80-82]. Using human resources would be expensive, and when dealing with Big Data, human resources are impractical. Deep learning techniques for POS and NER reduce the cost of human labeling and improve the model's performance [43, 47],[83].

Bidirectional Encoder Representation from Transformers (BERT) [84] is the most recent technique for many tasks, such as text classification, question answering, and NER. According to the BERT developers, training the Base version of BERT, 12 Layers and 768 Hidden sizes and 12 Self Attention heads (110 Million Total parameters) took four days to train using four cloud Tensor Process Units (TPUs). The Large version of BERT, 24 layers and 1024 hidden layers, and 16 self-attention heads (340 Million total parameters) took four days to train using 16 clouds TPUs. BERT's need one GPU for Base and one TPU for Large BERT in deployment. This model is a general language model, not a specific domain such as health care, which opens research and application to train domain-specific BERT. In the health care domain, BioBert [43] proposes and introduces as a pre-trained biomedical model. According to the authors, they utilize eight NVIDIA V100 GPUs for training, and it took 23 days.

In this research, we identify two challenges. First, the current state-of-the-art technique, BERT which requires extensive computational resources [43, 84]. Second, the original BERT implementation is not domain-specific, and therefore it could be further developed to model specific domains, such as biomedical, with improved outcomes.

We address these challenges and present experimental evidence of an alternative implementation from BioBERT by using different Embedding Layers combined with Bidirectional LSTM and Conditional Random Field to achieve domain specific NER in the Biomedical field. We improve the F1-Score over BioBert by 3% on Disease detection, 8% on detecting Protein (Contains DNA, RNA, and Cell), and 9% for Gene detection. Our results compare favorably with BioBERT when tested with five different standard databases.

5.2 Biomedical Named Entity (BINER) Approaches

We introduce three different BINER implementations using three Artificial Neural Network (ANN) architectures. We are experimenting with analyzing each architecture's results and comparing the best model with recent Named Entity Recognition on Biomedical text.

5.2.1 Base BINER Implementation

The first implementation seen in Figure 17 uses a shared BiLSTM layer to train the model and implement two CRFs (Base BINER). One for word segmentation, described by CRF_{iob}, and the other for sequence labeling CRF_{ner}. To optimize CRF models, we calculate the Log-Sum-Exp (LSE) for the sequence length of t using Equation (19).To aggregate the loss function for both word segmenting and labeling, we calculate Mean Square Error (MSE). The prediction results of individual CRFs are calculated by the LSE of CRF_{iob} and CRF_{ner}, are defined as S'_{iob} and S'_{ner} respectively. Besides, S_{iob} and S_{ner} are the ground truth score. We calculate MSE in Equation (20).

$$LSE(S) = x^* + \log(\sum_{t=1}^{T} \exp\,(x_t - x^*)) \qquad WHERE\ x^* = max_1^t\, x_t \tag{16}$$

$$MSE = \frac{1}{n}\sum_{t=1}^{n}[\big((s_{iob} + s_{ner}) - (s'_{iob} + s'_{ner})\big)^2] \tag{17}$$

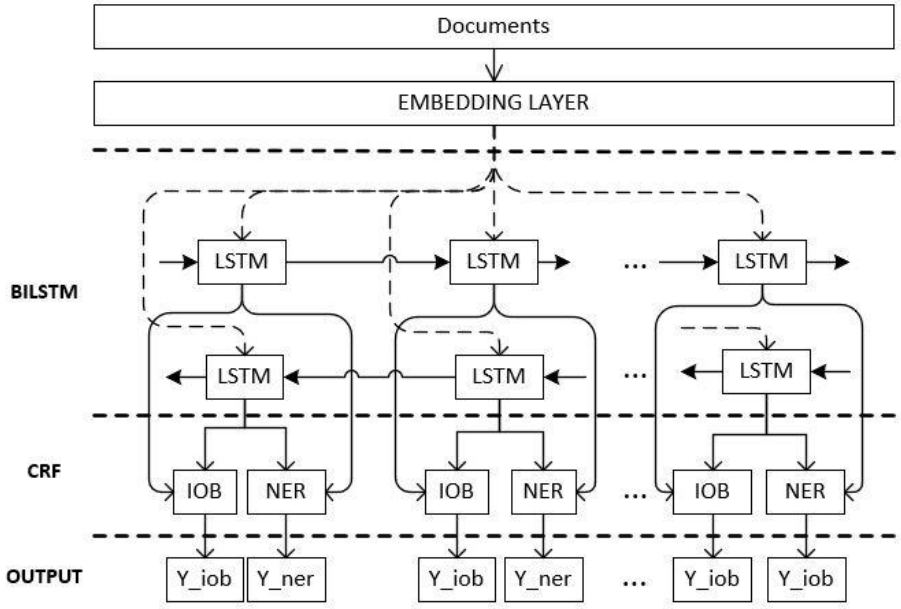

Figure 17 Connection Between Layers in Base-Biner Implementation

5.2.2 Parallel BINER Implementation

The second architecture contains individual BiLSTM and BiCRF. The idea behind this architecture is to integrate two deep learning models. The two models learn separately, and when combined, they learn from their mistakes together. This idea leads us to have parallel BiLSTM, one for segmenting words and the other for learning the labels in a sequence. Figure 18 shows the architectures in detail. This architecture follows the same loss function equation as represented in Equation (32).

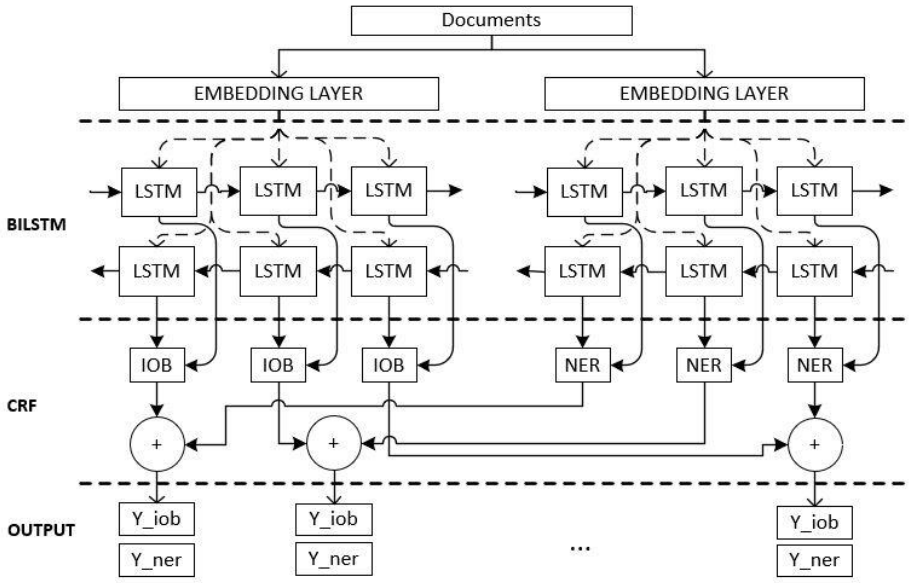

Figure 18 Architecture of Parallel BINER

5.2.3 Sequential BINER Implementation

The third architecture integrates the two layers in sequence for learning. In contrast to the parallel architecture, this one will use the first layer's knowledge, word segmenting, and feed that knowledge into the labeling layer. Therefore, we have a sequence of models, but they will not share the error. Figure 19 Connection of Layers in Sequence BINER Implementationillustrates how they are connected. This architecture has two outputs for the loss function: word segmenting and word labeling using Mean of Errors (ME). The ME calculate in Equation (21), where S' refers to the CRF score calculated for CRF_{iob} and CRF_{ner} separately, S is calculated based on the True Labels for both, and n represents the length of a given sequence.

$$ME = \frac{1}{n}\sum_{n=1}^{n}[(S - S')] \tag{18}$$

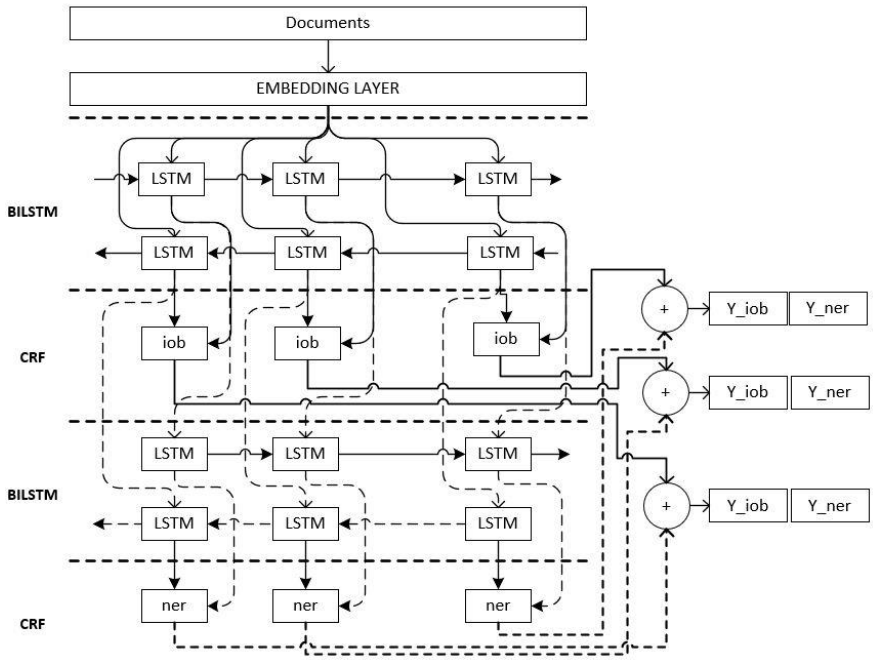

Figure 19 Connection of Layers in Sequence BINER Implementation

Sequence architecture requires multiple loss functions, so we use the *retain-graph* property in PyTorch v1.8 framework to back-propagate each loss function individually.

5.3 Evaluation Metrics

To train and evaluate the models, we split the data into three different sets—train, validation, and test. We use the validation dataset to assess the model while training. This dataset helps us have an indicator to select the best state of the model during the training. To be more accurate and remove the bias, we did not use the best-achieved F1-Score in training, but the Macro F1-Score select as our metric. Optiz & Burst discuss two types of Macro F1-Score, "Average F1" and "F1 of Averages" [85]. F1-Score described in Equation (21) which is the harmonic (H) mean of Precision and Recall calculated by Equations (19) and (20), respectively,

if we define $TP = True\ Positive,\ TN = True\ Negative,\ FP = False\ Positive,$ $FN = False\ Negative.$

$$P = \frac{TP}{TP + TF} \tag{19}$$

$$R = \frac{TP}{TP + FN} \tag{20}$$

$$F1 = H(P, R) = \frac{2 * P * R}{P + R} \tag{21}$$

The "Average F1" in Equation (22) and "F1 of Average" in Equation (23) are defined as follow:

$$F_1 = \frac{1}{n}\sum_x F1_x = \frac{1}{n}\sum_x \frac{2 * P * R}{P + R} \tag{22}$$

$$\mathbb{F}_1 = H(\bar{P}, \bar{R}) = \frac{2 * \bar{P} + \bar{R}}{\bar{P} + \bar{R}} = 2 * \frac{(\frac{1}{n}\sum_x P_x)(\frac{1}{n}\sum_x R_x)}{\left(\frac{1}{n}\sum_x P_x\right) + (\frac{1}{n}\sum_x R_x)} \tag{23}$$

Based on Optiz & Burst "F1 of Average" [85] is a heavily biased classifier, and in some cases, it can be misleading as an evaluation metric. This situation is more likely to happen when the data set is imbalanced. In this research, all the datasets are imbalanced then we report the result based on "Average F1" for simplicity; we name it F1-Score.

5.4 Case Study

5.4.1 Medical Literature

After defining the model's architecture, we use all the databases described in section 3.1.1 and evaluate the models based on evaluation metrics define in section 5.3. Hyperparameter optimization performs as seen on Figure 20. In this Figure, we split the database into three sections—train, eval, and test. Evaluation metrics decide which model should be saved,

and in the test section, we use it to select the winner model. In this process, we carefully follow the training process to avoid the trap of overfitting in our model; therefore, we report all the diagrams for each parameter that we tuned. In the end, we compare our result with the state-of-art techniques and conclude the best model with the values of the hyperparameters.

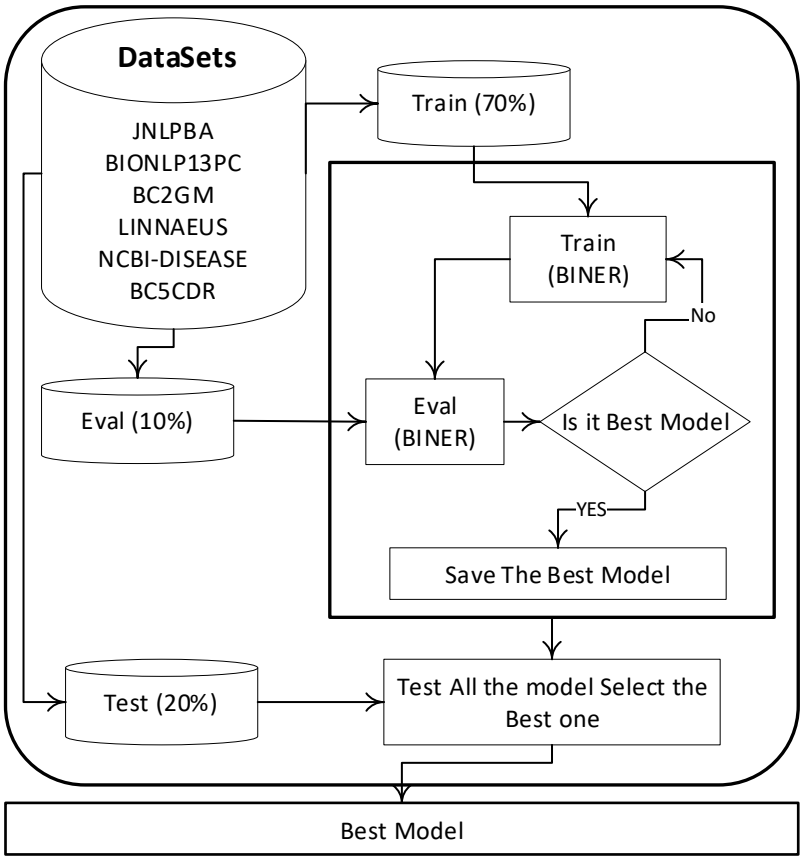

Figure 20 Process of Training, Evaluation, and Test a Model to Select the Best Model With Proper Hyper-parameters

5.4.1.1 BINER Hyper Parameters Optimization

- **EPOCHS**

'Epoch' is the number of iterations of the training set for a model. Choosing many epochs will increase the risk of over-fitting, and the model would end up losing generality. Therefore, choosing the correct number of epochs has a high impact on the model. Figure

21, Figure 22, and Figure 23 show the F1-Score over the number of epochs for different Embedding layers for the three proposed architectures using the NCBI-Disease database as an example. We conclude that most of the architectures stop learning after 300 Epochs. However, the lookup embedding layer shows a more stable result than the others, and we can see around 100 epochs that all the architectures are regular.

