# EXTRACTING SYMPTOMS FROM NARRATIVE TEXT USING ARTIFICIAL INTELLIGENCE

by

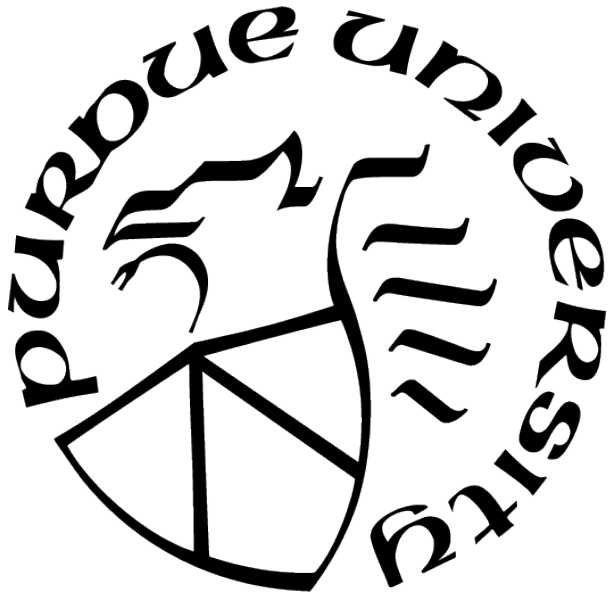
**Priyanka Gandhi**

**A Thesis**

*Submitted to the Faculty of Purdue University*

*In Partial Fulfillment of the Requirements for the degree of*

**Master of Science**




Department of Computer and Information Science

Indianapolis, Indiana

December 2020

# THE PURDUE UNIVERSITY GRADUATE SCHOOL
# STATEMENT OF COMMITTEE APPROVAL

**Dr. Xukai Zou, Chair**

Computer and Information Science

**Dr. Xiao Luo**

Computer and Information Technology

**Dr. Yuni Xia**

Computer and Information Science

**Approved by:**

Dr. Shiaofen Fang

To my parents,
Rakesh Gandhi, and
Keyoori Gandhi.

# ACKNOWLEDGMENTS

Foremost, I would like to thank my mother Keyoori Gandhi, my father Rakesh Gandhi, my grandfather Jayesh Gandhi, my sister Noopur Gandhi, my brother Raj Gandhi and my entire family for their unparalleled love and support. They have always believed in my abilities, and I dedicate this milestone to them.

I would like to sincerely thank Dr. Xiao Luo for her patience, encouragement, and direction that has had an undeniable impact on every step of this work. Thank you for the skills and knowledge you have imparted to me.

I owe my deepest gratitude to Dr. Xukai Zuo for his continuous guidance, encouragement, and mentorship. Thank you for having confidence in me and introducing me to research.

I would also like to thank Dr. Yuni Xia for agreeing to serve on my committee. Thank you for providing great insights and feedback on my thesis.

Last but not least, I would like to thank Ishani Mehta, Namita Gupta, and all my friends who have always been there for me.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ABSTRACT

Electronic health records collect an enormous amount of data about patients. However, the information about the patient's illness is stored in progress notes that are in an unstructured format. It is difficult for humans to annotate symptoms listed in the free text. Recently, researchers have explored the advancements of deep learning can be applied to process biomedical data. The information in the text can be extracted with the help of natural language processing. The research presented in this thesis aims at automating the process of symptom extraction. The proposed methods use pre-trained word embeddings such as BioWord2Vec, BERT, and BioBERT to generate vectors of the words based on semantics and syntactic structure of sentences. BioWord2Vec embeddings are fed into a BiLSTM neural network with a CRF layer to capture the dependencies between the co-related terms in the sentence. The pre-trained BERT and BioBERT embeddings are fed into the BERT model with a CRF layer to analyze the output tags of neighboring tokens. The research shows that with the help of the CRF layer in neural network models, longer phrases of symptoms can be extracted from the text. The proposed models are compared with the UMLS Metamap tool that uses various sources to categorize the terms in the text to different semantic types and Stanford CoreNLP, a dependency parser, that analyses syntactic relations in the sentence to extract information. The performance of the models is analyzed by using strict, relaxed, and n-gram evaluation schemes. The results show BioBERT with a CRF layer can extract the majority of the human-labeled symptoms. Furthermore, the model is used to extract symptoms from COVID-19 tweets. The model was able to extract symptoms listed by CDC as well as new symptoms.

# 1. INTRODUCTION

Doctor's short text notes on patient's illness, have now grown to a collection of large medical data through an electronic health record (EHR) systems. It is astonishing how technology radically changed over the past few years. Various problems are now faced while analyzing biomedical text. These massive datasets consist of demographic information, allergies, immunizations, diagnosis, medications, etc. However, the data collected does not contain symptoms observed by the patients in a structured format. Therefore, clinicians are required to manually annotate the symptoms by analyzing the clinical notes. There have been several studies suggesting that the EHR system has led to clinician's burnout by imposing a lot of documentation pressure. Clinicians are required to do excessive documentation to summarize and understand the illness suffered by patients. Artificial Intelligence (AI) uses complex reasoning and superior analytical algorithms to accomplish tasks at a higher scale, allowing humans to direct their time and energy to other productive tasks. AI simulates human intelligence and mimics cognitive abilities to study and solve problems. The machine makes use of Natural Language Processing (NLP), a field in AI, to understand human language. To create a smart system for understanding, parsing, and extracting information from the data, NLP combines linguistics and computer science. NLP is a study of analyzing lexicons, syntactic structure, semantics, the dependency of previous and next sentences, and pragmatics.

Over the past decade, researchers have found a great interest in implementing NLP to extract relevant and important information from the corpus. With the expanding deployment of EHRs in clinical settings, a huge volume of data on the patients that have been collected needs to be processed. These notes are highly valuable, consisting of current illness, past clinical history, medical history of the family, treatment, and vaccination, etc. One way to reduce the burden on clinicians and improve efficiency is by automating operations. The increase in the usage of technology has caused human intervention to fall drastically. New advances in technology aim to reduce human efforts, error rate, and time. Standard details about the patients are collected in a structured form. However, important information about the patient is obtained in unstructured free text. It is now possible to extract specific details

from free-text clinical notes. It is important to note, free-text clinical notes are unstructured, loaded with spelling mistakes, and comprise of medical terminologies.

A symptom can also be termed as a "clinical predicament", "diagnosis" or a "noted symptom". For instance, "ulcer" could be defined as any form of symptoms. Therefore, in the dataset clinician has tagged all those symptoms that are listed by the doctors and patients as symptoms. The EHR records extracted contain various types of records making it difficult for an ideal model to identify symptoms. It is important for models to consider the semantic context and linguistic relativity and not simply match strings as the records contain:

- questionable complaints such as "Patient might have had a fever."
- an indirect indication of a symptom such as "Patient takes medicines to breathe."
- negative statements such as "Patient had no chest pain or cough."
- a diagnosis such as "Patient is now showing symptoms of CHF"
- conditional symptoms such as "If the patient has a fever then visit a doctor."
- condition description to explain medication such as "Tab 1/2 -1 tablet at bedtime as needed for insomnia."
- informative statements such as "allergy to aspirin might cause itchy spots."

These sentences are some of the special cases that require the analytical skills to understand the context and tag relevant symptoms encountered in the records. Changing over the free-text of clinical notes into an appropriate format that can be fed into the machine learning models stays one of the main difficulties in the medical domain. Deep learning is highly dependent on labeled information. When implementing these machine learning algorithms on domain-related tasks, their primary issue lies in their requirement for significant human-annotated training corpus, which needs repetitive and costly work from domain specialists. The goal is to model an advanced neural network that can annotate samples and

automate the extraction of symptoms of any disease. The process of extracting and summarizing information from the free-text obtained by the EHR system is known as information extraction (IE) [1].

Progress in machine learning (ML) and NLP algorithms have enhanced the ability of computerized systems to mine data. It is now possible for computers to automate the classification process of documents, generate medical texts, concise patient illness, and answer medical-related questions. It is not feasible for a human to annotate all EHR recordings. Therefore, for this research, a subset of the dataset was first annotated by a clinician to identify all the symptoms in those recordings. To distinguish entities within clinical notes, named entity recognition (NER) is applied. NER is capable of automatically annotating entities, in this study, symptoms. There has been very limited research done to extract symptoms from the free text.

The research presented in this paper is different from previous research done in this domain as it includes:

1. This work is the first to automate the process of symptom extraction from the narrative text.

2. This work is the first to integrate neural networks with a CRF layer to annotate the symptoms. The neural networks implemented in this research include a deep neural network, a bidirectional LSTM, and BERT.

3. The models implemented are compared against the human-annotated symptoms as well as UMLS Metamap, a tool used to identify concepts in the biomedical text.