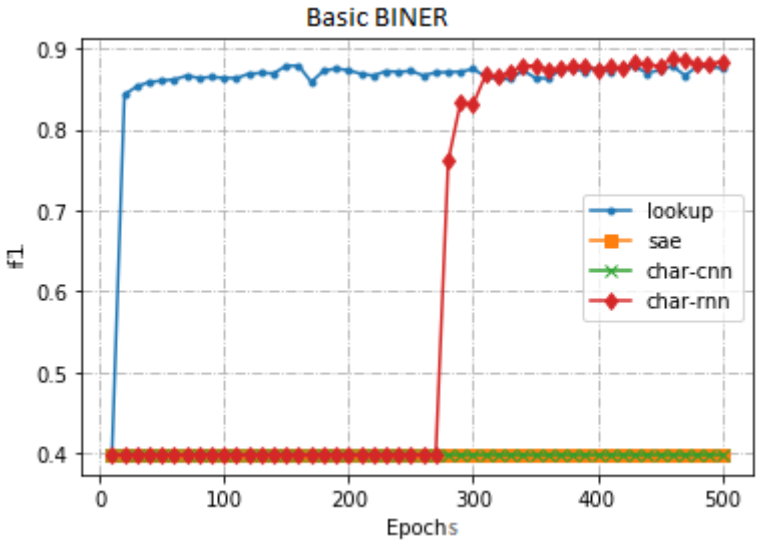

Figure 21 Epochs and Macro-f1-Score Basic BINER

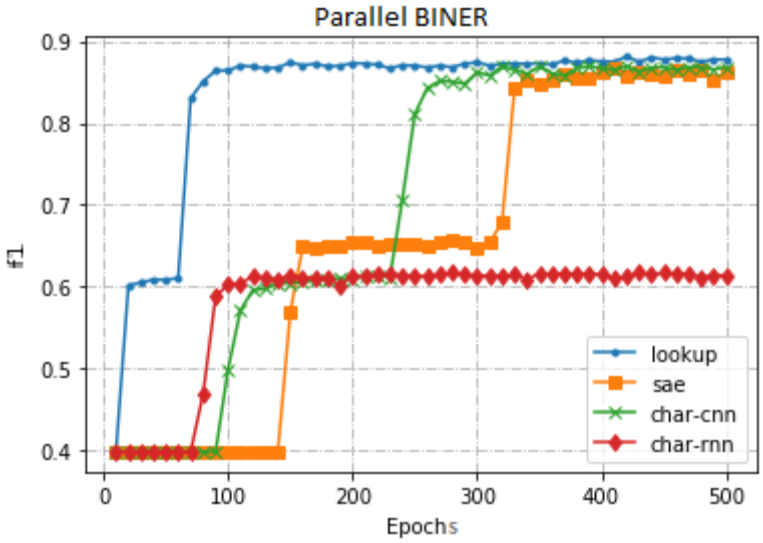

Figure 22 Epochs and Macro-f1-Score Parallel BINER

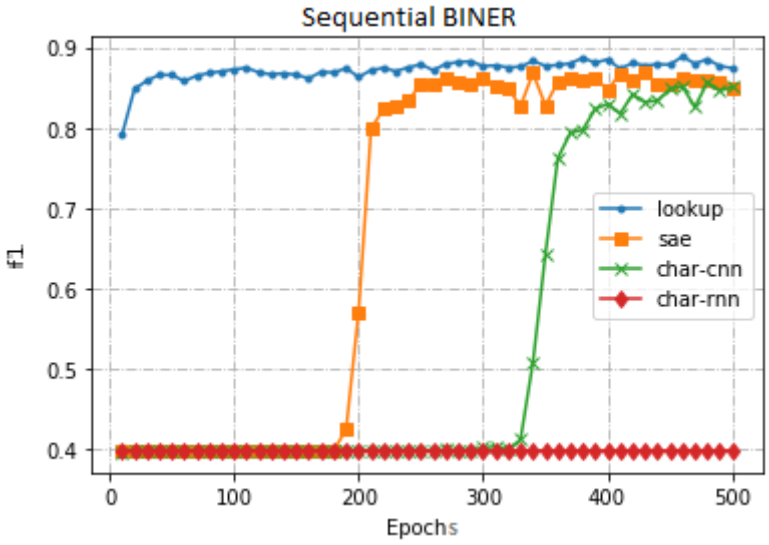

Figure 23 Epochs and Macro-f1-Score Sequential BINER

- **Embedding Size**

'Embedding Size' is the maximum length of a sentence in our model, explaining how much padding or cropping we might have to fit the sentences inside our model. Selecting this

parameter small will cause us to drop words from the sentences, and if we choose this number large, we need to add padding to some sentences to fit them. Thus, finding the best number for this hyper-parameter can play a significant role in the model performance. In Figure 24, we present the results from the three proposed implementations for NCBI based on F1-Score.

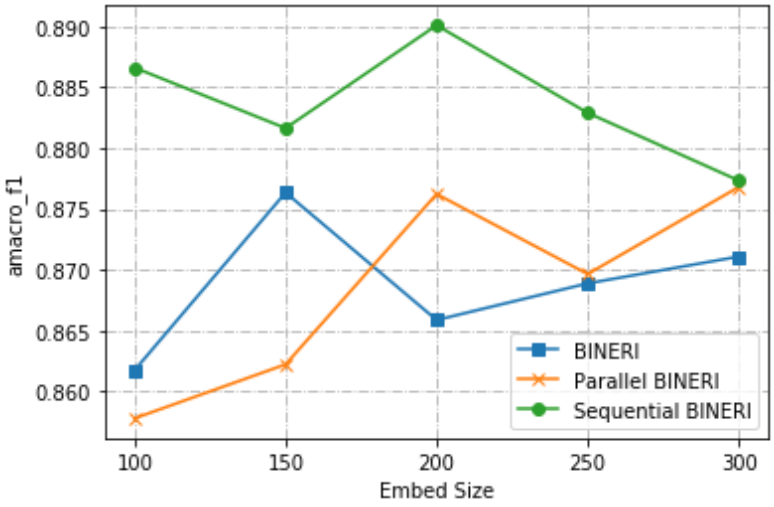

Figure 24 Relationship Between Embedding Size and f1-Score for Lookup Layer

- **Learning Rate**

This parameter uses for optimizing the weights in a neural network. It can change depending on the type of optimization function and architecture we use. For example, selecting a large number for the learning rate might pass the optimum point, makes the system unstable to converge. Choosing a small number might need more epochs and iterations for training. So, we need to select the correct number for our system to achieve weight optimization. We train our models for different learning rates and compare them using F1-Score. Figure 25 shows that the model will converge and have better results when choosing 0.0001 or 0.0002 as the learning rate for the NCBI-Disease dataset.

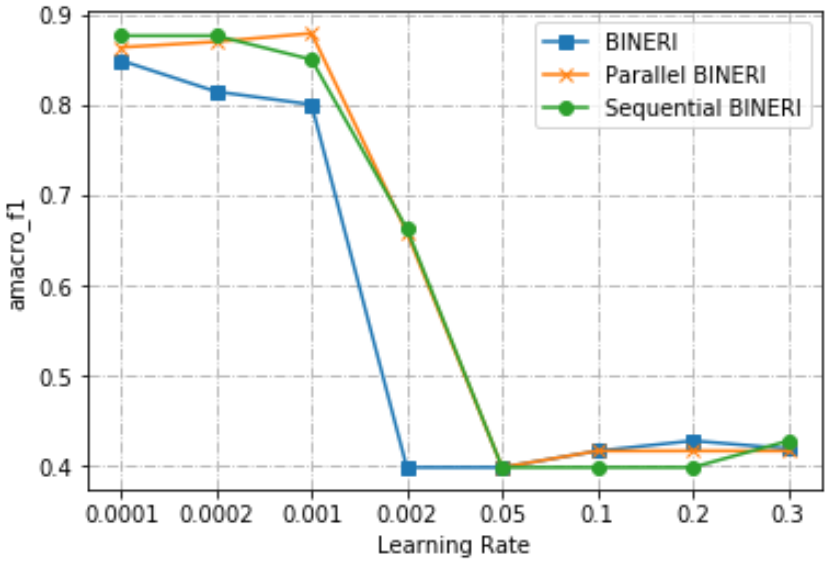

Figure 25 Illustrates the Relationship Between Learning Rate and F1-Score

- **Hidden Layers Size**

The hidden layer in a neural model defines the level of complexity. To choose the best number of nodes, we need to train a model to study their effect on the model performance. If we select this number too small will cause a limitation for learning all the patterns inducing information bottlenecks. However, choosing this number too large will increase the model's complexity and slow the convergence time, leading to unnecessary iterations, increasing the risk of falling into a local minimum. Figure 26 shows F1-Score over the different number of hidden layer nodes. This figure shows that 500 is the best choice as Hidden Size for Sequential BINER and 400 for Basic and parallel BINER.

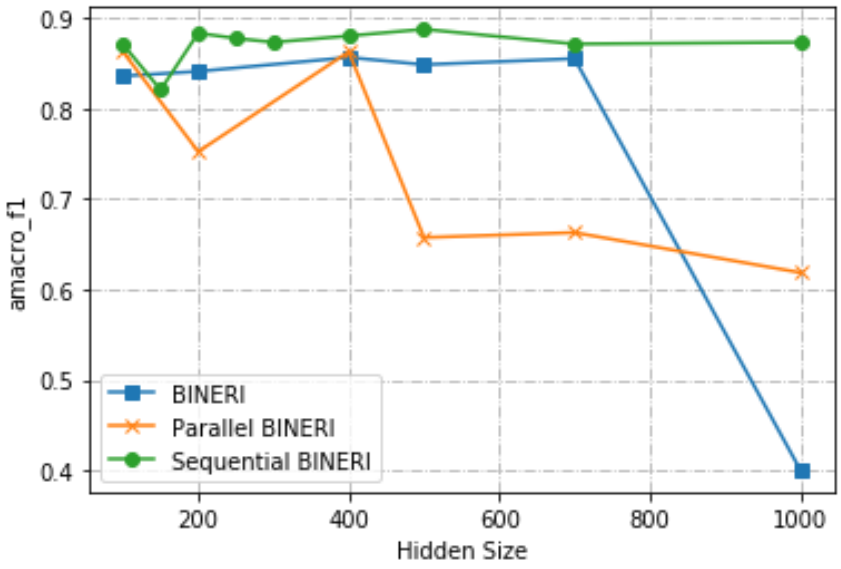

Figure 26 Illustrates the Relationship Between Hidden Size and F1-Score for Lookup Layer

- **Batch Size**

Batch size determines how many samples in each epoch we will feed to the neural network. Choosing the best number for this hyper-parameter may affect the model's performance in two aspects. First, the model's running time, selecting this number than large as possible, will limit memory limitations. However, the model will end up training faster. Second, choosing a more significant number will send more data to the model, and each epoch will do more calculations, which might decrease the generality of the learning. Therefore, choosing the correct number of batch size plays a significant role in the network. We select different batch size ranges and trained the model based on them to see the effect on F1-Score, Figure 27 Relationship between Batch Size and F1-Score for lookup layer on NCBI-Disease database.

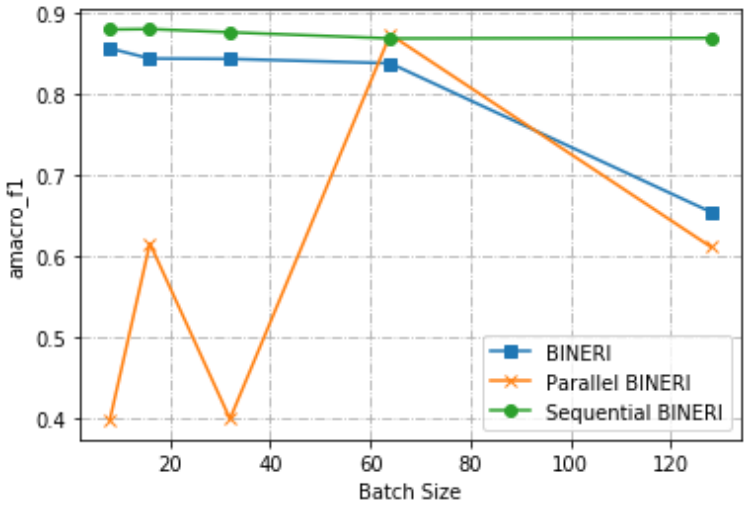

Figure 27 Illustrates the relationship between Batch Size and f1-Score for Lookup Layer on NCBI-Disease Database.

We obtain similar hyper-parameters when using the different databases. For this reason, we only present the results evaluated for the NBCI-Disease database. Finally, Table 5 shows the selected hyper-parameters to train every three implementations of the BINER approach.

Table 5 Shows the Hyper-parameters list for each Implementation.

	Models		
Hyper-parameters	**BINER sequential**	**BINER parallel**	**BINER basic**
Epochs	60	60	100
Learning Rate	0.0002	0.001	0.0001
Batch Size	16	64	8
Hidden Layer Size	500	400	400
Drop Out	0.5	0.2	0.4
Embed Size	200	300	150
Optimization Function	Adam	Adam	Adam

5.4.1.2 Results

In this section, we report our experimental results for our named entity recognition model over publication. We compare each of the different implementations with the two state-of-

the-art techniques BioBERT and BLSTM-CNN-CRF proposed [43, 46]. F1-Score use as our metric, which explains in Evaluation Metrics.

We report the F1-Score (F1), Precision (P), and Recall (R) for the Test datasets. In Table 5, we can see a comparison between the architectures with alternative Embedding Layers. Figure 30 and Figure 31 depicts the F1-Score of all the Embedding layer based on each database for the three types of proposed BINER models. We conclude from the represented results that Lookup shows more efficient and stable performance for word-level embedding compared to self-attentive encoder (SAE). Embedding convolutional neural networks (CNN) gives more consistent results across all three implementations on the character level. However, we propose using RNN character level embedding with the parallel BINER implementation as the best model. We compare this model with other experiments in Figure 32.

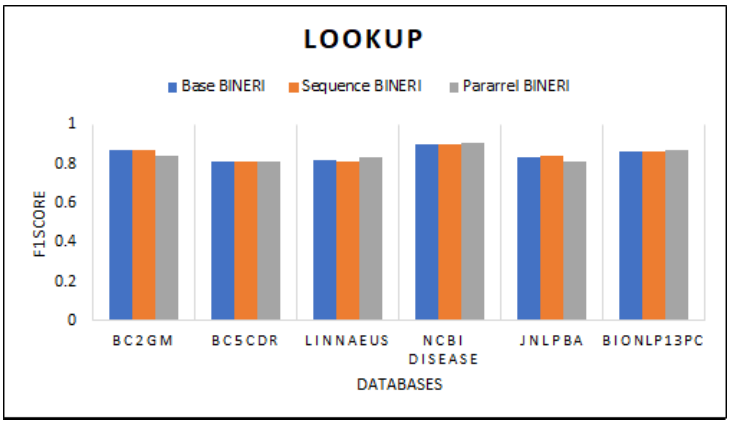

Figure 28 Reported F1-Score, Using Lookup Table as Word Embedding.

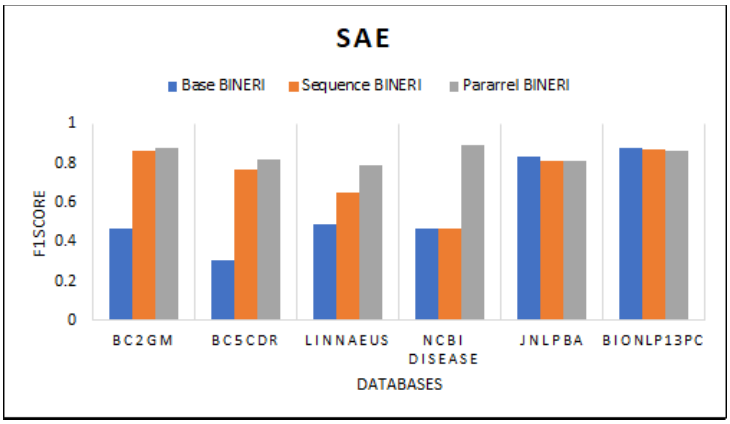

Figure 29 Reported F1-Score, Using Self-Attentive Encoder as Word Embedding.

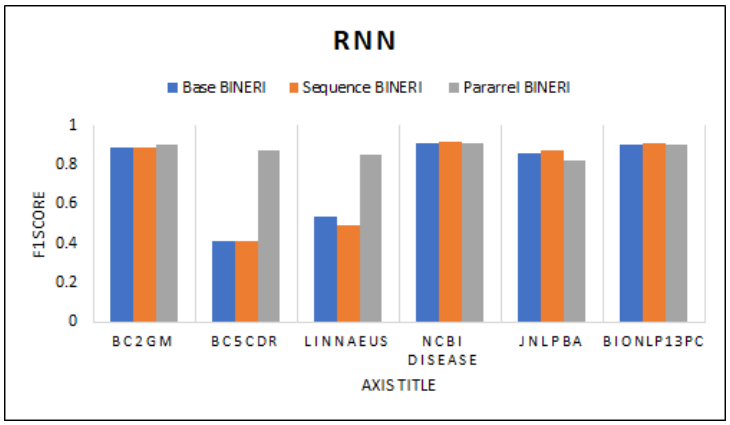

Figure 30 Reported F1-Score, Using Character Level With Recurrent Layer as Word Embedding.

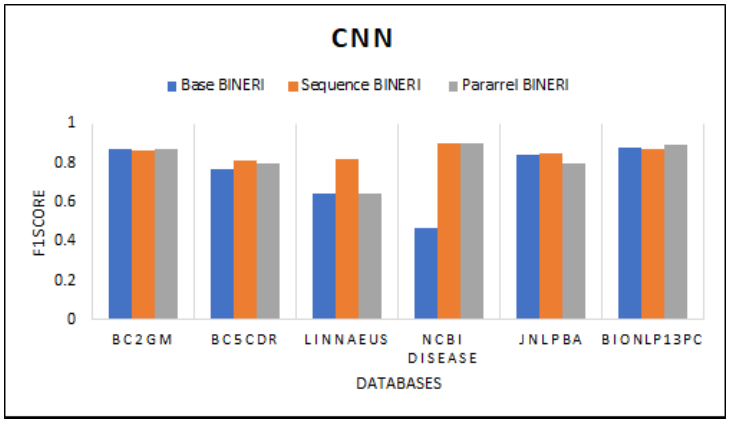

Figure 31 Reported F1-Score, Using Character Level With Convolutional Layer as Word Embedding.

Our recommended model is parallel BINER with RNN character level embedding. MTM-CW did not report results on LINNAEUS and BIONLP13PC. Also, BioBERT did not

report results on the BIONLP13PC database. F shows that parallel BINER performance is better or equivalent in almost all the cases unless using the Linnaeus dataset, where the BERT model performed better. The Linnaeus database specifically focused on species, and it is a relatively more minor dataset than the other five datasets. When we apply our approach, we achieved a 0.87 F1-Score which is less than BioBERT.