4. This work is the first to extract symptoms of the COVID-19 illness using Twitter tweets.

This research focuses on symptom extraction using neural networks. A discussion about the NLP concepts used to process the free-text obtained from EHR systems is presented in Chapter 3. Chapter 4 presents the related work done in this domain. Chapter 5 presents a detailed description of the models implemented for symptom extraction. The process of extraction, cleaning, and annotation of the dataset is described in Chapter 6. A subset of

annotated data by the clinician is compared with the other models implemented to evaluate the performance of models is presented in Chapter 7.

# 2. ELECTRONIC HEALTH RECORDS

An Electronic Health Record (EHR) is an electronic variant of a patient's clinical history, that is recorded over time, and may incorporate all the clinical information applicable to that individual including progress notes, medications, diagnosis, demographics, allergies, past clinical history, vaccinations and reports [2]. Table 2.1 shows the description of various types of data collected by the EHR system. Since EHR acts as a large repository of different types of data with personal details of the patients, it needs to be handled responsibly and not violate the clause of confidentiality.

EHR systems are deployed by the health care industry to gather and store the patient's clinical history. They act like patient-centered registries, designed for a purpose of extracting information by stating certain conditions [13]. EHR systems are utilized over clinical care and healthcare organization to capture an assortment of medical data over time, as well as to oversee clinical workflows. As per the National Academies of Medicine, an EHR does not restrict to the collection of patient's details but also supports many major functionalities, like capturing health data, orders and administration, clinical decision support, health data exchange, electronic communication, patient support, regulatory forms, and populace health detailing [14].

Over the past decade, the health care industry has widely accepted and promoted the use of EHR systems, partially because of the Health Information Technology for Economic and Clinical Health (HITECH) Act of 2009, encouraged medical clinics and hospitals to adopt EHR systems [15]. Initially, EHR systems were intended for operational purposes and later made use to process information. All the data collected by the EHR system can now be used for analysis. The large volume of data collected by progress notes and the swift increase in the adoption of EHR systems has led to an important research field of medical predictive analytic, that makes use of narrative progress notes.

Some of the data extracted from the EHR system, like medication, diagnosis, and demographics are in a structured format that can be used for data mining. However, the major chunk of data is in the unstructured format of progress notes. These are narratives that are an important form of communication, delivers a customized record of patient history with its

**Table 2.1.** Type of Data stored in EHR system.

| Data Type | Information |
|---|---|
| **Demographics** | Demographic contains socio-economic information about the patient. The collection of this data is authorized by the Meaningful Use(MU) objectives [3]. |
| **Diagnosis** | Diagnosis data should be rich and should meet the standards that are defined [4]. |
| **Problem List** | Problem list helps to differentiate between active and non-active diagnosis. |
| **Family History** | Data to know if any familial disorders, inherited diseases or risks involved. |
| **Allergies** | This helps to treat patients and can help in research purposes to know the effect of treatments on a particular diagnosis. |
| **Immunization** | Details about the vaccines given to the patient. |
| **Medications** | Medication data is recorded to keep a track of treatments on the patient and for research purposes about the treatment effects. Common standards to record medication data are NDC [5], RxNorm [6], SNOMED [7] and ATC [8] |
| **Procedures** | The procedure includes data about the surgery, radiology, pathology and laboratory undergone by the patient. Vocabulary standards for procedures are stated in ICD-CM [9], CPT [10] and HCPCS [11]. |
| **Lab Orders/ Values** | Laboratory information such as lab orders and lab results oh the patient. Specified standards for laboratory information are LOINC [12], SNOMED [7] and CPT [10] |
| **Vital Signs** | EHR is an important source of vital sign data. It includes body mass index (BMI), heartbeat rate, blood pressure and body temperature. Most common standard is LOINC [12] to record vital signs. |
| **Reports** | Reports generated by the procedures are stored for future reference. |
| **Utilization** | This is the cost incurred by patients, helps when insurance data is not available. CMS published the reimbursement guideline [4]. |
| **Biosample Data** | Meta-data of biological samples. |
| **Genetic Information** | Genome sequence data is an emerging data type of EHR and widely used for research. |
| **Social Data** | Data such as smoking status or living conditions can help in researching the impact of social variables on health data. |
| **Patient-Generated** | Patient generated data might include several parameters like physical activity, sleep schedules, patient-reported signs and symptoms. |
| **Geo-spatial** | Neighbourhood environment can be used to analyze the influence of surroundings on health. |
| **Surveys** | Medical data extracted from surveys are used to analyze patient-reported symptoms and outcomes of treatments. |
| **Free Text** | Any additional information or notes. |

evaluation, and conveys important information for medical decision making. In comparison with other data types, the progress notes give detailed and personalized information about the patient's history and treatments, presenting a better context of the data [16]. Progress notes, in which the medical reports are primarily composed in normal dialect, have been re-

spected as a capable asset to unravel distinctive medical questions by giving detailed patient conditions, medical reasoning, and medical deduction, which ordinarily cannot be gained by the other data types of EHR [17].

The traditional machine learning models have been applied to predictive analysis in the medical domain for years. In recent years, because of the superior performance of the deep learning models, many have been applied to medical disease predictions. For example, Jin et al. [18] and Maragatham et al. [19] developed a long short-term memory (LSTM) network model to predict heart failure using EHR data. Garske [20] applied a deep convolutional neural network (CNN) to predict diabetes. Wang et al. [21] also developed a CNN approach to detect Colorectal Cancer using diagnoses and medication of the patients in the EHR.

# 3. BACKGROUND

Initially, machines used to interpret the text by identifying the keywords. The advances in machine learning have changed this traditional way into a cognitive task by understanding the meaning and the context of those words. Natural language processing bridges the gap between machines and human language. In this section, the concepts used by NLP to extract information from the free text obtained from EHR systems are discussed.

## 3.1 Named Entity Recognition

Named Entity Recognition (NER) is used to process text and recognize words belonging to certain categories of Named Entities (NE). It is an important tool in NLP used to extract information within the documents. It is easier to retrieve information from data that is labeled through NER compared to raw data. The traditional marked categories are names of people, location, organization, and numerical formats. NER breaks the sentences into a sequence of token to recognize and classify the NE within the text. NER processes the data and detects the NE that is listed in the text. There are two ways to do so, ontology-based and deep learning. In ontology-based NER models specification of named entities depends on the level of detailing of the ontology, like an encyclopedia. Similarly, NER used in the medical field requires a detailed ontology had would have medical terminologies. The requirement of extensive knowledge set for feature engineering to receive good performance is what makes NER challenging. Compared to ontology-based, deep learning NER is more efficient. They are capable of gathering all the words and can also extract words that are unseen in the ontology. With the help of the dense architecture of deep learning models, the network learns to self learn the subject related terminologies. NER identifies the named entities in the document. Notably, the NER annotator combines more than one machine learning algorithms to tag entities with standards to identify numerical entities like time and date formats.

The objective of NER is to identify the symptoms tagged within the sentences extracted from the EHR system. Figure 3.1 shows a few sets of sentences from which complex, rare,

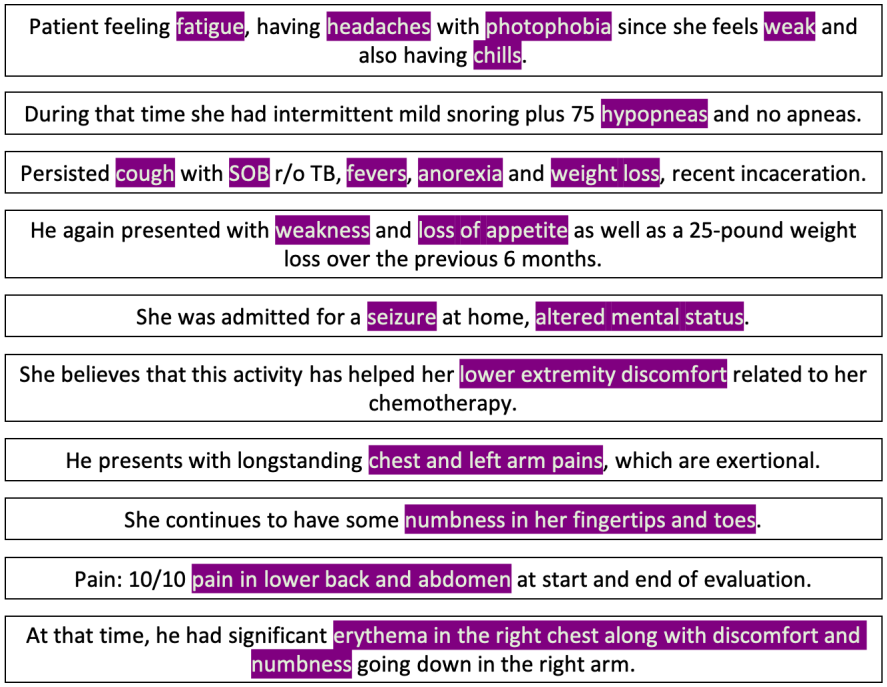



**Figure 3.1.** Example of Named Entity Recognition for Symptom Extraction.

and long phrases of symptoms that need to be tagged as symptoms. This can be achieved through:

1. **Lexicon Approach**: Identifies named entities from the set of a stated ontology. This approach cannot extract new entities that are encountered.