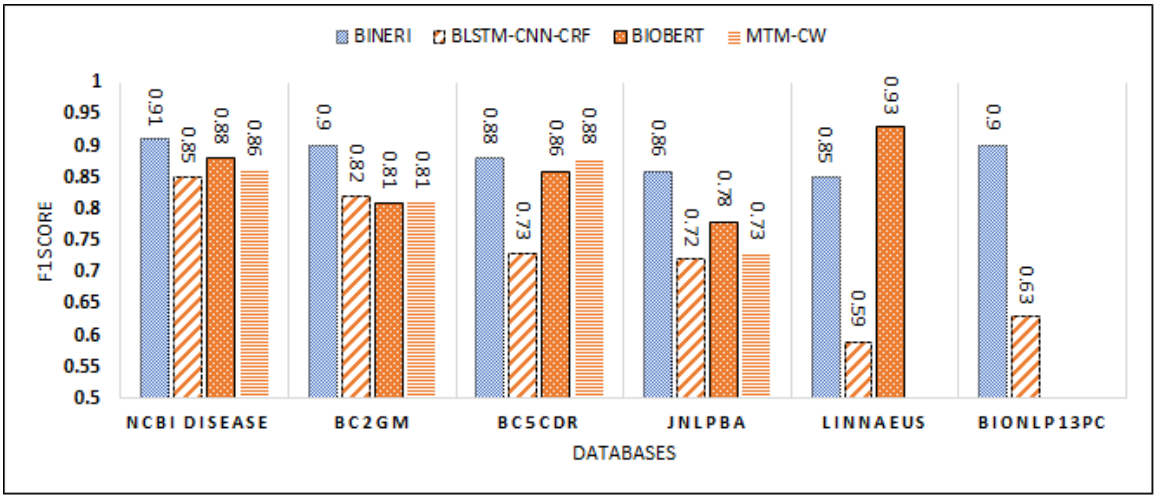

Figure 32 Comparison of BINER model with BIOBERT, BLSTM-CNN-CRF, and MTM-CW.

Table 6 Bolds and Underlines the Best Results in: The Precision, Recall and F1-Score Related to Each Architecture With Different Embedding Layers Represented.

Models		**BASE**				**Sequence**				**Parallel**				**BLSTM-CNN-CRF**	**BIOBERT**
Data Bases		RNN	CNN	Lookup	SAE	RNN	CNN	Lookup	SAE	RNN	CNN	Lookup	SAE		-
BC2GM	P	0.92	0.91	0.91	0.4	0.92	0.9	0.9	0.8	0.92	0.91	0.86	0.9	0.87	0.81
	R	0.89	0.86	0.86	0.5	0.87	0.84	0.87	0.9	0.9	0.87	0.89	0.9	0.81	0.81
	F1	0.89	0.87	0.87	0.5	0.89	0.86	0.87	0.9	0.9	0.87	0.84	0.9	0.82	0.81
BC5CDR	P	0.53	0.85	0.9	0.6	0.55	0.87	0.89	0.8	0.91	0.87	0.89	0.9	0.79	0.89
	R	0.33	0.77	0.76	0.3	0.33	0.79	0.75	0.8	0.85	0.79	0.77	0.8	0.73	0.83
	F1	0.41	0.77	0.81	0.3	0.41	0.81	0.81	0.8	0.88	0.8	0.81	0.8	0.73	0.86
Linnaeus	P	0.9	0.97	0.9	0.5	0.74	0.92	0.85	1	0.92	0.97	0.93	0.9	0.9	0.92
	R	0.52	0.58	0.76	0.5	0.5	0.74	0.78	0.6	0.79	0.58	0.76	0.7	0.51	0.94
	F1	0.54	0.64	0.82	0.5	0.49	0.82	0.81	0.7	0.85	0.64	0.83	0.8	0.59	0.93
NCBI DISEASE	P	0.94	0.45	0.93	0.5	0.94	0.94	0.94	0.5	0.95	0.94	0.94	0.9	0.9	0.88
	R	0.91	0.5	0.9	0.5	0.91	0.89	0.88	0.5	0.91	0.9	0.89	0.9	0.85	0.89
	F1	0.91	0.47	0.9	0.5	0.92	0.9	0.9	0.5	0.91	0.9	0.91	0.9	0.85	0.88
JNLPBA	P	0.86	0.84	0.85	0.8	0.86	0.84	0.85	0.8	0.86	0.85	0.83	0.9	0.72	0.74
	R	0.87	0.85	0.83	0.9	0.88	0.86	0.84	0.9	0.87	0.86	0.84	0.8	0.75	0.83
	F1	0.86	0.84	0.83	0.8	0.87	0.85	0.84	0.8	0.86	0.85	0.83	0.8	0.72	0.78
BioNLP13PC	P	0.92	0.9	0.86	0.9	0.93	0.9	0.9	0.9	0.92	0.91	0.89	0.9	0.71	-
	R	0.9	0.88	0.89	0.9	0.9	0.86	0.85	0.9	0.89	0.88	0.87	0.8	0.59	-
	F1	0.9	0.88	0.86	0.9	0.91	0.87	0.86	0.9	0.9	0.89	0.87	0.9	0.63	-

5.4.2 Clinical Note

Many pre-trained models are creating by large datasets and powerful machines, such as ELMO, BERT, GPT3, which few companies or research centers can build, maintain, and deploy. On top of accessing enormous resources, the cost of data labeling and keeping the models updated to create a bottleneck for expanding the models in the real world. In this case study, we show the effect of shifting data in the same domain, such as training data for detecting drug names on publication but using the same model to detect the drug names in clinical notes. We also study the effect of the data labeling size for training on deep learning architectures.

We design our experiment by bringing unseen data. A set of documents that are clinical notes, and physicians annotate them for drug adversaries. Thus, we focused on chemical and drug entities. The Adverse Drug Events (ADEs) [86] dataset proposed creating a set of medical documents that physicians annotate the drugs-related information, including drug names, dosage, strength, duration, frequency, form, and reason. This database detects the medicine and creates two types of relation: drug with specific symptoms and disease and drug with adverse events.

We select one state-of-the-arts technique named BERT, this model trains on PubMed and Mimic dataset [88]. To compare with our proposed BINER model in the first step, we use both models without tuning or retrain to detect the drug names. The result shows that both model needs retrain and tuning again to perform on a selected dataset.

Retraining the model might be sound easy, however, preparing and annotating new data is a time-consuming and expensive process. We design an experiment to study the training size effect in transfer learning. Figure 33 shows how we split the database into separate

sections for Train and Test. We randomly select 20% for the test and 80% for the train in the first split. In the second split, we create ten small datasets from the 80% training set. We randomly select 10% of the train and incrementally add it to the dataset. The first dataset contains 10%, the second dataset 20% until the 10th dataset, representing the 80% training set.

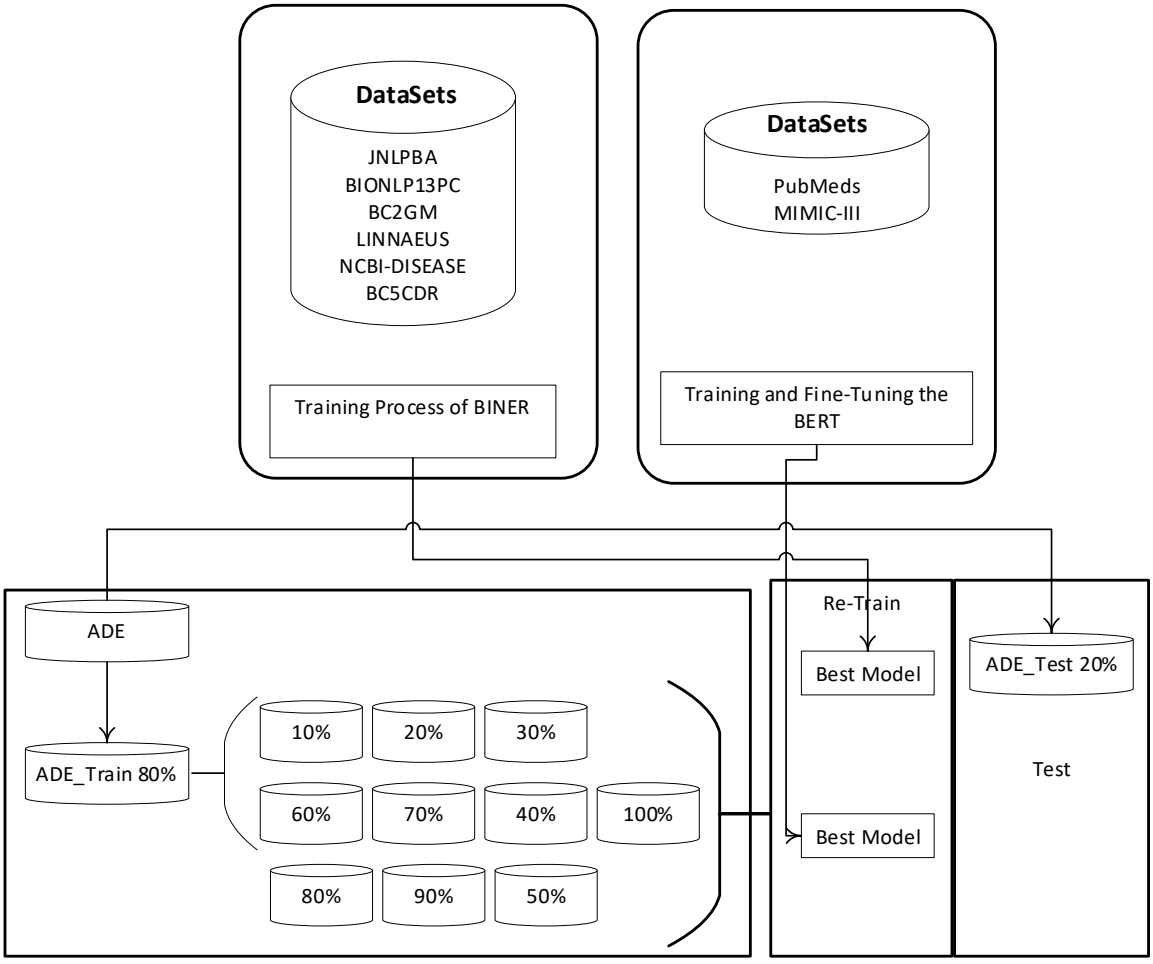

Figure 33 Shows the Process of Retraining the Models for the ADE Database

In this experiment, we consider two factors first, the size of the training set, which we have ten datasets, second is the number of epochs we did 10,20, and 30 Epochs. Bert model as state-of-arts selected to compare with the models. Compare the BINER Parallel with Bert;

the results show that Bert performs better. On the other hand, BINER Sequence shows better accuracy compare with BERT. The results are represented in Figure 34 and Figure 35.

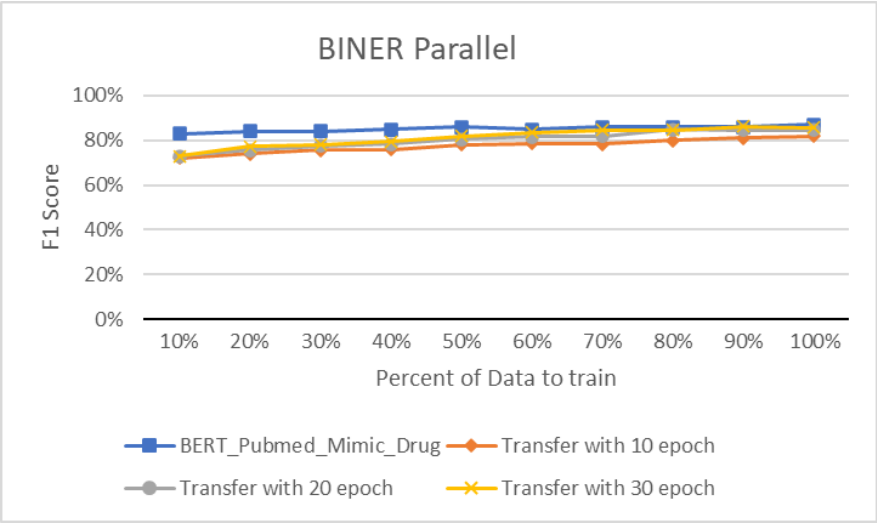

Figure 34 Compare BINER Parallel With BERT

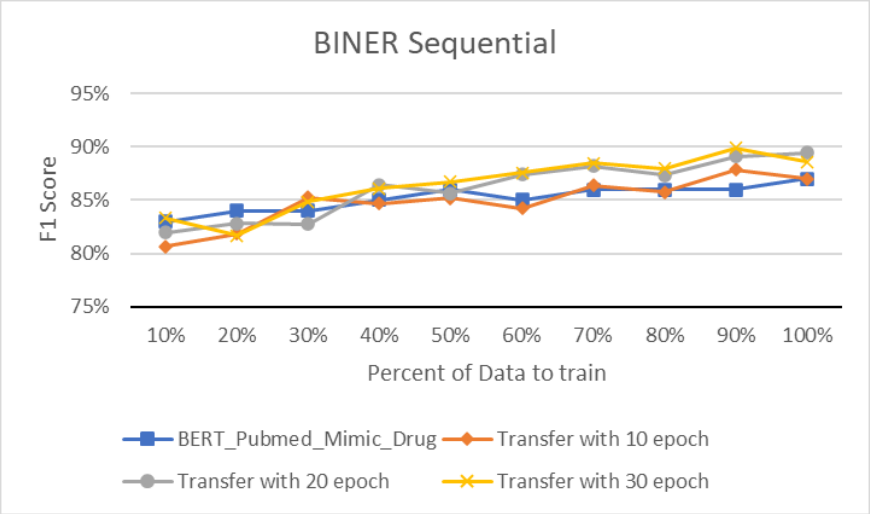

Figure 35 Compare BINER Sequence With BERT

CHAPTER 6

UNIFIED MODEL FOR LINKING SOCIAL MEDIA WITH BIOMEDICAL TEXT

6.1 Introduction

We capture trends on HealthCare and NLP using our framework defined 4. We collect 1,059 unique papers using “NLP” and “health” keywords. In the publications from 1992, a significant barrier was managing handwritten notes in emergency physician department and the research proposes digital dictation is the gate to electronic record [87], up to July 2021 which we found 136 publications. The highest number of publications was in 2019, with 212 publications. Figure 36 shows the movement of publications during the years. This chart represents there is a huge break out on this area of research as in 2013. There were just six publications in 2012, and in 2013 we see 58 publications. Shows 89% growth on this topic, and growth average is 20% each year up to the end of 2020.

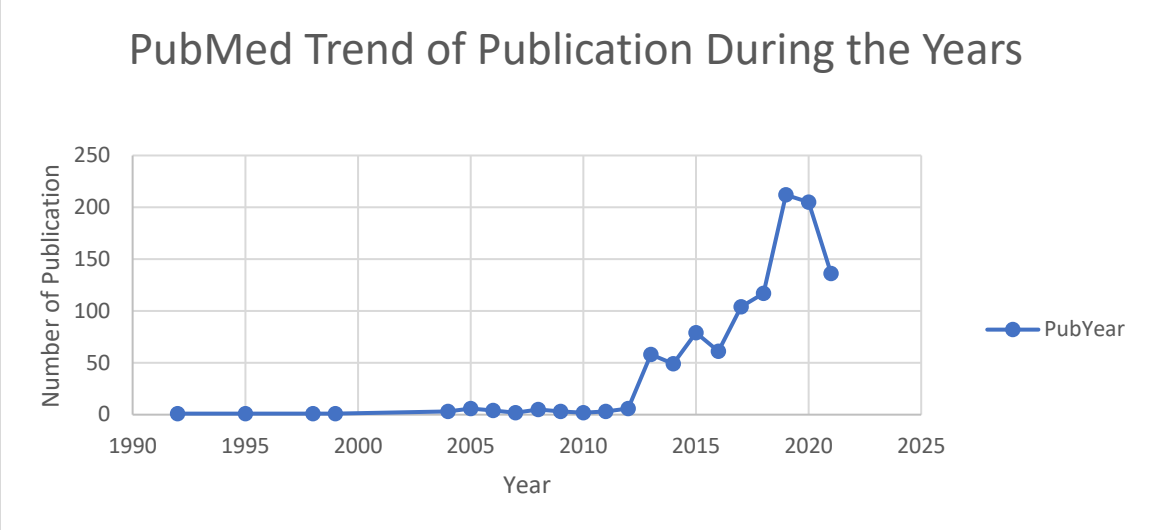

Figure 36 NLP Publication Trends from 1992 to July 2021

In 2012-2013, researchers begin to leverage data-mining techniques for un-structural data, specifically healthcare text. Figure 37 shows Bayes Theorem, Super vector machine, and using ROC curve was the most critical word using in the publications. From 2018 to 2019, Deep learning introduces Transformers, Elmo [54] (2017), Bert [43, 67, 84] (2018), GPT3 [88] (2020) models to the NLP community, and healthcare also affected by that, and researchers focused on study and application of techniques, such drug overdose, survey, and questionnaire. Interestingly, analysis on keywords in 2020 papers shows the shift of the researchers and focuses on Covid-19, Pandemics, Coronavirus infections, beta coronavirus [89-97]. The Word Cloud Analysis shown in Figure 37 during 2014 EHR systems becomes available to be part of the research. The word "Machine Learning" did not appear firmly up to 2019 and become bolder in 2020; this can explain the jump in the number of publications from 2018 to 2019. Database annotation for medical records, such as annotate the name of the disease, genes, drugs, or DNA, happened in 2013. Later in

2018, many conferences created some TASK for annotation and did a lot of research on different databases.