2. **Rule-based Approach**: Identifies named entities based on a set of rules or patterns observed such as phone numbers, SSN, etc.

3. **Machine Learning-Based Approach**: Identifies named entities based on the previous examples seen by the model. This approach requires pre-annotated data samples.

4. **Hybrid Approach**: Combines a machine learning-based approach with a rule-based approach to identify the entities in the text. The machine learning models are trained with annotated data and fine-tune the values to identify new entities.

## 3.2 Conditional Random Field

Conditional models identify decision boundaries for classification by understanding knowledge from perceived data. One of such models is Conditional Random Field(CRF). CRF is implemented for models that require understanding the context of the documents and the neighboring values influence the prediction of the current value. It has previously been used for various purposes such as NER systems, POS tagging, prediction of genes, etc.

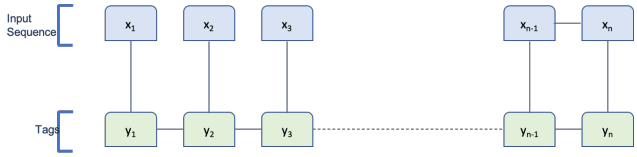



**Figure 3.2.** Conditional Random Field(CRF) Model.

The NER can extract information but it has a problem in detecting the segments. For instance, "shortness of breath" could be extracted as individual symptoms: "shortness", "of" and "breath" One of the ways to solve this problem is by integrating NER with a CRF. It is observed that when CRF is combined with NER, good confidence and efficiency are achieved [22]. Figure 3.2 shows the CRF model of tag sequence $y_1, y_2, y_3, , \ldots, y_n$ in $Y$ of the words in input sequence $x_1, x_2, x_3, \ldots, x_n$ in $X$. By training the model parameters, the CRF model predicts the conditional probability of Y using the equation 3.1. The model calculates the conditional probability through normalization factor $Z(x)$, eigenfunctions specified on transfer feature $t_k$ and state feature $s_1$. The values $\lambda_{\mathrm{k}}$ and $\mu_1$ are the weights assigned to $t_k$ and $s_1$ respectively. If the characteristic condition is satisfied then the transfer feature and state feature values are 1 else it is 0.

$$P(y \mid x) = \frac{1}{Z(x)} \exp\left(\sum_{\mathrm{i,k}} \lambda_{\mathrm{k}} t_{\mathrm{k}}\left(y_{\mathrm{i}-1}, y_{\mathrm{i}}, x, \mathrm{i}\right) + \sum_{\mathrm{i,1}} \mu_1 s_1\left(y_{\mathrm{i}}, x, \mathrm{i}\right)\right) \tag{3.1}$$

## 3.3 Word Embeddings

Word embeddings are a numerical illustration of all words in the text data. This numerical representation could be a binary, integer, or a complex vector that signifies many

characteristics of the word. Word embeddings have gotten to be prevalent as researches prove that word embeddings can successfully probability density of words or phrases, linguistics, and semantics of the words. Mikolov, et al. suggested that word embeddings could transfer the connections of the physical world into the continuous vector space [23].

The recent development in NLP has gained a lot of interest in research of word embeddings by utilizing word vectors where each word is represented by a high-dimensional word vector. These word vectors are dependent on the concurrence of words and phrases in the text document. These concurrences are changed over into a vector representation by applying a likelihood function. These word vectors are obtained by utilizing reasonably simple neural systems with several layers, in an unsupervised way, on large corpora.

Touching the very basic, every system or algorithm at the machine level requires numerical values that it can interpret. Nevertheless, while audio, images, and videos contain high dimensional vectors that contain all information to store, retrieve, or process the files, the text is interpreted as atomic symbols. To bridge the gap between human intelligence to a machine. Processing text data is challenging since the machine cannot interpret the meaning of the text the way humans do. Data mining requires numeric values as an input, therefore translating text from their crude shape to a numeric value is important. Due to this restriction, it is essential to change over the characters in the string to numbers. The effect of word embeddings has made them a gainful introductory step in all sorts of machine learning systems. In a parcel of complex profound neural systems, word embeddings are utilized as inputs rather than crude content. Embedding words have evolved into embedding phrases, sentences, and paragraphs. The need for word embeddings is to achieve all the co-relating words in the vector space in a close cluster. All the similar words are clustered together, this is done through their vectors generated. Figure 3.3 depicts close clusters formed by related words.

Word embeddings have been widely used in NLP. To gain the grammatical significance of words in text analysis, the vector representation of words has been proved advantageous. Embeddings of the words are usually generated considering related words are clustered and group together, thus modeling the local contexts of words. Word embeddings are the numerical representations of the words. A representation for a word that is learned where words

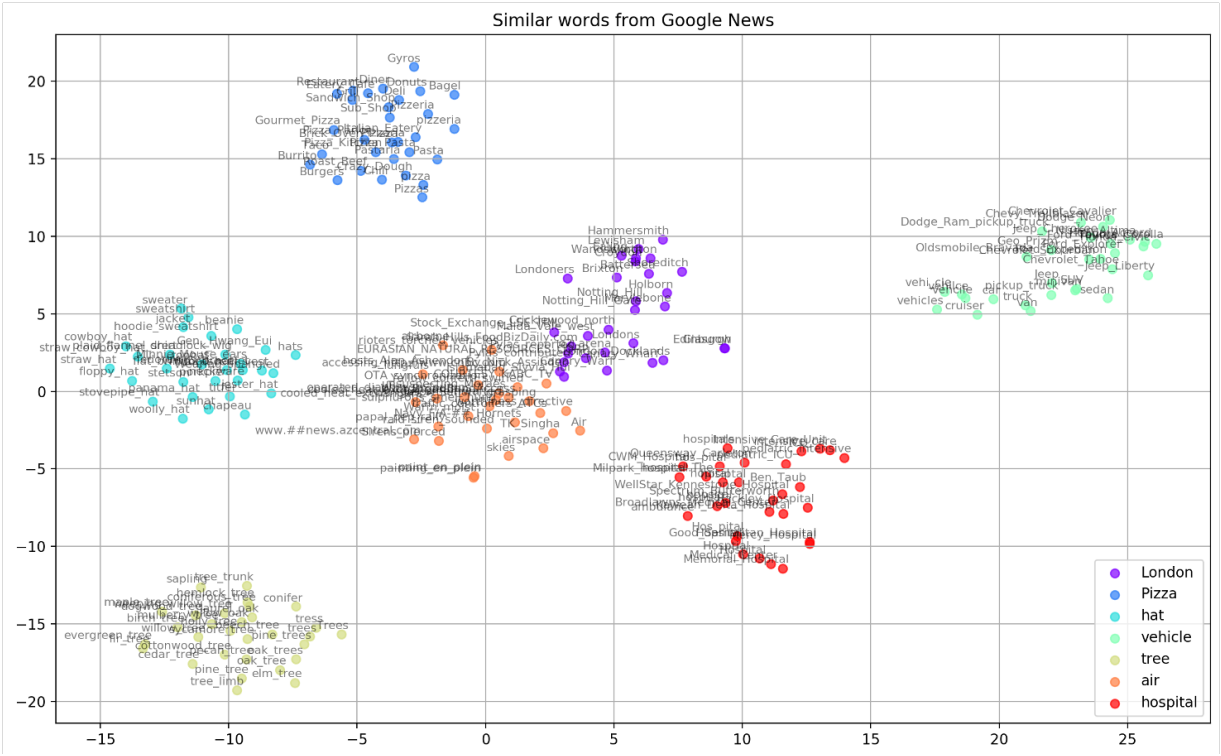



**Figure 3.3.** Example of Close Clusters Generated by Word Embeddings.

that have a similar meaning have a close representation. For a fact, word embeddings are a set of techniques where every unique word in the dataset corresponds to real-valued vectors in a predefined n-dimensional vector space. There could be several ways to represent the same word. If the input sequence $\mathrm{X} = \left\{x_1, x_2, \ldots, x_T\right\}$, where $x_\mathrm{i}$ in X is mapped to a vector $\mathrm{e_i}$. The input sequence X is mapped into an embedding matrix E, which contains vectors for each word in the corpus. This is done with the help of a dictionary, a list consisting of all the unique words that are present in X.

Initially, a simple method was developed to convert text to vectors, known as bag-of-words (BoW). It recorded the frequency count of every word in the text. A registry of words associated with their occurrence count is created. There is no information regarding the sequence or arrangement of words in the text, only the count of times the word has appeared in the text. The pipeline to generate a bag of words is:

1. Collect the data to be processed.

2. Create a vocabulary of a list of unique words by stripping punctuation

3. Count the frequency of each word in the vocabulary that has appeared in the input text

4. Generate document vectors by concatenating the frequency count

James caught cold 3 days ago and suffering from shortness of breath.

| Vocabulary | shortness | today | james | breath | from | good | cold | of | 3 | ago | and | days | caught | suffering |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Vector Generated:

| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

**Figure 3.4.** Example of Bag-Of-Words Embedding.

Later, a variant of BoW was introduced, known as Term Frequency. The difference between the two embeddings is that Term Frequency maintains the sequence of the words in the document. Similarly, a lot of word embedding models were developed to encode the words in the text to numerical form.

James caught cold 3 days ago and suffering from shortness of breath.

| Vocabulary | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | shortness | today | james | breath | from | good | cold | of | 3 | ago | and | days | caught | suffering |
| | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Vector Generated:

| 3,1 | 13,1 | 7,1 | 9,1 | 12,1 | 10,1 | 11,1 | 14,1 | 5,1 | 1,1 | 8,1 | 4,1 |
|---|---|---|---|---|---|---|---|---|---|---|---|

**Figure 3.5.** Example of Term Frequency Embedding.

# 4. LITERATURE REVIEW

Machine learning methods have been extensively used to analyze the EHR records to predict and classify disease states [24][25], model disease progression [26][27], recommend interventions [26][28], and predict future risks [29]. Although the diagnosis can be used as labels for certain diseases in the clinical domain, not all diseases have the corresponding diagnosis code, including chronic cough, and the inconsistent usage of the diagnosis code in the EHR also brings up challenges. Sometimes, annotating and labeling are needed through chart review, but extensive chart reviews of a large amount of clinical data are very costly and time-consuming.

Named entity extraction is a primary subtask of data extraction. NER systems can be based on handcrafted rules or machine learning approaches. The common NER strategies to annotate text are based on rules, word references, machine learning, and deep learning. There are various experiments conducted in numerous fields [30][31]. Relation Extraction is additionally a vital task of data extraction. There are two models to do this, pipeline models and joint models. Pipeline models treat entity extraction and relation extraction as two isolated tasks while joint models see them as a collective task [32]. Classifying further, there are three sorts of strategies of extricating relationships through pipeline models: completely supervised learning methods [33][34], distant supervised learning methods [35] and tree-based methods [36].

In recent years, the distributed representation of words or concepts which is called embedding gained interest in the research areas of text mining, natural language processing, and health informatics [23] [37] [38]. The embedding has been studied for biomedical text classification, clustering [38] [39], and biomedical entity extraction, where a word is a basic unit for the text documents and the word embedding is learned through neural networks. There are various word embeddings made available such as Word2Vec, GloVe, FastText, ELMo, BERT, etc.

Collobert et al. [40], neural network NER frameworks have ended up prevalent due to the minimal feature engineering requirements, which contributes to a higher domain independence [41]. The CharWNN model [42] expanded the work of Collobert et al. [40] by

adding a convolutional layer to extricate character-level highlights from each word. These highlights were concatenated with pre-trained word embeddings and after that utilized to perform sequential classification. It was observed that a simple CNN was not able to solve the long-distance dependency problem. To address this problem, RNN [43], BiLSTM [44], Dilated CNN [45] and BERT [46] were implemented instead on CNN. However, adding a CRF layer enhanced their performance. The LSTM-CRF design [47] has been commonly utilized in NER task [48]. The model consists of two bidirectional LSTM systems that extricate and merge character-level and word-level highlights. A sequential classification is later performed by the CRF layer.

# 5. METHODOLOGY

While dealing with enormous data, the key challenge is not to find the right documents but to extract the important information within the documents. To extract symptoms from a large chunk of clinical notes is a comprehensive process. It is difficult for the machine to interpret clinical terminologies and analyze them. In recent years, applications of deep learning and natural language processing algorithms to the medical data have gained much attention. Researches have been done to make use of clinical notes in the Electronic Health Record (EHR) systems for clinical decision support [49], such as referring to specialist [50], finding similar cases [51] and so on. Typically, the "free-text" clinical notes include discharge summaries, patient instructions, and progress notes, which contain patients' medical history, family history, treatment history, and so on. Managing, classifying the clinical text, and extracting critical information from the clinical text by using learning algorithms are always challenging. Previously, we used concept embeddings to measure the semantic similarities between all extracted symptoms and the seed symptoms to identify additional symptom expressions within the EHR clinical notes. However, the initial definition of the eight symptom clusters is a set of seed words defined by the clinician [52]. To overcome this limitation of human defined seed words, this work is an extension of automating symptom extraction. This section focuses on the technologies used and model architectures that have been created to extract symptoms.

## 5.1 UMLS Metamap

The Unified Medical Language System(UMLS) combines and shares essential vocabulary, classification and coding criteria, and linked resources to encourage the development of efficient and interoperable biomedical data operations and assistance, including electronic health records. The UMLS is a collection of data and software that integrates various health and biomedical terminologies and standards to facilitate interoperability among the health care network. UMLS is a combination of three dominant knowledge sources [53]:

1. **Metathesaurus** - A large biomedical wordbook describing the meaning and their associations of terminologies from RxNorm [6], SNOMED [7], CPT [10], LOINC [12] and ICD-CM [9].

2. **Semantic Network** - Aims to reduce the complexity of Metathesaurus by grouping notions concerning the general topic categories, also known as semantic types, that have been assigned to them.

3. **Specialist Lexicon** - A group of NLP tools to associate a user's language with biomedical resources.

The basic functions of UMLS are:

- Connecting terms and regulations within the medical organization
- Synchronize patient care between departments of a hospital
- Processing textual content to extract concepts, associations, or knowledge
- Ease mapping between vocabularies
- Create an information retrieval system
- Obtain particular terminologies from the Metathesaurus
- Formulate and manage a local terminology
- Generate a vocabulary assistance
- Analyze vocabularies

The textual content in progress notes extracted from the EHR is required to be processed to obtain biomedical concepts. Therefore, we use a UMLS tool called Metamap. The UMLS Metamap is a natural language processing tool that uses various sources to categorize the phrases or terms in the text to different semantic types. Metamap can be used to extract information, classify content, summarize textual data, answer certain questions, mining data, understanding medical notes, indexing based on UMLS concepts, and natural

language processing of biomedical text. Metamap is a highly flexible tool allowing its users to customize their outputs by setting certain flags. Some of these flags were used to get the desired concepts such as:

- **Short Semantic Types** - Displays the abbreviated form of UMLS Semantic Types rather than the original category, e.g., "sosy" instead of Sign or Symptom and "phsf" instead of Physiologic Function
- **Show CUIs** - Displays UMLS identified concept
- **Enable NegEx** - Displays information about negated concepts of UMLS, eg., "no cough" is represented as "N Cough"
- **Use Word-Sense Disambiguation** - In cases where Metamap maps two or more concepts to a recognized entity in content, the WSD Server will endeavor to decide which concept is the most excellent choice for the entity utilizing the setting in which the entity occurs. The WSD Server permits one to utilize either the included disambiguation strategies or ones provided by the client. The word sense disambiguation setting is also used only to consider the best mapped semantic type for each term. This is set to deal with ambiguous content. A phrase may fall into several concepts containing different CUIs. Table 5.1 shows an example of the phrase "cold".

**Table 5.1.** Word-Sense Disambiguation of Metamap.

| Concept | CUI |
|---|---|
| Cold Sensation | C0234192 |
| Cold Temperature | C0009264 |
| Common Cold | C0009443 |
| Cold Therapy | C0010412 |
| Cold brand of chlorpheniramine-phenylpropanolamine | C0719425 |
| Colds homeopathic medication | C1949981 |
| Chronic Obstructive Airway Disease | C0024117 |

Metamap is a readily available tool that uses various sources to categorize the phrases or terms in the text to different semantic types. The tool gives the users an insight into the unified medical language system (UMLS) Metathesaurus from clinical text. Through

its ability to identify the abbreviations of medical terminologies, skimming Metathesaurus concepts in fragments of clinical notes, identifying negation to determine the polarity of the sentences, and word sense disambiguation (WSD) the notes are processed. To classify chronic cough patients, patient-reported symptoms written in the clinical notes are also considered. Figure 5.1 provides an example of clinical notes, and some of the terms, such as "abdominal pain", and "coughing", are mapped into "Sign or Symptom" and "cold" is mapped into "Physiologic Function" by UMLS Metamap.