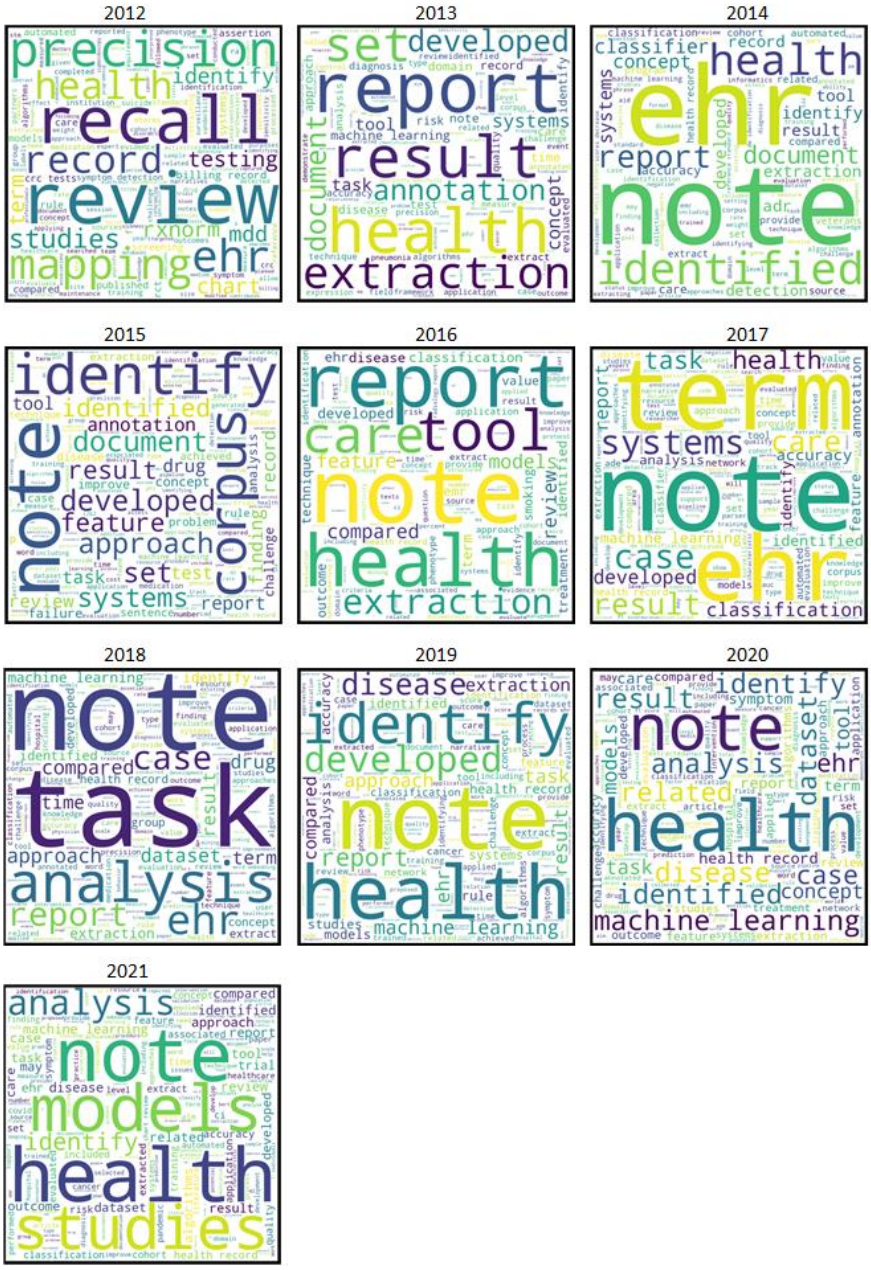

Figure 37 Word Cloud Analysis Over 2012 to 2021 PubMed Papers Collected by Health and NLP Keywords

When analyzing abstracts of over 1,059 papers, we did not find specific publications referring to unifying different data sources in the medical domain. We believe that knowledge is spread throughout multiple sources, such as social media, biomedical

journals, and EHR systems. We can see a lack of research on how we can link these data sources together.

The most significant effort is introducing the Unified Medical Language system (UMLs) as a knowledge source that brings completeness, consistency, and usability in the medical domain. A complete knowledge should have good coverage, meaning we should capture the basic concept and the connection between concepts. Consistency directly relates to the concept's connectivity and creates a logical grouping in concepts. Finally, is usability, which mainly talking about centralizing a database of concepts to make the knowledge more accessible.

However, linking different data sources and creating a unified knowledge in healthcare still has gaps and requires more research and study. With more study, we can high quality medical knowledge more accessible to experts and the general public. To fill some of the gap, we create a unified model to link social media, literature, and clinical notes. This section will discuss the framework and represent a case study and explain how we implement each part in a web application.

6.2 Unified deconstructed model

There are two primary layers in the framework, ETL is the first layer helps us create unified model after extracting and processing the data. Figure 38 represents all the framework. Before we start to do Extract Transfer and Load (ETL), we define our three data sources; the clinical notes data-source drawn by dotted border because it is private and needs authentication and privileges to access. In the Extract Transfer Load (ETL) layer, we define two processes for collecting data one private, and the other is a public collector. Private

Collector uses for Clinical Notes Data-Source, also drawn by a dotted line because extracting data from Electronic Health Records (EHRs) systems such as Cerner or Epic. This process, because of the different data architects, needs a separate explanation. We store clinical notes largely as a Binary Large Object (Blob) and compress them by applying OCF method used to compress tiff images. The public data collector uses data from social media and publications websites. For example, we support Twitter, Reddit, New York times, and YouTube to collect data from social media and NCBI PubMed for publication. The pre-processing section contains all the steps from tokenization to general cleaning and deep cleaning, all introduced in Data Cleaning Module; we provide more information in our publication [21]. After data extraction and transform them in the correct format, we send the data to Unified Deconstructed Model to store them; we load them in separate data sources because the models and usage of these data are different. Finally, we send the social media data source to the Semi-Supervised Topic modeling engine because social media contains a lot of noise. Therefore, we used noise cancelation defined in section 4.4 after that social media topic extract by the method we describe in section 4.5. As we show in our case study for health trend detection, we observe the topics changes. This dissertation defines a general drift Indicator, and a method for distance-based detection for the drift. We focus more on covariant shift and define this module in section 6.2.1 below. We process the biomedical publication and clinical notes with BINER Named entity recognition system to detect the entities in parallel to process social media. In the end, we define a structure to store and make the extracted knowledge accessible and searchable in entity/document storage.

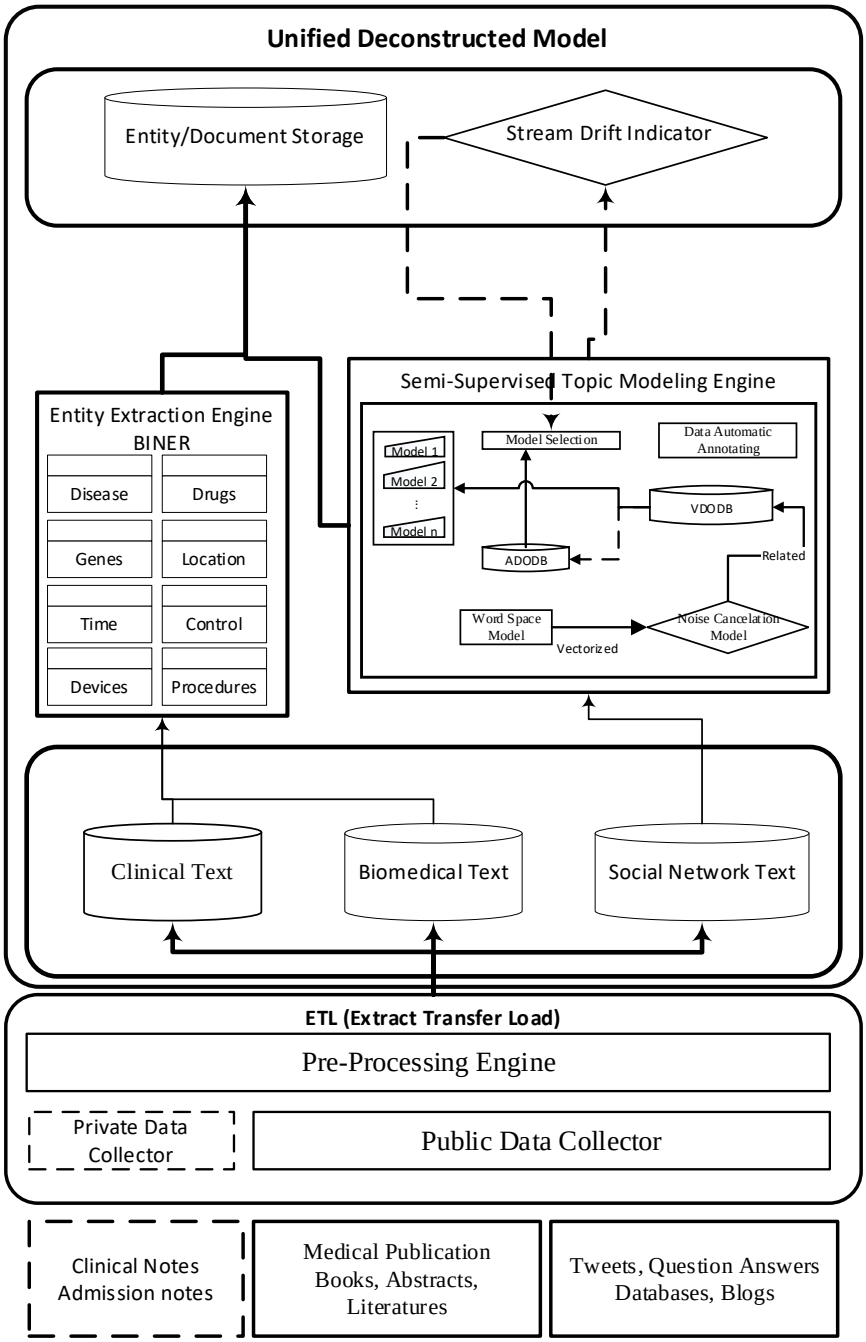

Figure 38 Unified Deconstructed Module

6.2.1 Stream Drift Indicator

Our focus is on *covariate shift*; this concept can happen in an abrupt or gradual time frame; here, we define this as a window for a time set. If we assume, the time set is defined as

$t_i = \{t_0, t_1, \ldots, t_n\}$. Each time frame t_i includes a set of records known as $V_j = \{v_{j_0}, v_{j_1}, \ldots, v_{j_m}\}$ where m is the number of records in the set V at a time j that is in the range of $0 \leq j \leq n$. A metric is needed to evaluate the situation at each time frame to explain a shift to be captured. In this procedure, we defined several steps to do the analysis. In the first step, we need to calculate a set of centroids for each V_j. A distance function d must be defined to calculate the centroid. In this study, we use Euclidean distance. However, this function can be determined based on the data type; for example, if we are processing text, it could be better to use cosine-distance. C is defined as a set of calculated centroids.

$$C = \{c_0, c_1, c_2, \ldots, c_{n-1}\} \tag{24}$$

This set will be calculated and updated for each V_j which represents new data; therefore, we always need to update the current centroid based on the previous dataset, in other words, if we take n as the time, then we need to calculate c_{n-1} to calculate C_n, to start this recursive iteration, we must calculate c_0 as:

$$c_0 = \frac{\text{argmax}\left[d\left(v_{0i}, v_{0j}\right)\right]_{0 \leq i,j \leq m}}{2} + \alpha \tag{25}$$

By using Equation (25), we calculate the distance between all possible pairs at time zero in v_0. Where $argmax$ will return the position of the pair of extreme points. To represent the centroid location, we calculate the geometrical center in Equation (26) by dividing the max distance between two points by two; however, there will not necessarily be any item to be considered the center. In this scenario, we define α as the radius around the center. This parameter can be determined by a user as a threshold to increase the area of the center

until we have at least one item as the center. This method of finding the center is sensitive to outliers; however, having outliers in the train set V_0 could mislead the algorithm. In other sets that arrive later, defined as V_j, this could be known as a drift. Our solution to this problem is to study the small set of "first-generation" data to handle these outliers before calculating c_0. After calculating c_0 we update the centroids in each period of time; we define the recursive procedure:

$$sc_k = \left\{ \frac{argmax\left[d\left(v_{k-1,i}\,, v_{k-1,j}\right)_{0\le i\,,j\le m}\right]}{2} + \alpha \right\}_{1\le k\le n} \tag{26}$$

$$[\, d(v_i, v_j) : \; v_i\,, v_j \in \; c_{j-1} \cup \; v_j]$$

In the next step, we need to estimate the PDF of the distance between each point in V_j and the centroid of v_0. Kernel Density Estimation (KDE) methodology is used to estimate the PDF. Where $\{x_1\,, x_2\,, \ldots x_m\}$ and x_i represents the distance from the last data distribution centroid C_{k-1}. We know that this distribution is drawn from an unknown density function f. The estimation function is calculating the density estimation kernel K and bandwidth h. We can then define $\hat{f}$ as the distribution KDE calculated as follows:

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K((x - x_i)/h) \tag{27}$$

If we define $\hat{\sigma}$ As the standard deviation and IQR as the interquartile range, the optimal choice for h is found using a Gaussian function as the kernel. The most accurate calculation is as follows:

$$h \;=\; 0.9 \;*\; min\,(\hat{\sigma}\;, IQR\;/\;1.34) \;*\; n^{1/5} \tag{28}$$

The range of *covariate shift* is known as β or concept drift indicator (CDIC), representing the difference between the density function of distances before and after drift. Each time a new set of data arrives in a stream, we assumed drift did not happen and took it as the null hypothesis. We use the Two-Sample K-S test to prove our hypothesis. The K-S test is sensitive to any change in distribution, and this sensitivity will enable the methodology to detect any change caused by a concept drift. The range of β will be normalized between [0,1]. To reject the null hypothesis and be confident that the data came from the same distribution, we need to have β close to zero; otherwise, we cannot conclude there is any drift. β or CDIC will be calculated as follows, where SUP is the supremum function:

$$\beta = SUP_i \, |\hat{f}(V_{i-1}) - \hat{f}(V_i)| \tag{29}$$

6.2.2 Entity and Knowledge Base

The Unified Medical Language System (http://umlsks.nlm.nih.gov) is a repository of biomedical vocabularies developed by the US National Library of Medicine. The UMLS integrates over 2 million names for some 900,000 concepts from over 60 families of biomedical vocabularies and 12 million relations among these concepts. Vocabularies combined in the UMLS Metathesaurus include the NCBI taxonomy, Gene Ontology, the Medical Subject Headings (MeSH), OMIM, and the Digital Anatomist Symbolic Knowledge Base. UMLS concepts are not only inter-related but may also be linked to external resources such as GenBank. In this dissertation, we defined our knowledge base as compatible with UMLS to connect our knowledge with powerful tools, such as UMLS.

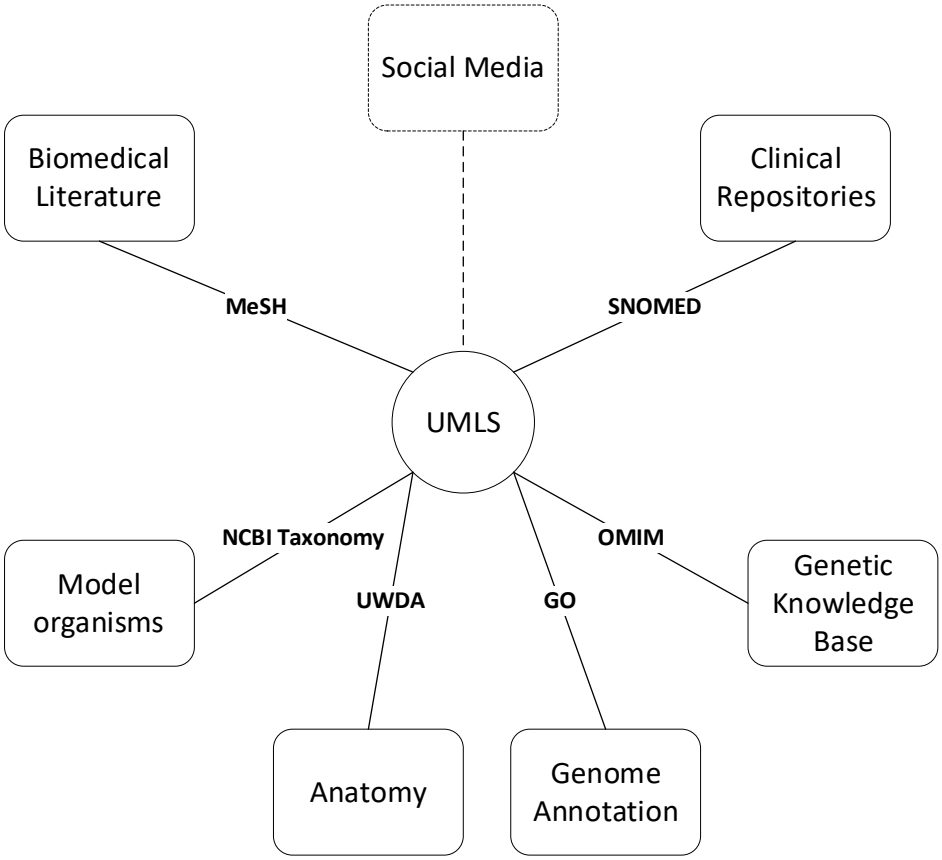

Figure 39 UMLS Subdomains, We Introduce the Social Media Domain and Add to the UMLS

UMLS does not explicitly design for bioinformatic needs; it contains terminologies that are used for bioinformatics. For example, recently integrated languages include the NCBI taxonomy, used for identifying organisms, and Gene Ontology, used to annotate gene products across various model organisms. The sub-domain described best is probably the clinical component of biomedicine, with general terminologies such as SNOMED International, Clinical Terms Version 3, and the International Classification of Diseases (ICD) name a few. Clinical genetics resources include the Online Mendelian Inheritance in Man (OMIM). Other categories of terminologies in the Metathesaurus include specialized disciplines (e.g., nursing, psychiatry) and components of the clinical information system (e.g., diseases, drugs, procedures, adverse effects). Figure 39 illustrates

how the UMLS Metathesaurus, integrating these various terminologies, can link the vocabularies and the sub-domains they represent. We introduce the social media sub-domain as an alternative source to help the healthcare community identify and access public health. This connection is bidirectional; this connection can be a gate for the public to access the publication and knowledge collected in UMLS. Our purpose is to connect Biomedical Literature (MeSH) to Clinical repositories (SNOMED) with social media; Figure 40 illustrates how each defined module in the framework links these three areas (Biomedical Literature, Clinical Note, and Social media) based on UMLS standards.