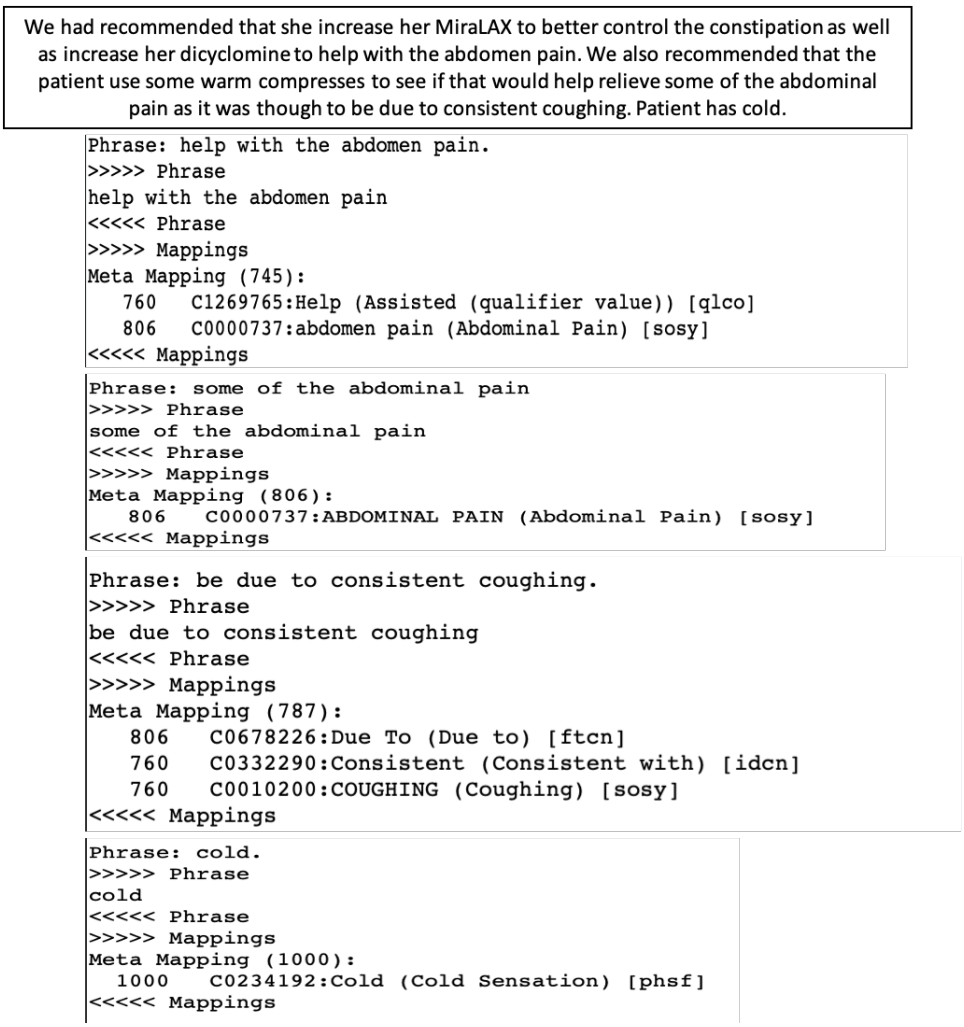



**Figure 5.1.** Example of UMLS Metamap Output.

In this research, the focus is on symptoms of three semantic types - Sign or Symptom, Physiologic Function, and Mental or Behavioral Dysfunction. Figure 5.2 provides an example displaying the mapping of biomedical text to the concepts in UMLS Metathesaurus. In

this example, “poor sleep”, “back pain”, “shooting pain”, “SOB”, “burning”, “abdominal pain”, “anxiety”, “depression“”,“despondency”, “breathing” and “airflow” are mapped as symptoms. The negation detection functionality of the UMLS Metamap is turned on to exclude the negative cases. To maintain some context information, the original text that contains terms that are tagged as either “Sign or Symptom”, “Physiologic Function” or “Mental or Behavioral Dysfunction” are extracted. For this example, the original text “help with the abdominal pain”, “some of the abdominal pain”, “due to consistent coughing” and “cold” are extracted.

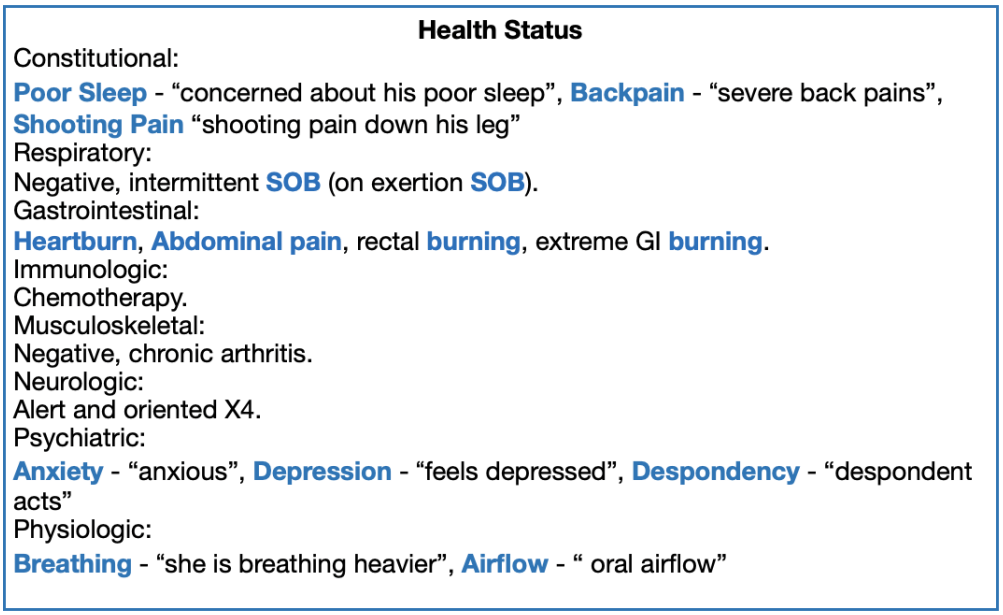



**Figure 5.2.** Symptom Extraction using UMLS Metamap.

## 5.2 Syntactic Dependency Tree with Deep Neural Network

Sometimes individual sentences in large clinical notes are not scanned thoroughly. It is convenient to analyze data when sentences are expressed in terms of words or short phrases that are occurred repeatedly in data. There have been several models proposed ranging from simple bag-of-words to neural networks. The advantage of implementing a neural network is that word embeddings can be fine-tuned into vector representations that closely relate to the context. In our previous work we implemented deep neural network to classify vehicle-

pedestrian encountering risks in natural road environment [54]. For this research, deep neural network is used to train dependency tree correlation and extracting symptoms through NER.

### 5.2.1 Model Architecture

Syntactic parsing is a method by which sentences are tokenized and the part-of-speech tagged sentence is converted into a graph that exhibits the associations among tokenized words administered by syntax standards. A dependency parser is responsible to convert the sentence into a dependency tree. There are several parsers available, for this work, Stanford CoreNLP is used. The text is then broken down into a sequence of tokens followed by the other processes of the Stanford CoreNLP to generate a dependency graph. The dependencies within the sentences are generated along with the POS tagging. The syntactic embeddings are generated for the enhanced dependency graph of Stanford CoreNLP. These embeddings are fed into the feed-forward neural network. In this type of network, there are several fully connected hidden layers between the input and the output layer.Figure 5.3 shows the model architecture.

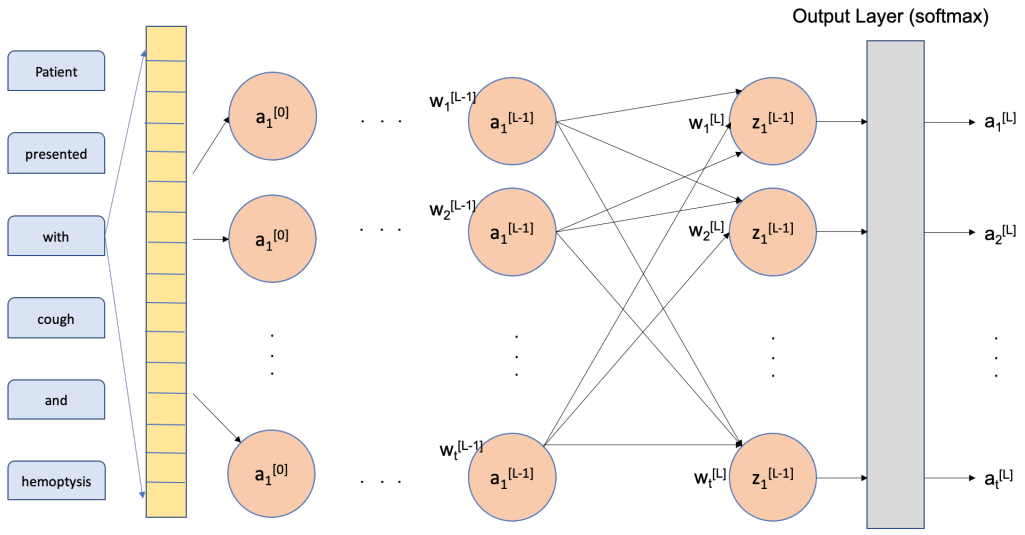



**Figure 5.3.** Stanford CoreNLP with Deep Neural Networks(DNN) Model Architecture.

The sentence is broken into a sequence of tokens to find the dependency between the words. The sentence is analyzed for syntactic evaluation by the dependency parser, Stanford CoreNLP, and a dependency tree with POS tags are generated. The syntactic embedding

converts the dependency tree and POS tags of the words into a vector. The vector is fed as an input to a deep neural network. The tags corresponding to each word is predicted by the network. Figure 5.4 shows the settings of DNN model with the input and output dimensions of each layer.

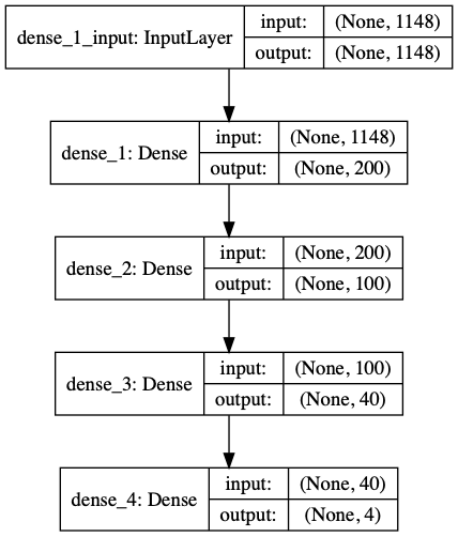



**Figure 5.4.** DNN Model Summary.

### 5.2.2 Model Description

#### Dependency Parser

Dependency trees depict the syntactic relations that exist between elements. It considers the semantics and the knowledge of associations over words. For a sentence, a tree is a directed acyclic graph with nodes representing the words $S\{w_0, w_1, \ldots, w_n\}$ and edges representing the associations $E\{\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_3\}$. Every word in a sentence is associated with another word in some way, the link $\mathrm{e_i}$ connects to words. The first word in the sentence s called as a root node. If the word modifies another word then it is an outgoing link. Likewise, if the word is getting modified by another word then the node has an incoming link as shown in Figure 5.5. A dependency parser is responsible for converting the sentence into a dependency tree.

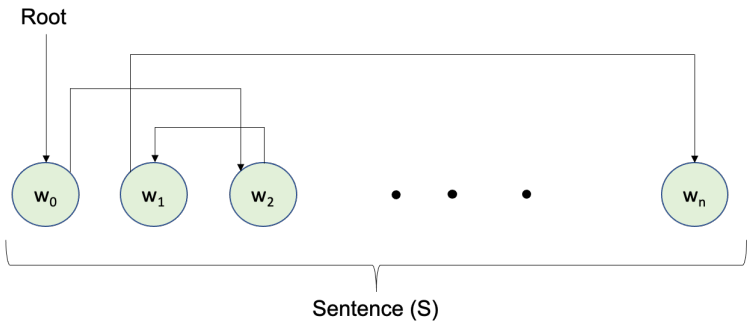



**Figure 5.5.** Dependency Tree.

Stanford CoreNLP is one of such tools supporting six languages including English that allow users to obtain semantic annotations for the content by tokenization, parts of speech, named entities, dependency graphs, and associations. The annotators of Standford CoreNLP are compatible with any character encoding, default is UTF-8 encoding. The parser makes use of Penn Treebank style (PTB) style tokenizer that generates a sequence of tokens corresponding to the words in the text which can handle noisy web data. Tokenization is a process of breaking the words into parts, called tokens. A set of rules are defined to do so. There are many linguistic traditions in different parts of the world. Parts of speech are also known as lexical categories, word classes, tags, or POS, a conventional abbreviation. There are 8 parts of speech commonly known as noun, verb, adjective, preposition, adverb, conjunction, pronoun, and interjection. These are the parent categories that have further subcategories like noun could be a proper noun or common noun. The task of the POS tagger is to determine for every word what it's part of speech is in the context it is being referred to in the running text. Its input is a set of tokens. POS tagger first examines all the possible parts of speech associated with every token. POS tagger then analyzes the two preceding tags and two proceeding tags to conclude what could be the part of the speech of a given word.

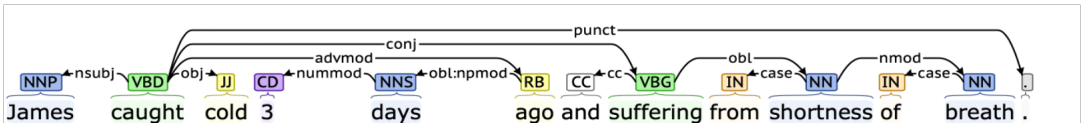



**Figure 5.6.** Example of Dependency Parser in Stanford CoreNLP.

**Table 5.2.** Few POS tags from Penn Treebank used in Stanford CoreNLP.

| Tag Abbreviation | POS Tag |
|---|---|
| NNP | Proper Noun |
| VBD | Verb (Past Tense) |
| JJ | Adjective |
| CD | Cardinal Number |
| NNS | Noun (Plural) |
| RB | Adverb |
| CC | Coordinating Conjunction |
| VBG | Verb (Gerund or Present Tense) |
| IN | Preposition or Subordinating Conjunction |
| NN | Noun (Singular) |

**Table 5.3.** Few Dependencies used in Stanford CoreNLP.

| Link Name | Dependency |
|---|---|
| nsubj | Nominal Subject |
| obj | Object |
| nummod | Numeric Modifier |
| advmod | Adverb Modifier |
| conj | Conjunct |
| punct | Punctuation |
| CC | Coordination |
| npmod | Noun Phrase Modifier |
| nmod | Noun Modifier |
| obl | Oblique Nominal |
| case | Case-Marking |

The POS tags are used to represent the grammatical relationship between the words within the sentence through the dependency tree. It makes it easier for people to understand the syntactical dependency without being a linguistic expert. The description of the POS tags and dependencies in shown in Table 5.2 and Table 5.3 respectively. There are 36 POS tags of Penn Treebank used by Stanford CoreNLP. Table 5.4 is the tabular representation of the dependency tree for the example shown in Figure 5.6.

**Table 5.4.** Tabular Representation of Stanford CoreNLP.

| | pos | in_nsubj | pos_before1 | pos_after1 | pos_before2 | pos_after2 | out_nsubj | out_xcomp | in_ROOT | in_xcomp | in_nummod | in_obl:npmod | in_advmod | in_cc | in_conj:and | in_case | in_obl:from | out_ROOT | in_nmod:of |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| James | NNP | TRUE | | VBD | | JJ | | | | | | | | | | | | | |
| caught | VBD | | NNP | JJ | | CD | TRUE | TRUE | TRUE | | | | | | | | | | |
| cold | JJ | | VBD | CD | NNP | NNS | | | | TRUE | | | | | | | | | |
| 3 | CD | | JJ | NNS | VBD | RB | | | | | TRUE | | | | | | | | |
| days | NNS | | CD | RB | JJ | CC | | | | | | TRUE | | | | | | | |
| ago | RB | | NNS | CC | CD | VBG | | | | | | | TRUE | | | | | | |
| and | CC | | RB | VBG | NNS | IN | | | | | | | | TRUE | | | | | |
| suffering | VBG | | CC | IN | RB | NN | TRUE | | | | | | | | TRUE | | | | |
| from | IN | | VBG | NN | CC | IN | | | | | | | | | | TRUE | | | |
| shortness | NN | | IN | IN | VBG | NN | | | | | | | | | | | TRUE | | |
| of | IN | | NN | NN | IN | | | | | | | | | | | TRUE | | | |
| breath | NN | | IN | | NN | | | | | | | | | | | | | TRUE | TRUE |