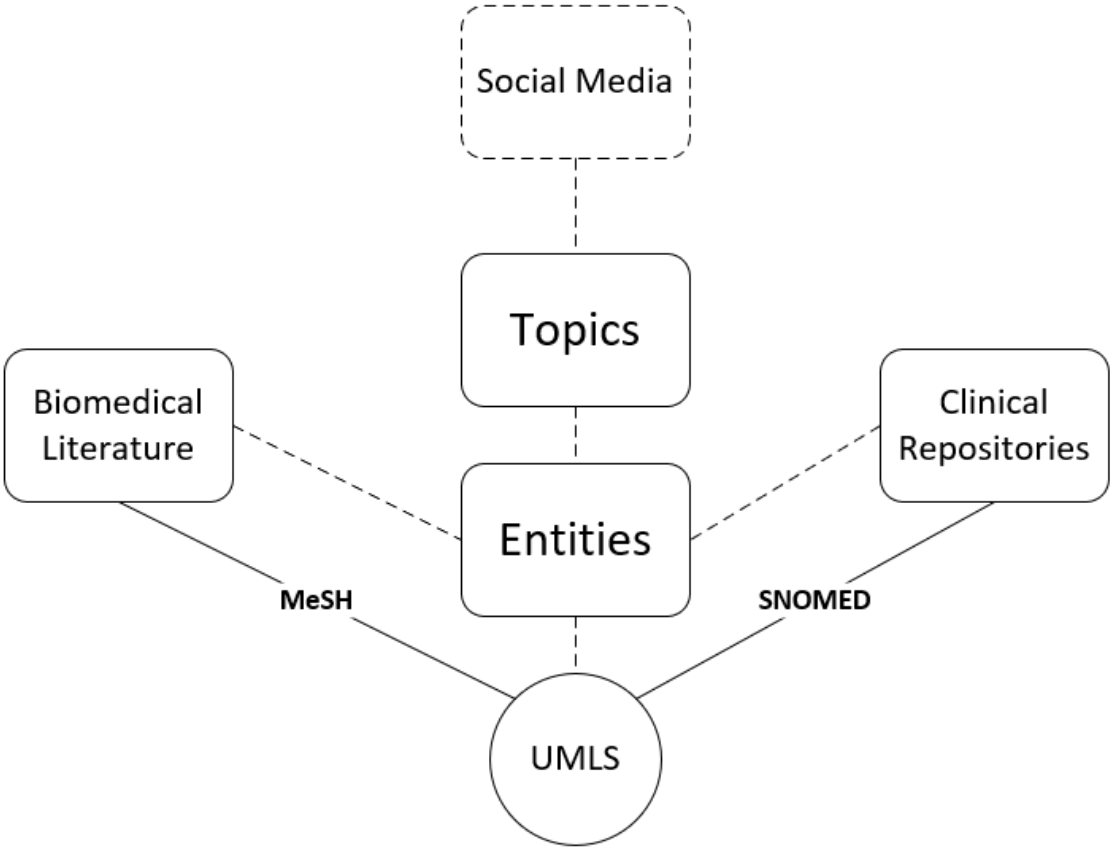

Figure 40 Illustrates the Connection Between Social Media Biomedical Literature and Clinical Repositories Based On UMLS

6.3 Web Implementation

Linking the knowledge and make the text available for physicians, and analyzing unstructured data is the primary purpose of our framework. We implement three types of applications.

The first one, we implement a web application to enhance physicians' ability to access the text by a search engine. Figure 41 shows how we create a search engine for search in the clinical notes using Electronic Health Records EHRs as the data source. This System defines six sections; a physician can input a text as a search term to search in the target source such as Radiology or Pathology. The search engine will perform a similarity method to rank the notes; we provide a score for each document.

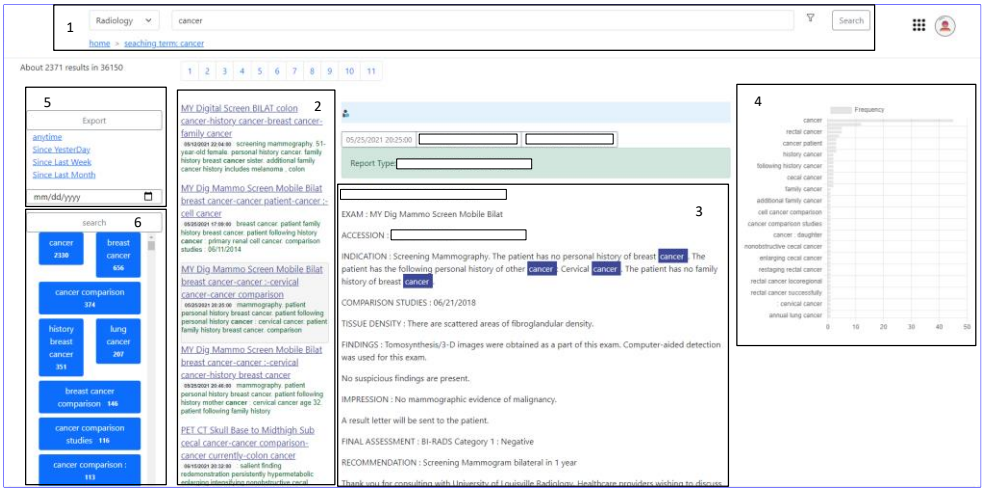

Figure 41 The clinical note search engine application

One capability of this search engine is that it lets you perform a query such as “Select Cancer from Radiology Where not Breast Cancer since last week in Facility.” Figure 42 shows the query details, which can filter based on Include, Domain of Search, Exclude, Time, and Location. In addition, the System will provide suggested tags based on search

results; for example, if we use our search term “Cancer,” the suggested procedure will create these terms “breast cancer,” “history breast cancer,” “lung cancer.” Each search procedure will return many documents, so we page the search; there is an analysis of the most frequent three-gram words generated per page, which means we can overview the documents on each page.

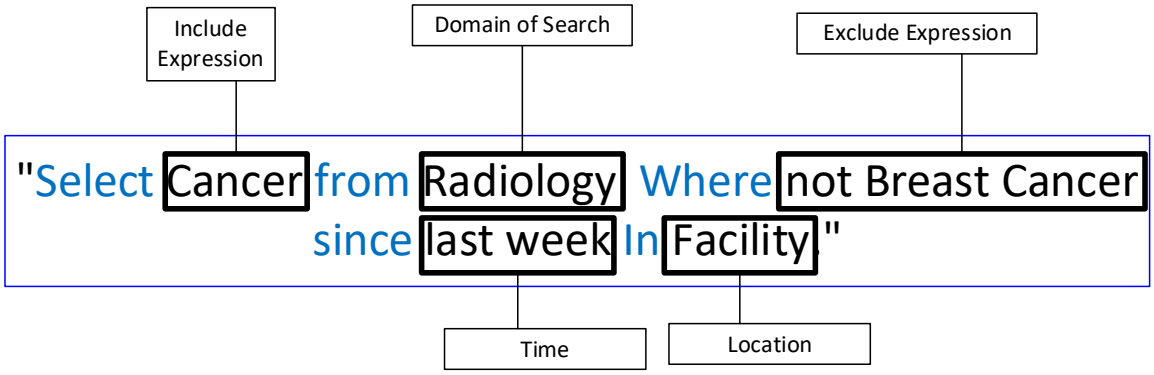

Figure 42 Sample Search Engine Query

National Library of Medicine develops a system called PubTator [98-100], a web-based system providing automated annotation of biomedical concepts such as disease, chemical, and genes. Part of our study focused on creating models for automatic annotation, which results in Bio Named Entity Recognition (BINER); in this application, we deployed our proposed BINER model in a Flask Project. The model trained on an NVIDIA GTX 2080; however, in deployment, we used CPU. For proof of concept, we deploy just a model to detect the chemicals in publication. We randomly select a paper about Breast Cancer and compare this example with Pubtator to annotate the chemical; simultaneously, we use BINER on the exact text. Figure 43 shows a comparison. On the left is BINER run the application on a local machine, and on the right, we have Pubtator representing the annotation for chemical. In this example, we can see the BINER addition to “exemestane” detect two more chemical, “estrogen” and “fulvestrant.”

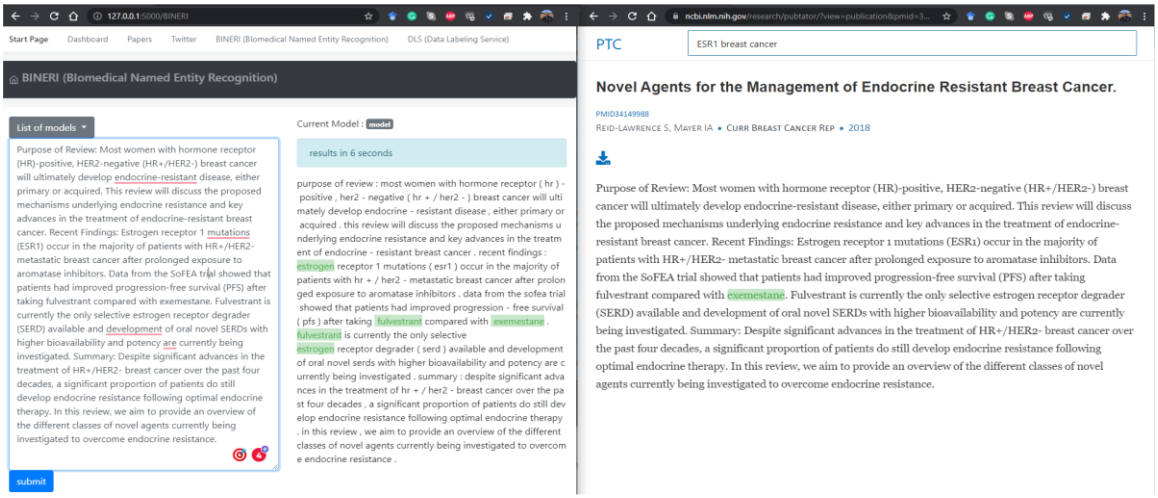

Figure 43 Presents the BINER algorithm on the left and Pubtator on the Right, Both Annotating the Exact Same Text

In classification metrics, there are several standard terms such as precision, recall, and F1-Score. Precision is a positive predictive value, and Recall is sensitivity. The-recision will explain "How many classified comments are related out of all the comments," and Recall/Sensitivity will describe "Out of all the Related Comments how many classified correctly." However, reporting all these numbers for comparison might confuse, so we used a metrics which is calculated by the average of Precision and Recall named F1-Score explained in Equations (32) and Precision and Recall also represented in Equation (30) and (31) respectively, if we define $TP = True\ Positive$, $TN = True\ Negative$, $FP = False\ Positive$, $FN = False\ Negative$.

$$P = \frac{TP}{TP + TF} \tag{30}$$

$$R = \frac{TP}{TP + FN} \tag{31}$$

$$F1 = H(P, R) = \frac{2 * P * R}{P + R} \tag{32}$$

6.4 Case Studies: A Recurrent Neural Networks For Kidney Donation Comments in Social Media

6.4.1 Overview

Living kidney donors currently comprise of approximately a quarter of all kidney transplants. Still, there are barriers that may prevent prospective donors from following through on donation, such as medical ineligibility, psychosocial pressures, and long-term health risks. We seek to further understand why people choose to be living kidney donors by using machine learning techniques to classify internet comments. We collect comments related to living kidney donation from the New York Times (NYT), Reddit, YouTube, and Twitter articles. To the best of our knowledge, this research is the first to create a public database of internet user comments that can analyze and understand the motivations and barriers surrounding living kidney donation (LKD).

6.4.2 Modeling

We design a Neural Network that detects Live Kidney donation comments in social media. Seeing as the neural network cannot process text, a component to transform the text to numbers is required. Through a process called embedding we can achieve this. Two examples of existing embedding processes for this transformation technique include Google Word2Vec [52] and Stanford Glove [10]. Although, other methods can be used to develop customize embedding, which is the route we find to be most effective in achieving our desired results. By feeding the vocabularies to the Pytorch embedding tool, we are able

to build this layer. GLOVE and Word2Vec are not agnostic to word misspelling; thus, word to vector will not face deficiency. We customize the embedding layer to improve our classification's power. We also experiment differently between word tokenization and character tokenization. Figure 44 illustrates how we create the embedding layer based on word and character tokenization.

We define our neural network architecture by two layers: in the first we represent a hidden layer (where transformations occur) and in the second, an output layer (which determines the final classification). The hidden layer is made of Recurrent Neural Network (RNN) nodes which are constructed with a Bidirectional Long Short-Term Memory (B-LSTM) cell [101].

In the end, we aggregate the hidden layer and generate the probability for the output layer. This means if the output layer generates a number between zero to 0.5 for a given comment, it will be classified as “Not Related” and if between 0.5 to 1, as “Related”. We used CrossEntropy to define the loss function for the training process [102] and tested learning rates for the Adam optimizer [103] function. The optimization process will change the hyperparameters to reduce the error. This process requires repetition where each iteration of the process is called an epoch.

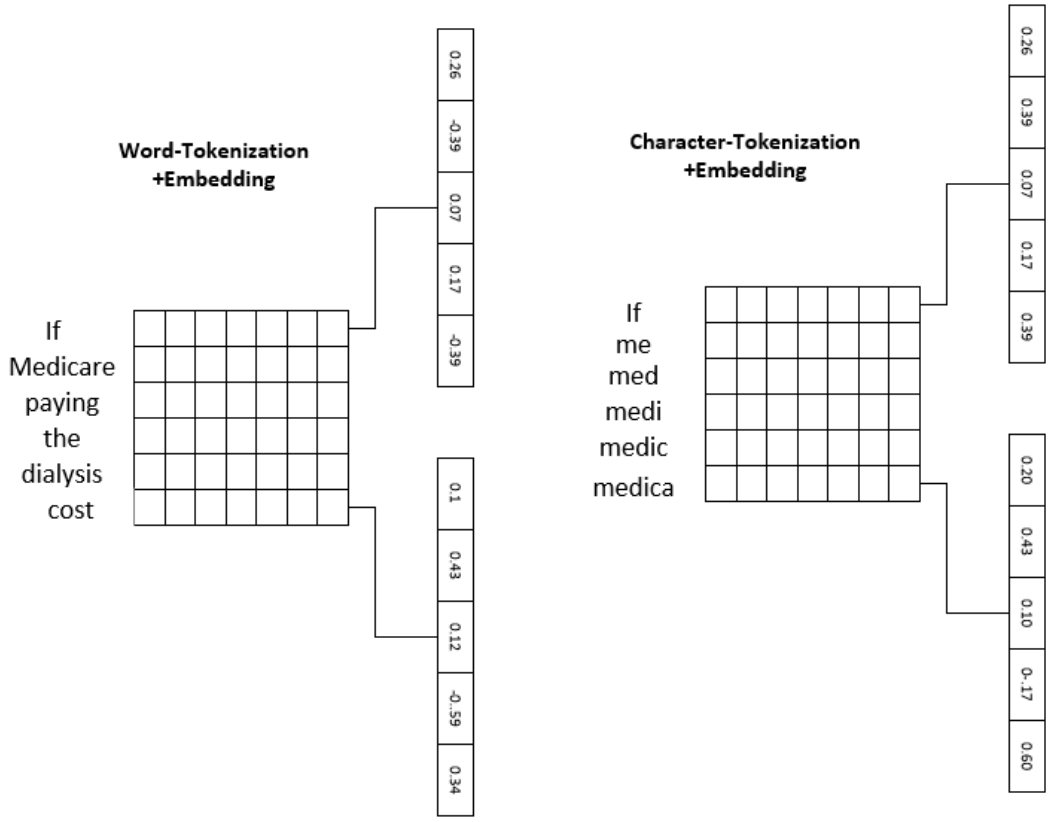

Figure 44 Embedding Layer With Word and Character Tokenization

We represent the Neural network architecture in Figure 45 above. After we develop the embedding, we need to create a layer that can read the words or the character-gram we made in the tokenization process. This let us know which set of words relate to the point of attention of comments relevant to the kidney provision. With keeping this in mind, we create a layer inspired by a transducer.