**Deep Neural Network**

The Deep Neural Network is a subset of machine learning. Neural networks analyze the human-labeled training set and learn to identify or do certain tasks. It is a collection of densely connected neurons or nodes. A neural network is the interconnection of nodes distributed among layers. Every node in the network acts as a perceptron implementing multiple linear regression. A simple sequential model is said to be a feed-forward network as information flows in uni-direction. A node can receive information from several nodes in the preceding layer and can feed its processed data to several nodes in the succeeding layer. The node designates "weights" to the incoming connections $w_1, w_2, \ldots, w_t$. The product of the incoming data and the weight assigned is calculated by the node. The node then sums all the values and the result is then forwarded to the nodes in the succeeding layer. An activation function that could be nonlinear is applied to the output generated by the nodes. The rectified linear activation function (ReLU) is used at the nodes residing in the hidden layers. If the output is a positive value then it forwards it to the succeeding layer else the output is set to zero. The input layer simply takes in the data $x_1, x_2, \ldots, x_t$ and the output layer generates the results of the softmax activation function. The softmax function normalizes the values through probability distribution where the final output values add up to one. All the layers within the input and output layers are called hidden layers. Initially, all weights are set with random numbers. The input weights are fine-tuned by the hidden layers until the minimum margin of error is obtained. The weights of all the layers including the final layer and the preceding layers are altered through the cost function to minimize the cost of the following prediction. The weights are calculated by:

$$z = \sum_{\mathrm{i}=1}^{t} w_{\mathrm{i}} x_{\mathrm{i}} \tag{5.1}$$

The ReLu activation function g is applied to produce the output that is later forwarded to other neurons.

$$a = g(z) = g(\sum_{\mathrm{i}=1}^{t} w_{\mathrm{i}} x_{\mathrm{i}}) \tag{5.2}$$

$$a = g(z) = max(0, z) \tag{5.3}$$

The softmax function is calculated by taking the ratio of the exponential value of the input parameters to the summation of the exponential of all parameters, shown in Figure 5.3.

$$a_{\mathrm{i}} = \frac{\mathrm{e}^{z_{(\mathrm{i})}}}{\sum_{k=1}^{n[L]} \mathrm{e}^{z_k}} \tag{5.4}$$

### 5.2.3 Syntactic Embedding

The straightforward approach of transforming words to vectors is to designate a one-hot vector in $R|V|$ where $|V|$ signifying the vocabulary size to each word. The vector would set only one value, keeping all other values as zero to represent the respected word. The position where the word resides is set to 1 and all the other positions are marked at 0. One way is to generate syntactic embedding is by creating a dictionary with n-words and each word has an associated index number. The binary representation of the index value becomes the vector representing the word. However, this method would require a large training set to train the model. The syntactic embedding has the similarity issues, 2 similar words being the name of cities like “Paris” and “London” should be recognized. However, their indexes could be far, and no way to identify their closeness. Another issue to be considered is, as the vocabulary size n increases, the word embedding vector size also increases. High dimensional vectors of basic embedding are mostly zeros and some models might not be efficient in processing sparse features with those vectors. For this work, syntactic embedding is generated for the encoding of the Stanford CoreNLP’s dependency graph.

In this case, features are the properties and the relationship of the word with other words in the sentence. With the help of Stanford CoreNLP, parse dependency graphs were generated. The incoming and outgoing links of the word show the association of the word with other words. The dependency parser graph takes into consideration: the POS tag, POS tags of two preceding and two proceeding words, and the type of incoming/outgoing links. Table 5.5 shows POS tagging of one sentence is encoded by syntactic embedding. Similarly, syntactic embedding is applied to encode other information about the dependency parser graph.

**Table 5.5.** Example of Syntactic Embedding of POS Tags in Stanford CoreNLP.

| Words | pos_CC | pos_CD | pos_IN | pos_JJ | pos_NN | pos_NNP | pos_NNS | pos_RB | pos_VBD | pos_VBG |
|---|---|---|---|---|---|---|---|---|---|---|
| John | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| caught | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| cold | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| days | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| ago | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| and | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| suffering | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| from | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| shortness | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| of | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| breath | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

## 5.3 Long Short Term Memory Neural Network with Conditional Random Field

Applications of machine learning have got a lot of attention in recent years. Most of them are done through Recurrent Neural Networks(RNN), particularly Long Short Term Memory(LSTM). RNN is a looping network that connects the previous data collected to perform the current operation. The past information is stored in the memory for a purpose, the information collected and generated in the network is further used in the next steps. The hidden states preserve this information. This enables the network to co-relate information between segments that are separated in the input and this is known as long term dependencies. In RNN, weights are distributed over the input sequence.

### 5.3.1 Model Architecture

The Bi-directional LSTM (BiLSTM) is capable of classifying data but when it is combined with a CRF layer, a strong performance is observed on NER predictions. The previous works of Lample et al.[47] and Peters et al. [55] have shown the effectiveness of CRF when connected with neural networks. A convolution neural network has also been implemented with CRF layer to model character level information extraction and successfully achieved good results in the sequence tagging task of NLP. CRF additionally help the models in tagging decision by analyzing the dependencies of neighboring tags. For this research, a Linear Chain CRF model is added on the top of Bi-LSTM as shown in Figure 5.7 to capture the hard constraints in identifying dependencies in the output tags.

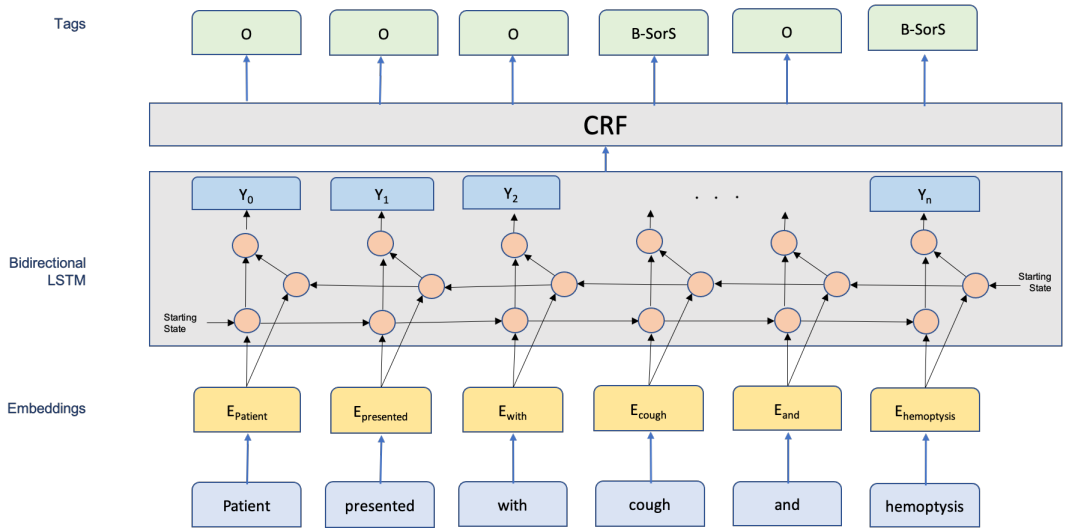



**Figure 5.7.** Bi-directional LSTM (BiLSTM) + CRF Model Architecture.

The sentence is broken into a sequence of tokens. For each token, the BioWord2Vec embedding is generated. The vectors are then fed into the BiLSTM network. The forward pass and reverse pass architecture of BiLSTM fine-tunes the network and the CRF layer extends its functionality of finding dependencies between words and helps in extracting longer phrases of symptoms. Table 5.6 shows the settings of the hyperparameters of BiLSTM model.

**Table 5.6.** Hyperparameters of BiLSTM Model.

| Hyperparameters | |
|---|---|
| Embedding Size | 200 |
| Dropout | 0.5 |
| Epochs | 25 |
| Units in LSTM Cell | 100 |

### 5.3.2 Model Description

LSTM is one of the promising types of RNN. Traditional RNN's performance might decline if the input sequence is long where the internal state remains unchanged. Whereas, LSTM contains an additional gate called as forget gate that manages the dependencies in these long sequences of input and also helps in better interpretations of the meaning.

LSTM efficiently filters the elements from the hidden state that should be passed to the next succeeding cell. LSTM consists of contextual hidden states that comprise of long and short term memory cells. These are used to keep the track of all the previous states rather than just the last preceding input. As per the state of the long and short-term memory cells the network is updated. All the predictions are governed by the network's previous inputs. Since it only knows about the previous information, it is not able to consider or predict future information efficiently. For this reason, bi-directional LSTM is used where both the previous and future information is captured, combined, and stored.

LSTM can handle long-term dependencies problem thus makes it a special type of RNN. Unlike traditional RNN, LSTM contains four gates interacting with each other in different ways. There are three inputs given, $x_t$ is the current input, $c_{t-1}$ is the preceding state, and $h_{t-1}$ is the output of the preceding state. LSTM network highly relies on the state of its cells and the state of the cells is updated with the help of these four gates.