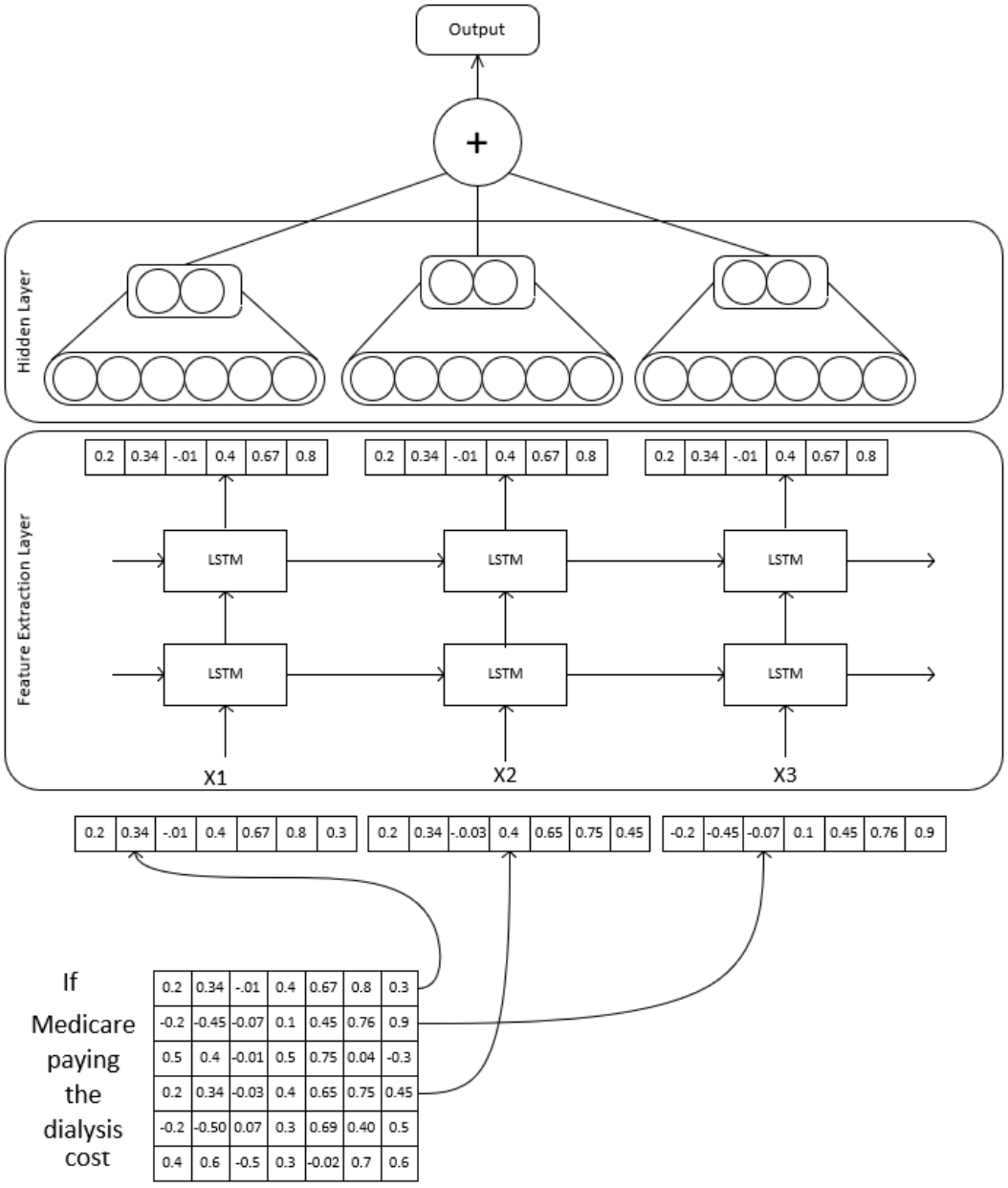

Figure 45 Classification Neural Network Architecture

A transducer produces an output $\widehat{t_i}$ for each input, and each $\widehat{t_i}$ compute loss function based on a true label. One of the everyday uses of this structure is sequence tagger [Xu et al., 2015]. This methodology uses language modeling as well [Jozefowicz et al., 2016]. Figure 46 shows the structure of this type of Recurrent Neural Network (RNN).

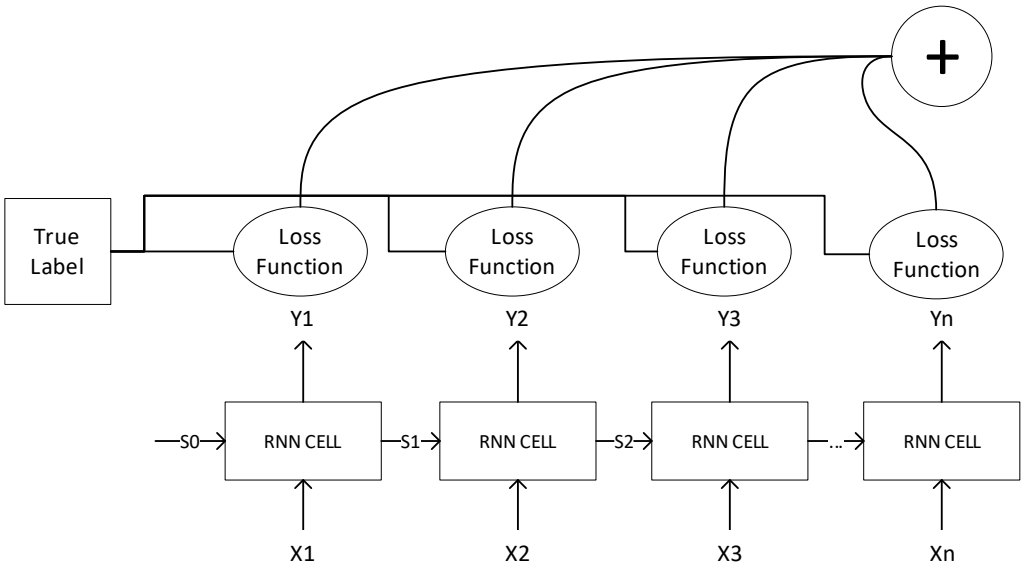

Figure 46 Transducer RNN Training Graph

The transducer inspires our model; however, we add an extra layer between loss function and Concat. We design customized Multi-Head attention [Attention Is All You Need]. We describe attention as a mapping between input and output where both are pairs of vectors. The output of our system is binary, so we customized the model to single head attention. We construct the original Multi-head attention from linear layers scaled dot-product attention layer, and Concat produces a linear layer. Figure 47 represents the scale dot-product attention and multi-head attention.

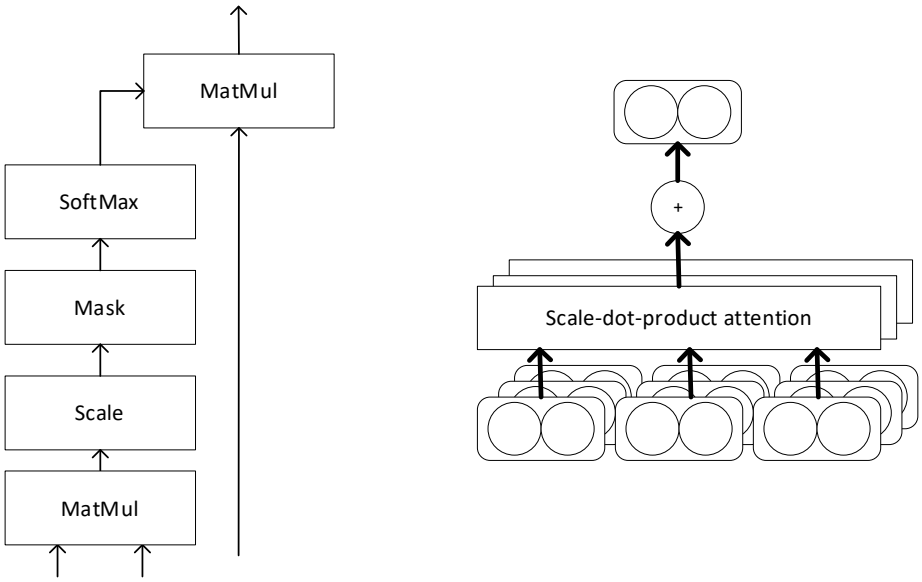

Figure 47 Shows the Scaled Dot Product at Left and Multi Headed Attention at Right

We implement our attention on top of the customized Transducer.

We define and create our dataset through gathering, filtering, and cleaning data before using the data or placing it in any storage. Collecting data from different sources, including comments on newspaper articles published in New York Times and comments on the social media sites Twitter, YouTube, and Reddit. Collecting 224,752 messages from December 2020 to May 2021. The breakdown of comments is: 162,559 from Twitter, 6,809 from the New York Times, 52,234 from Reddit, and 3,150 from YouTube.

Since annotating comments is time-consuming and expensive, we manually annotated/classified a small percentage of the data (1,210 out of 224,752) and design a neural network to harness the power of machine learning to annotate the remaining comments. This dataset can aid in the understanding of people's perceived barriers to living donation and the factors that motivate them to donate.

6.4.3 Training Phase

The dataset that we annotate has a few comments, which creates a possibility of losing the generality when building a classification model [103]. Annotating more comments is time-consuming, especially considering the need for constant collaboration to achieve consensus. We, therefore, use different strategies to cover the model's generality. We use the K-Fold validation procedure to guarantee that model generality.

We use a nested K-Fold model [104, 105] in the first iteration, and randomly separate 20% of the data to build the *validation dataset*. We split the rest (80%) of the data into ten separate folds. In our validation, we used tenfold validation Figure 48, Kohavi [106] empirically shows that tenfold performs better than other techniques, such as leave-one-out cross validation, or Krstajic [105] shows that tenfold perform better compared to ten fivefold. Thus, we separate our training set tenfold.

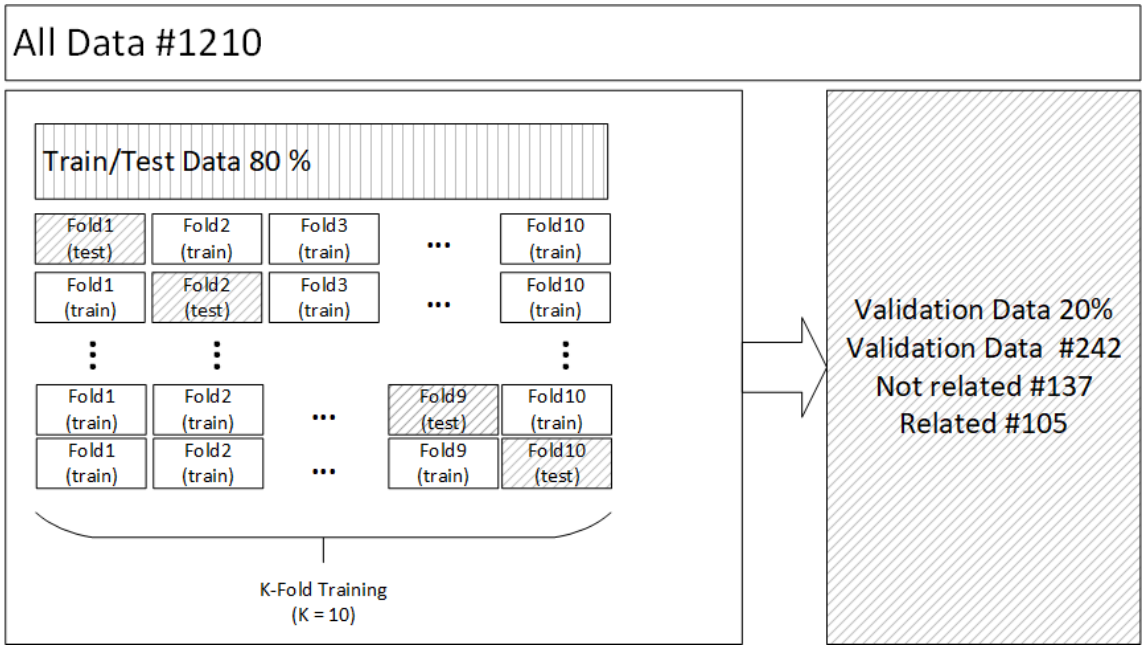

Figure 48 10-Fold Validation, We Have 1,210 Comments in Total and 20% or 242 Comments for Validation, and 10-Fold Training Run Based 80% For Training.

We use several metrics to evaluate the performance of the classification model, including precision, recall, and F1-Score. The precision metric explains "How many comments are classified as *related* out of all the comments," and the recall/sensitivity metric describes "How many comments, out of all the comments classified as 'related', were correctly classified". We also report the metric F1, which is the average of the Precision and Recall metrics.

6.4.4 Results

We separate our data into two classes, related and not related to organ provision. Figure 49 shows the most common words in each class; the y-axis shows the words, and the x-axis is the frequency of a word out of the 1,125 manually classified comments. We observe that the keywords "Kidney" and" Donate" are not the best features to classify the text in our research since they are widespread in both the related and not related classes. Words like

"Family, "Make," and "Someone" appear only in the "Related" class, so they seem to be useful for classification.

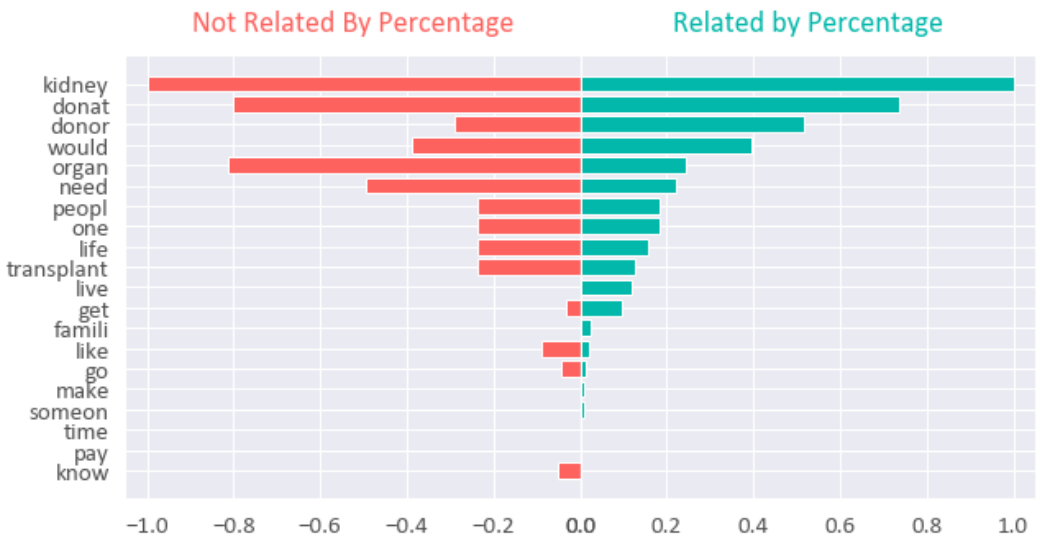

Figure 49 Frequency of Essential Words in Related and Not Related Class

F1-Score metrics, as we defined, were used to select the best model out of 1,125 for each fold. We then run the best model using the unseen Validation Data (20% of all the data), resulting in the F1-scores shown in Figure 50. We used the average of F1-Scores for each Word and Character level tokenization to select the most general model. The horizontal line representing the average F1-Score, and you find the closest corresponding model to that score from fold five using Word tokenization. We must select the models close to the middle line even though we have a model that yielded a 96% accuracy. We can conclude that models in fold-five must be a better choice for fulfilling the model's generalization.

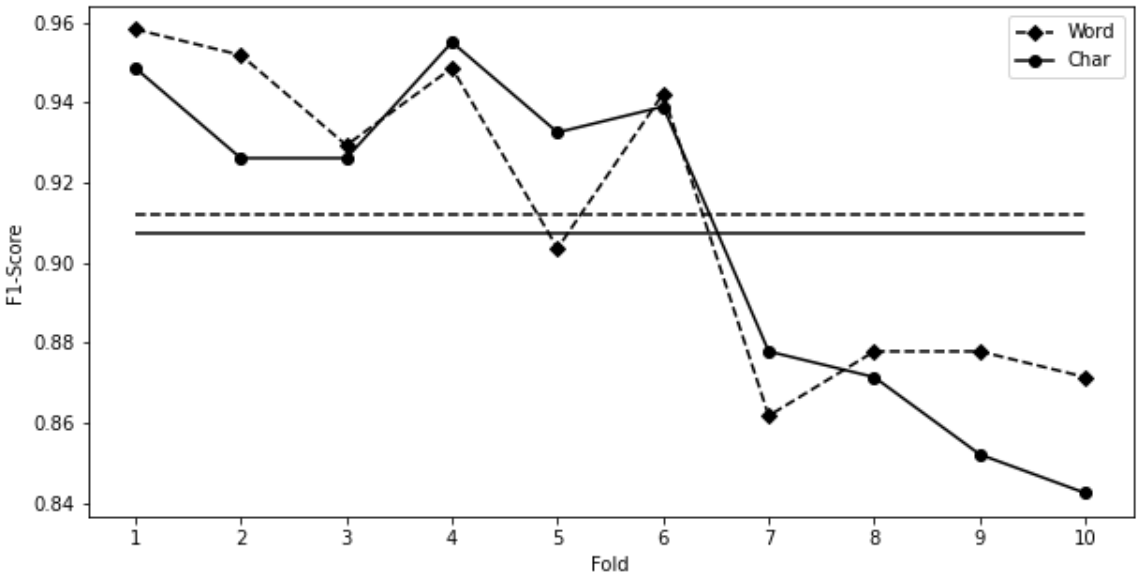

Figure 50 Outlines the 10-Fold Experiment: F1-Score Represents Each Iteration and the Horizontal Line Represents the Average F1-Score for All the Experiment

Tenfold strategy narrows the model selection process down to two models, and both models are in fold-five. To decide which one performs better for our classification, we used recall/sensitivity, which means "how many comments out of all the comments classified as related, were correctly classified," and precision, which means "How many comments are classified as *related* out of all the comments."

Figure 51 represents *related comments* results on the right side of the bar chart with green, and *unrelated comments* result on the left with the color red. We report recall and precision for each group individually.

The Word tokenization process successfully predicts the *unrelated* comments with a recall of 0.99. However, we can see that the model is biased to *unrelated* comments; thus, we miss classified 29 comments related to the kidney provision, resulting in a low recall 0.78 in the *related* comments group on the right. The same analysis on character tokenization shows better performance in two groups; as we can see, recall on both sides is 0.91.

Words distribution in each model gives insight into bias on word tokenization. As the bar chart represents, words like "want" and "medic" were the keywords to label the comments as *unrelated*; on the other hand, those words were removed when we used Character tokenization.

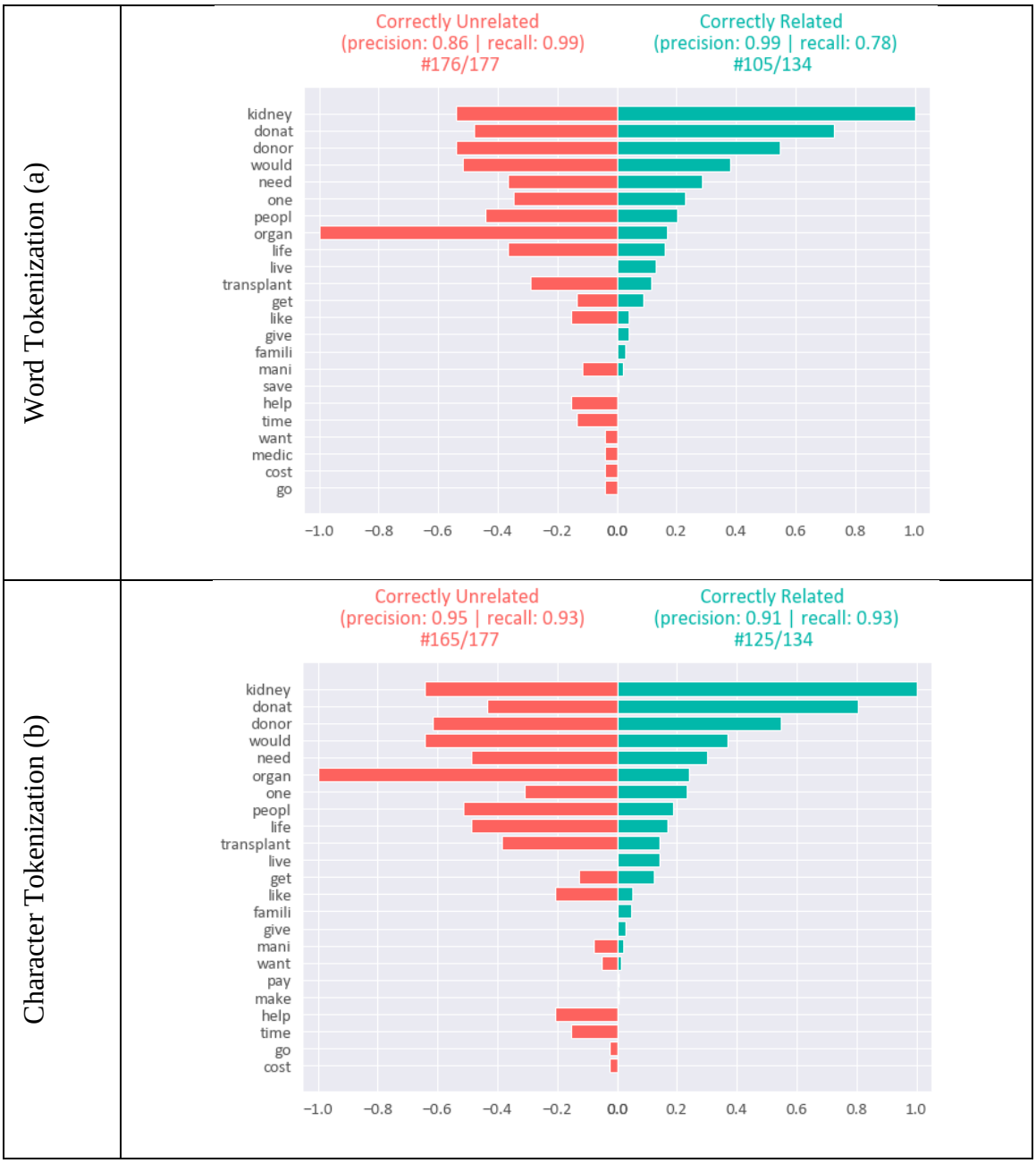

Figure 51 Comparison of Two best model select by 10-fold procedure; (a) The model with Word-Tokenization; (b) The model with character tokenization; each bar-chart have two side, left represents the word frequency for Unrelated and right Related ti Kidney provision comments.

To better understand, we analyze false negatives in statistics known as Error Type II. Figure 52 shows the word frequency on word tokenization (left in blue color) and character tokenization (right green color). Words like "want," "cost," "transplant," "get," "need," "live," "go," "ask," "think" make a comment related to kidney provision but appear as errors when we use word tokenization techniques.

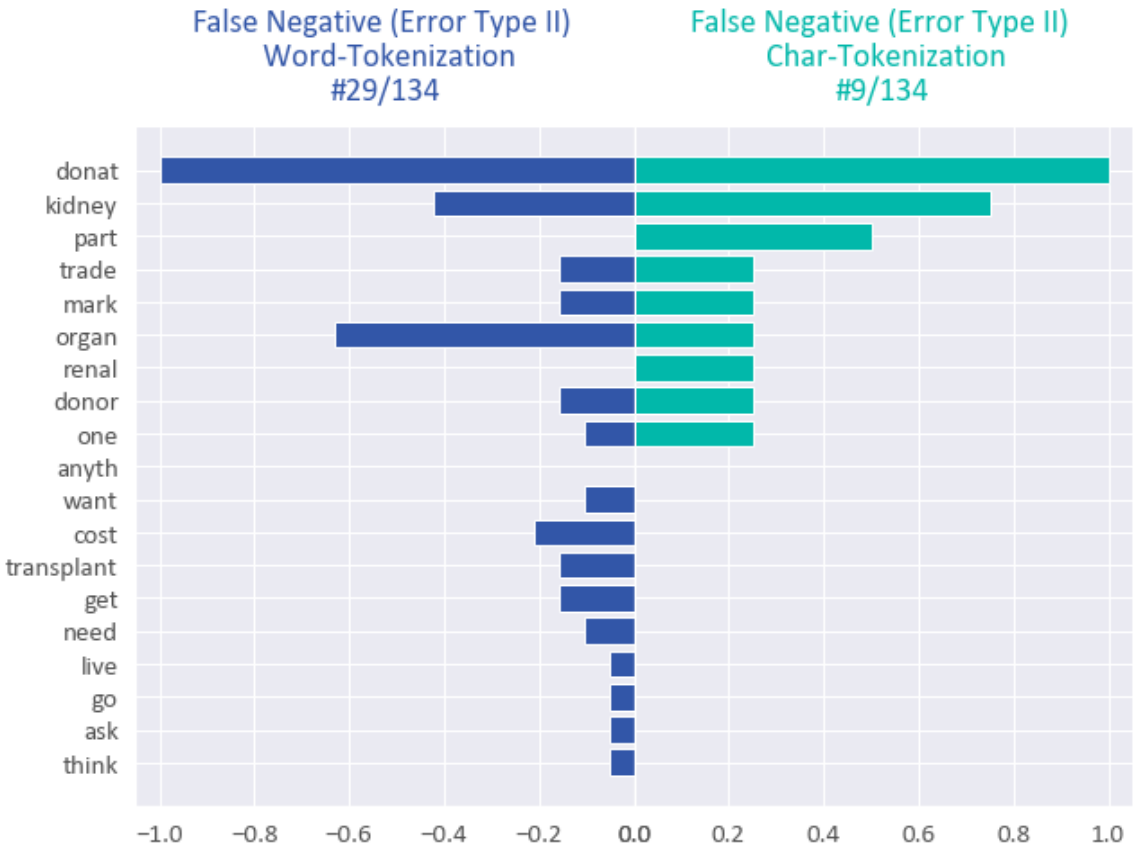

Figure 52 Word Frequency for False Negative (Error Type II) For Word Tokenization on the Left and Character Tokenization on the Right

In the end, we represent the best models with hyperparameters settings in Table 7. Also, the average Precision, Recall, and F1-score were reported for each model. We recommend using character tokenization with the window of 10 to process the text for better performance.

Table 7 Best Hyperparameters With Different Embedding, Embedding Size (ES), Batch Size (BS), Hidden Size (Hs), Learning Rate (LR)

Epoch	**Embedding**	**ES**	**BS**	**HS**	**LR**
Precision	Recall	F1-Score	Accuracy		
134	Word Tokenization	1204	32	400	10^{-5}
0.92	0.89	0.90	0.90		
132	Char Tokenization	600	16	50	10^{-5}
0.93	0.93	0.93	0.93		

CHAPTER 7

CONCLUSION AND FUTURE WORKS

This dissertation's research links different sources of data and creates a unified knowledge in healthcare, improving access to medical expertise for the public and experts. We use several techniques from semi-supervised classification, classification, deep learning, and transfer learning. As a result, we meet all the objectives that we define in the introduction. This study provides several contributions, including data gathering, introducing two databases, create a novel noise cancelation system for social media, creating a deep learning model for named entity recognition, and use transfer learning to generalize the model. These contributions result in published three conference papers and two journal publications. We have two more publications that are in the process of review by the journals.

- We fulfill the objectives one through three which they have focus on introducing a robust decision-making procedure for selecting a model to explain the active topics, using transfer learning techniques to enhance the framework's ability to detect unrelated messages over Twitter data streams. Though an automatic deep cleaning method, we strengthen data quality to perform better classification in a noisy environment. Lastly, we create a database that helps people study trending health topic seen on social media and introduce an end-to-end framework for this. We publish these findings in two papers:

Title: Trends on health in social media Analysis using Twitter topic modeling **Author: *Asghari, Mohsen and Sierra-Sosa, Daniel and Elmaghraby, Adel*** **Journal: 2018 IEEE international symposium on signal processing and information technology (ISSPIT)** **Year: 2018**
Title: Atopic modeling framework for Spatio-temporal information management **Author**: *Asghari, Mohsen and Sierra-Sosa, Daniel and Elmaghraby, Adel* **Journal**: Information Processing & Management Elsevier **Impact Factor:** 4.787 **Year**: 2020

This framework allows us to study and publish a paper related to the demographic influence of rehab centers on opioid misuse in the Louisville area. Resulting in the following paper:

Title: Demographic influence on opioid misuse

Author: ***Asghari, Mohsen and Sierra-Sosa, Daniel and Elmaghraby, Adel***

Journal: 2018 IEEE international symposium on signal processing and information technology (ISSPIT)

Year: 2018

- We create a concept drift indicator to enhance the topic modeling and classification models. We propose and evaluate a novel concept drift detection model. This model helps detect covariate shift streaming data. It uses an unsupervised machine learning approach, which removes the data labeling burden. We base our CDIC indicator on a distance definition and statistical analysis and eliminate the need for feature analysis in the detection of concept drift. This method is flexible as it allows a user to select the distance function based on their data type and distribution. The proposed method illustrates that we can detect drift without considering each feature. We tested this approach using synthetic and real datasets. In all cases, we obtain accurate and consistent results. Catching the drift at the right time and the right place can be critical in many applications to make better decisions. CDIC can help data scientists better understand their data and use it in machine learning applications in dynamic streaming data environments.

Title: Aggregate density-based concept drift identification for dynamic sensor data models

Author: ***Asghari, Mohsen and Sierra-Sosa, Daniel and Telahun, Michael and Kumar, Anup and Elmaghraby, Adel S***

Journal: Neural Computing and Applications Year: 2020 Impact Factor: 4.774

- We introduce low-cost Named Entity Recognition in the Biomedical space using deep learning techniques. By experimenting with different Embedding Layers combined with Bidirectional LSTM and Conditional Random Field to do Named Entity Recognition in the Biomedical Domain, we show different embedding effects layers and architectures. We use and test different architectures and identify a Parallel BINER model that outperforms other architectures. Our proposed model requires fewer layers than Bert and BioBERT. We define this model as having two primary layers, each of them has 10 Layers with 400 Hidden size (32 Million Total Parameters). We train this model in 7 days using NVIDIA 2080 GTX; the model performance did not affect BioBert. Our results compare favorably with BioBert tested with five different standard databases. We improved the F1-Score over BioBert by 3% on Disease detection, 8% on detecting Protein (Contains DNA, RNA, and Cell), and 9% for Gene detection.

Title: BINER: A Low-cost Biomedical Named Entity Recognition

Author: ***Asghari, Mohsen and Sierra-Sosa, Daniel and Elmaghraby, Adel S***

Journal: Information Sciences Submit Data: March 2021

Journal: Information Sciences Year: 2021 Impact Factor: 6.795

Accepted (July 2021)

- We use transfer learning to create a multi-source named entity recognition model to handle biomedical text and clinical text. Retraining the model might be sound easy, however, preparing and annotating new data is a time-consuming and expensive process. We design an experiment to study the training size effect in transfer learning. During this part of the dissertation, we realize we could enhance and improve the process after training on 30% of the data. However, this result cannot generalize all types of transfer learning; each domain has various levels of complexity in its data and many other factors change this percentage. However, our findings illustrate the possibilities of using this state-of-the-art technique alongside a standard data set used extensively by researchers and annotated by the expert of the field.
- We create a cohesive, unified framework to extract biomedical, clinical text, and social media knowledge.

REFERENCES

1. Liu, F., C. Weng, and H. Yu, *Natural language processing, electronic health records, and clinical research*, in *Clinical Research Informatics*. 2012, Springer. p. 293-310.
2. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* nature, 2015. **521**(7553): p. 436-444.
3. Shickel, B., et al., *Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis.* IEEE journal of biomedical and health informatics, 2017. **22**(5): p. 1589-1604.
4. Hodapp, C., *Unsupervised learning for computational phenotyping.* arXiv preprint arXiv:1612.08425, 2016.
5. Che, Z., et al. *Deep computational phenotyping*. in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2015.
6. Choi, E., et al., *Medical concept representation learning from electronic health records and its application on heart failure prediction.* arXiv preprint arXiv:1602.03686, 2016.
7. Tran, T., et al., *Learning vector representation of medical objects via EMR-driven nonnegative restricted Boltzmann machines (eNRBM).* Journal of biomedical informatics, 2015. **54**: p. 96-105.
8. Nguyen, P., et al., *\mathtt {Deepr} : a convolutional net for medical records.* IEEE journal of biomedical and health informatics, 2016. **21**(1): p. 22-30.
9. Beam, A.L., et al. *Clinical concept embeddings learned from massive sources of multimodal medical data*. in *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2020*. 2019. World Scientific.
10. Pennington, J., R. Socher, and C.D. Manning. *Glove: Global vectors for word representation*. in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.
11. Minarro-Giménez, J.A., O. Marín-Alonso, and M. Samwald, *Applying deep learning techniques on medical corpora from the world wide web: a prototypical system and evaluation.* arXiv preprint arXiv:1502.03682, 2015.
12. Kuperman, G.J., et al., *Medication-related clinical decision support in computerized provider order entry systems: a review.* Journal of the American Medical Informatics Association, 2007. **14**(1): p. 29-40.
13. Bates, D.W., et al., *Big data in health care: using analytics to identify and manage high-risk and high-cost patients.* Health Affairs, 2014. **33**(7): p. 1123-1131.
14. Reddy, C.K. and C.C. Aggarwal, *Healthcare data analytics*. Vol. 36. 2015: CRC Press.
15. Chapman, B.E., et al., *Document-level classification of CT pulmonary angiography reports based on an extension of the ConText algorithm.* Journal of Biomedical Informatics, 2011. **44**(5): p. 728-737.

16. Uzuner, Ö., X. Zhang, and T. Sibanda, *Machine learning and rule-based approaches to assertion classification.* Journal of the American Medical Informatics Association, 2009. **16**(1): p. 109-115.
17. Mowery, D., *Developing a Clinical Linguistic Framework for Problem List Generation from Clinical Text*. 2014, University of Pittsburgh.
18. Chapman, W., J. Dowling, and D. Chu. *ConText: An algorithm for identifying contextual features from clinical text*. in *Biological, translational, and clinical language processing*. 2007.
19. Gyrard, A., et al. *Knowledge Extraction for the Web of Things (KE4WoT) WWW 2018 Challenge Summary*. in *Companion Proceedings of the The Web Conference 2018*. 2018.
20. Asghari, M., D. Sierra-Sosa, and A. Elmaghraby. *A Semi-Automatic System for Data Management and Cleaning*. in *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. 2018. IEEE.
21. Asghari, M., D. Sierra-Sosa, and A.S. Elmaghraby, *A topic modeling framework for spatio-temporal information management.* Inf Process Manag, 2020. **57**(6): p. 102340.
22. Sierra-Sosa, D., et al., *Demographic Influence on Opioid Misuse.* 2019 Fifth International Conference on Advances in Biomedical Engineering (ICABME), 2019: p. 1--4.
23. Asghari, M., D. Sierra-Sosa, and A. Elmaghraby, *Trends on Health in Social Media: Analysis using Twitter Topic Modeling.* 2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), 2018: p. 558--563.
24. Deerwester, S., et al., *Indexing by latent semantic analysis.* Journal of the American society for information science, 1990. **41**(6): p. 391--407.
25. Dumais, S.T., *Latent semantic analysis.* Annual review of information science and technology, 2004. **38**(1): p. 188--230.
26. Blei, D.M., A.Y. Ng, and M.I. Jordan, *Latent dirichlet allocation.* Journal of machine Learning research, 2003. **3**(1): p. 993--1022.
27. McCallum, A.K., *Mallet: A machine learning for language toolkit.* http://mallet. cs. umass. edu, 2002.
28. Yan, X., et al., *A biterm topic model for short texts.* The 22nd international conference on World Wide Web, 2013: p. 1445--1456.
29. Scanfeld, D., V. Scanfeld, and E.L. Larson, *Dissemination of health information through social networks: Twitter and antibiotics.* American journal of infection control, 2010. **38**(3): p. 182-188.
30. Prier, K.W., et al. *Identifying health-related topics on twitter*. in *International conference on social computing, behavioral-cultural modeling, and prediction*. 2011. Springer.
31. Terry, M., *Twittering Healthcare: Social Media and Medicine.* Telemedicine and e-Health, 2009. **15**(6): p. 507-510.
32. Surian, D., et al., *Characterizing Twitter discussions about HPV vaccines using topic modeling and community detection.* Journal of medical Internet research, 2016. **18**(8): p. e232.

33. Hwang, M.-H., et al., *Spatiotemporal transformation of social media geostreams: a case study of twitter for flu risk analysis.* The 4th ACM SIGSPATIAL International Workshop on GeoStreaming, 2013: p. 12--21.
34. Zhao, L., et al., *Spatiotemporal event forecasting in social media.* the 2015 SIAM international conference on data mining, 2015: p. 963--971.
35. Laylavi, F., A. Rajabifard, and M. Kalantari, *Event relatedness assessment of Twitter messages for emergency response.* Information processing \& management, 2017. **53**(1): p. 266--280.
36. Zahra, K., M. Imran, and F.O. Ostermann, *Automatic identification of eyewitness messages on twitter during disasters.* Information processing \& management, 2020. **57**(1): p. 102107.
37. Thom, D., et al., *Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages.* The 2012 IEEE Pacific Visualization Symposium, 2012: p. 41--48.
38. Guo, L., et al., *Big social data analytics in journalism and mass communication: Comparing dictionary-based text analysis and unsupervised topic modeling.* Journalism \& Mass Communication Quarterly, 2016. **93**(2): p. 332--359.
39. Rathore, M.M., et al., *Advanced computing model for geosocial media using big data analytics.* Multimedia Tools and Applications, 2017. **76**(23): p. 24767--24787.
40. Krestel, R., P. Fankhauser, and W. Nejdl, *Latent dirichlet allocation for tag recommendation.* The third ACM conference on Recommender systems, 2009: p. 61--68.
41. Șerban, O., et al., *Real-time processing of social media with SENTINEL: a syndromic surveillance system incorporating deep learning for health classification.* Information Processing \& Management, 2019. **56**(3): p. 1166--1184.
42. Koylu, C., *Modeling and visualizing semantic and spatio-temporal evolution of topics in interpersonal communication on Twitter.* International Journal of Geographical Information Science, 2019. **33**(4): p. 805--832.
43. Lee, J., et al., *BioBERT: pre-trained biomedical language representation model for biomedical text mining.* arXiv preprint arXiv:1901.08746, 2019.
44. Wang, X., et al., *Cross-type biomedical named entity recognition with deep multi-task learning.* Bioinformatics, 2019. **35**(10): p. 1745-1752.
45. Marasovi, *Srl4orl: Improving opinion role labeling using multi-task learning with semantic role labeling.* arXiv preprint arXiv:1711.00768, 2017.
46. Ma, X. and E. Hovy, *End-to-end sequence labeling via bi-directional lstm-cnns-crf.* arXiv preprint arXiv:1603.01354, 2016.
47. Chiu, J.P.C. and E. Nichols, *Named entity recognition with bidirectional LSTM-CNNs.* Transactions of the Association for Computational Linguistics, 2016. **4**: p. 357--370.
48. Santos, C.D. and B. Zadrozny, *Learning character-level representations for part-of-speech tagging.* 2014: p. 1818--1826.
49. Mikolov, T., et al., *Recurrent neural network based language model.* 2010.
50. Zaremba, W., I. Sutskever, and O. Vinyals, *Recurrent neural network regularization.* arXiv preprint arXiv:1409.2329, 2014.

51. Pennington, J., R. Socher, and C.D. Manning, *Glove: Global vectors for word representation.* 2014: p. 1532--1543.
52. Mikolov, T., et al., *Efficient estimation of word representations in vector space.* arXiv preprint arXiv:1301.3781, 2013.
53. Kitaev, N. and D. Klein, *Constituency parsing with a self-attentive encoder.* arXiv preprint arXiv:1805.01052, 2018.
54. Peters, M.E., et al., *Deep contextualized word representations.* arXiv preprint arXiv:1802.05365, 2018.
55. Svenstrup, D., J.M. Hansen, and O. Winther, *Hash embeddings for efficient word representations.* arXiv preprint arXiv:1709.03933, 2017.
56. Lafferty, J., A. McCallum, and F.C.N. Pereira, *Conditional random fields: Probabilistic models for segmenting and labeling sequence data.* 2001.
57. McCallum, A., D. Freitag, and F.C.N. Pereira, *Maximum Entropy Markov Models for Information Extraction and Segmentation.* 2000. **17**: p. 591--598.
58. Ratnaparkhi, A., *A maximum entropy model for part-of-speech tagging.* 1996.
59. Huang, Z., W. Xu, and K. Yu, *Bidirectional LSTM-CRF models for sequence tagging.* arXiv preprint arXiv:1508.01991, 2015.
60. Kim, J.-D., et al., *Introduction to the bio-entity recognition task at JNLPBA.* 2004: p. 70--75.
61. Smith, L., et al., *Overview of BioCreative II gene mention recognition.* Genome Biology, 2008. **9**(S2): p. S2.
62. Leaman, R. and G. Gonzalez, *BANNER: an executable survey of advances in biomedical named entity recognition.* Pac Symp Biocomput, 2008: p. 652-63.
63. Campos, D., S. Matos, and J.L. Oliveira, *Gimli: open source and high-performance biomedical name recognition.* BMC Bioinformatics, 2013. **14**(1): p. 54.
64. Mi, H. and P. Thomas, *PANTHER pathway: an ontology-based pathway database coupled with data analysis tools.* Methods Mol Biol, 2009. **563**: p. 123-40.
65. Gerner, M., G. Nenadic, and C.M. Bergman, *LINNAEUS: A species name identification system for biomedical literature.* BMC Bioinformatics, 2010. **11**(1): p. 85.
66. Doan, R.I., R. Leaman, and Z. Lu, *NCBI disease corpus: a resource for disease name recognition and concept normalization.* Journal of biomedical informatics, 2014. **47**: p. 1--10.
67. Li, J., et al., *BioCreative V CDR task corpus: a resource for chemical disease relation extraction.* Database, 2016. **2016**.
68. Suominen, H., et al. *Overview of the ShARe/CLEF eHealth evaluation lab 2013*. in *International Conference of the Cross-Language Evaluation Forum for European Languages*. 2013. Springer.
69. Kelly, L., et al. *Overview of the share/clef ehealth evaluation lab 2014*. in *International Conference of the Cross-Language Evaluation Forum for European Languages*. 2014. Springer.
70. Bodenreider, O., *The Unified Medical Language System (UMLS): integrating biomedical terminology.* Nucleic Acids Research, 2004. **32**(90001): p. 267D-270.
71. Smith, B. and C. Fellbaum, *Medical WordNet: a new methodology for the construction and validation of information resources for consumer health.* 2004.

72. Beeferman, D. *The Datamuse API*. 2019; Available from: http://www.datamuse.com.
73. Asghari, M., D. Sierra-Sosa, and A. Elmaghraby. *Trends on health in social media: Analysis using twitter topic modeling*. in *2018 IEEE international symposium on signal processing and information technology (ISSPIT)*. 2018. IEEE.
74. Rosenberg, A. and J. Hirschberg, *V-measure: A conditional entropy-based external cluster evaluation measure.* The 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), 2007: p. 410--420.
75. Rder, M., A. Both, and A. Hinneburg, *Exploring the space of topic coherence measures.* The eighth ACM international conference on Web search and data mining, 2015: p. 399--408.
76. Albishre, K., M. Albathan, and Y. Li, *Effective 20 newsgroups dataset cleaning.* The 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2015. **3**: p. 98--101.
77. Fralick, M. and A.S. Kesselheim, *The US Insulin Crisis-Rationing a Lifesaving Medication Discovered in the 1920s.* The New England journal of medicine, 2019. **381**(19): p. 1793.
78. Yen, S.-J., et al., *A support vector machine-based context-ranking model for question answering.* Information Sciences, 2013. **224**: p. 77--87.
79. Vanetik, N., et al., *An unsupervised constrained optimization approach to compressive summarization.* Information Sciences, 2020. **509**: p. 22--35.
80. Ratinov, L. and D. Roth, *Design challenges and misconceptions in named entity recognition.* 2009: p. 147--155.
81. Passos, A., V. Kumar, and A. McCallum, *Lexicon infused phrase embeddings for named entity resolution.* arXiv preprint arXiv:1404.5367, 2014.
82. Luo, G., et al., *Joint entity recognition and disambiguation.* 2015: p. 879--888.
83. Fei, H., Y. Ren, and D. Ji, *Dispatched attention with multi-task learning for nested mention recognition.* Information Sciences, 2020. **513**: p. 241--251.
84. Devlin, J., et al., *Bert: Pre-training of deep bidirectional transformers for language understanding.* arXiv preprint arXiv:1810.04805, 2018.
85. Opitz, J. and S. Burst, *Macro F1 and Macro F1*. arXiv preprint arXiv:1911.03347, 2019.
86. Buchan, K., *Annotation Guidelines for the Adverse Drug Event (ADE) and Medication Extraction Challenge.* n2c2, US, 2018.
87. DeHart, K. and J. Holbrook, *Emergency department applications of digital dictation and natural language processing.* The Journal of ambulatory care management, 1992. **15**(4): p. 18-23.
88. Brown, T.B., et al., *Language models are few-shot learners.* arXiv preprint arXiv:2005.14165, 2020.
89. Wang, J., et al., *Accelerating Epidemiological Investigation Analysis by Using NLP and Knowledge Reasoning: A Case Study on COVID-19.* AMIA Annu Symp Proc, 2020. **2020**: p. 1258-1267.
90. Shen, T.S., et al., *COVID-19-Related Internet Search Patterns Among People in the United States: Exploratory Analysis.* J Med Internet Res, 2020. **22**(11): p. e22407.

91. Izquierdo, J.L., et al., *Clinical Characteristics and Prognostic Factors for Intensive Care Unit Admission of Patients With COVID-19: Retrospective Study Using Machine Learning and Natural Language Processing.* J Med Internet Res, 2020. **22**(10): p. e21801.
92. Low, D.M., et al., *Natural Language Processing Reveals Vulnerable Mental Health Support Groups and Heightened Health Anxiety on Reddit During COVID-19: Observational Study.* J Med Internet Res, 2020. **22**(10): p. e22635.
93. Neuraz, A., et al., *Natural Language Processing for Rapid Response to Emergent Diseases: Case Study of Calcium Channel Blockers and Hypertension in the COVID-19 Pandemic.* J Med Internet Res, 2020. **22**(8): p. e20773.
94. Jelodar, H., et al., *Deep Sentiment Classification and Topic Discovery on Novel Coronavirus or COVID-19 Online Discussions: NLP Using LSTM Recurrent Neural Network Approach.* IEEE J Biomed Health Inform, 2020. **24**(10): p. 2733-2742.
95. Mackey, T.K., et al., *Big Data, Natural Language Processing, and Deep Learning to Detect and Characterize Illicit COVID-19 Product Sales: Infoveillance Study on Twitter and Instagram.* JMIR Public Health Surveill, 2020. **6**(3): p. e20794.
96. Odlum, M., et al., *Application of Topic Modeling to Tweets as the Foundation for Health Disparity Research for COVID-19.* Stud Health Technol Inform, 2020. **272**: p. 24-27.
97. Zheng, Y., et al., *The hemocyte counts as a potential biomarker for predicting disease progression in COVID-19: a retrospective study.* Clin Chem Lab Med, 2020. **58**(7): p. 1106-1115.
98. Wei, C.-H., et al., *PubTator central: automated concept annotation for biomedical full text articles.* Nucleic acids research, 2019. **47**(W1): p. W587-W593.
99. Wei, C.-H., H.-Y. Kao, and Z. Lu, *PubTator: a web-based text mining tool for assisting biocuration.* Nucleic acids research, 2013. **41**(W1): p. W518-W522.
100. Comeau, D.C., et al., *PMC text mining subset in BioC: about three million full-text articles and growing.* Bioinformatics, 2019. **35**(18): p. 3533-3535.
101. Zhou, P., et al., *Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling.* arXiv preprint arXiv:1611.06639, 2016.
102. Zhang, Z. and M.R. Sabuncu, *Generalized cross entropy loss for training deep neural networks with noisy labels.* arXiv preprint arXiv:1805.07836, 2018.
103. Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980, 2014.
104. Cawley, G.C. and N.L. Talbot, *On over-fitting in model selection and subsequent selection bias in performance evaluation.* The Journal of Machine Learning Research, 2010. **11**: p. 2079-2107.
105. Krstajic, D., et al., *Cross-validation pitfalls when selecting and assessing regression and classification models.* Journal of cheminformatics, 2014. **6**(1): p. 1-15.
106. Kohavi, R. *A study of cross-validation and bootstrap for accuracy estimation and model selection*. in *Ijcai*. 1995. Montreal, Canada.

CURRICULUM VITA

Mohsen Asghari

Research Interest

Natural Language Processing, Natural Language Understanding, Deep Learning, Machine Learning, Multi Models, Topic Modeling

Education

Ph.D. University of Louisville Computer Science

Louisville, KY 2017- August 2021

Master of Science in Information Technology, **K.N.T University of Technology**

Major: System Management Tehran, Iran 2012-2014

Bachelor of Science in Software Engineering, **University Central Tehran**

Computer Software Engineering Tehran, Iran 2004-2008

Publications

- **2021** BINERI: Biomedical Named Entity Recognition by Integrating Deep Learning Approaches (**Information Science**)
- **2020** A Topic Modeling Framework for Spatio-Temporal Information Management (**Journal of Information Processing & management**)
- **2020** Using AI for IoT Dynamic Sensor Data Models (**Journal of neural Computing and Application**)
- **2019** Demographic Influence on Opioid Misuse (2019 Fifth International Conference on Advances in Biomedical Engineering (ICABME))
- **2018** Trends on Health in Social Media: Analysis using Twitter Topic Modeling (2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)

- **2018** A Semi-Automatic System for Data Management and Cleaning (2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT))
- **2017** Analysis of the Factors on Intrauterine Insemination (IUI) Results by Clustering
- **2015** A New Similarity Measure by Combining Formal Concept Analysis and Clustering for Case-Based Reasoning
- **2014** Proposing a prediction model for diagnosing causes of infertility by data mining algorithms (Journal of Health Administration)
- **2013** Utilizing data mining techniques for investigating factors influencing the failure of Intrauterine Insemination infertility treatment. (Journal of Health Administration (JHA))

Awards

Computer Science and EngineeringCSE Doctoral Award. 2021

- For theComputer Science and Engineering doctoral student selected by the CSE departmentwith priority given to those pursuing academic/research careers.

Raymon I. Fields Award-April 2019

- For the Computer Engineering and Computer Science graduate who contribute the most to the department and school in leadership and service.

Full-Scholarship for Midwest SAS User group conference-June 2017

- Presenting "Dimensionality Reduction using Hadamard, Discrete Cosine and Discrete Fourier Transforms in SAS"

Certifications

- **2020** IBM Quantum Practitioner
- **2020** IBM Data Science Practitioner
- **2010** Microsoft Certification of "Microsoft.NET Framework
  - Application Development Foundation
  - (Certification ID: 6932701)
- **2009** Microsoft Certification of "Microsoft.NET Framework
  - 3.5, ASP.NET Application Development
  - (Certification ID: 6932701)