1. Forget Gate - It is necessary to remove the irrelevant data that has been received from the preceding hidden state. Forget gates are responsible for retrieving all the important information from the preceding hidden state and discard the rest of the information from $h_{t-1}$. To obtain a value between 0 and 1, a sigmoid function is applied. The value denotes the amount of information to be retained, the value is multiplied with the previous state, closer the value is to 1, the more the information is retained. This is expressed as following where $b_f$ is the bias vector:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5.5}$$

2. Input Gate - This gate determines which values are to be updated. This includes what amount of information is to be retained from the current input to the current state.

$$\mathrm{i}_t = \sigma(W_\mathrm{i} \cdot [h_{t-1}, x_t] + b_\mathrm{i}) \tag{5.6}$$

3. Input Modulation Gate - Here, a vector is created with new values known as candidate values $\widetilde{C_t}$. These values are later added to the current cell state.

$$\widetilde{C_t} = \tanh(W_n \cdot [h_{t-1}, x_t] + b_n) \tag{5.7}$$

To calculate the current state $c_t$, first the old state is multiplied by the $f_t$. All the irrelevant data is dropped and we add the product of $\mathrm{i}_t$ and $\widetilde{C_t}$.

$$C_t = f_t * C_{t-1} + \mathrm{i}_t * \widetilde{C_t} \tag{5.8}$$

4. Output Gate - After updating the state, need to determine what information is going to be the output. For this, output gate applies tanh function on the cell state to obtain all values between -1 and 1 and then multiplies it with the sigmoid function.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5.9}$$

$$h_t = o_t * \tanh(C_t) \tag{5.10}$$

LSTM solves the vanishing gradient problem. However, the network has access only to the past information and therefore output is computed only on what it posses. To extend its capabilities, bi-directional LSTM (BiLSTM) was introduced. BiLSTM comprises of two hidden networks connected instead of one. It connects two independent LSTM networks to generate an output H. One network traverses the information from the past to the future, known as forward pass ($\overrightarrow{h_x}$) and another network traverses the information from the future to the past, known as reverse pass ($\overleftarrow{h_x}$). We have used an element-wise sum operation to combine the outputs of the forward pass and reverse pass. For every x word in the input sequence, we have computed:

$$h_x = \overrightarrow{h_x} \oplus \overleftarrow{h_x} \tag{5.11}$$

The CRF layer computes:

$$P(y \mid x) = \frac{1}{Z(x)} \exp\left(\sum_{\mathrm{i,k}} \lambda_{\mathrm{k}} t_{\mathrm{k}}\left(y_{\mathrm{i}-1}, y_{\mathrm{i}}, x, \mathrm{i}\right) + \sum_{\mathrm{i,1}} \mu_1 s_1\left(y_{\mathrm{i}}, x, \mathrm{i}\right)\right) \tag{5.12}$$

### 5.3.3 Word2Vec Embedding

Word2Vec was proposed by Mikolov et al to determine and generate vectors of the words. The proposed method calculated vectors of the words by using a simple recurrent neural network consisting of an input layer with a layer to forward preceding execution to the neural system, a hidden layer, and an output layer. Word2Vec is an analytical approach, learns a standalone word embedding from the textual content. Word2vec generates a vocabulary list n, with their vectors $|V|$. The input to the network is a $|V|$ of vocabulary size n produced by the basic embedding or the one-hot encoding. The network is then trained with back-propagation to maximize the log-likelihood function. The output of the network is this likelihood function of all words in the vocabulary of being the next plausible word. The proposed solution was able to obtain linguistic regularities and gained credibility in the field of research. Two distinct architectures of Word2Vec were introduced by Mikolov et al. [23]. Word2Vec compares each word with its neighboring words in the corpus to predict the context of the words through skip-gram. Another technique is to understand the context so that the network can predict the target word this approach is called a continuous bag of words (CBOW). These are similar techniques yet different, one process is an inversion of another. However, skip-gram links the neighboring words with the target word, considering each as a different observation which benefits in large data sets.

To capture the semantic associations between the words or concepts through word embeddings a variant of Word2Vec was used, known as BioWord2Vec. The BioWord2Vec [56] includes pre-trained biomedical word embeddings [57] [58] using PubMed and the clinical notes from MIMIC-III Clinical Database [59]. The fastText was applied to compute 200-dimensional word embeddings. Given a symptom term consisted of more than one word, it computes the symptom embedding by computing the element-wise sum of the representations of each word embedding. The semantic similarities between the symptoms can be then calculated by measuring cosine similarity between the embeddings. Figure 5.9 shows

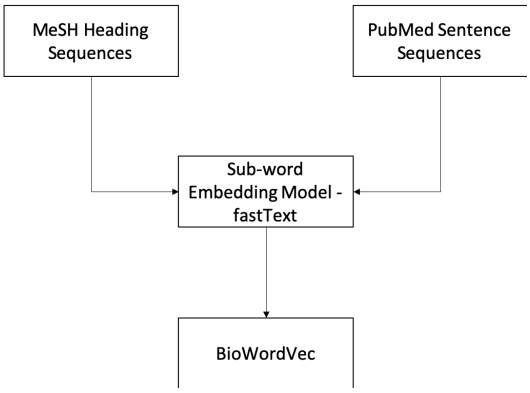



**Figure 5.8.** BioWord2Vec Embedding.

a few of the symptoms through the heatmap of the symptom cosine similarity matrix using embeddings generated from the BioWord2Vec. The higher the similarity score is that is the lighter the cell is, the more similar the symptoms are from the semantic point of view. Based on similarities in Figure 5.9, closely related symptoms show high similarities. For example, "nausea" and "vomiting" are closely related terms, and cosine similarity (0.9) between them is high.

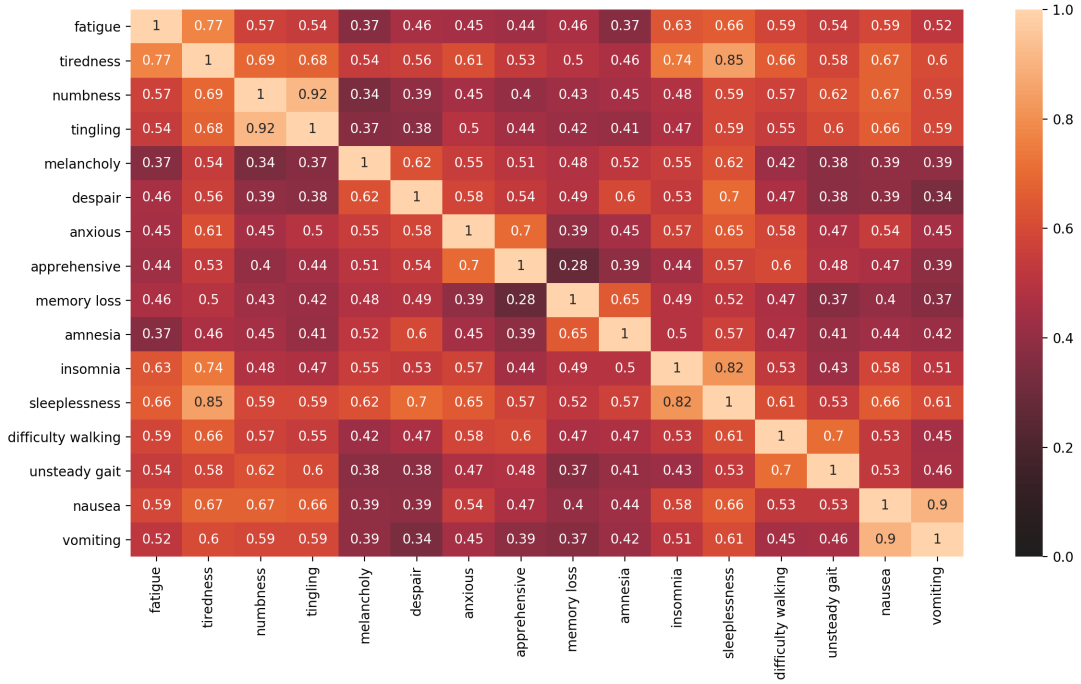



**Figure 5.9.** Heatmap based on the Cosine Similarity Matrix using BioWord2Vec embeddings.

## 5.4 Bidirectional Encoder Representations from Transformers with Conditional Random Field

### 5.4.1 Model Architecture

Bidirectional Encoder Representations from Transformers (BERT) is a sequence classifier that considers every sequence one at a time and makes a local decision. It takes into consideration the adjacent data before making a decision but does not examine the output sequence to analyze the neighboring values. While CRF takes into account the output sequence to maximize the probability and models the dependency of adjacent output tags. In recent years, various models are developed to improve the NER sequence tagging. However, very limited studies have been done on combining BERT with the CRF model to do the same. Sauza et al. [60] implemented Portuguese BERT with CRF to tag ten named entities. This research demonstrates BERT with CRF model, as shown in Figure 5.10 can be used to analyze and extract information from the clinical documents. The sentence is broken into a sequence of tokens. BERT examines the context of the sentence and assigns an embedding to each token. For this research, pre-trained BERT and pre-trained BioBERT is used. BioBERT is trained on medical corpus thus can recognize the medical terminologies within the text. The CRF layer extends its functionality of finding dependencies between words.

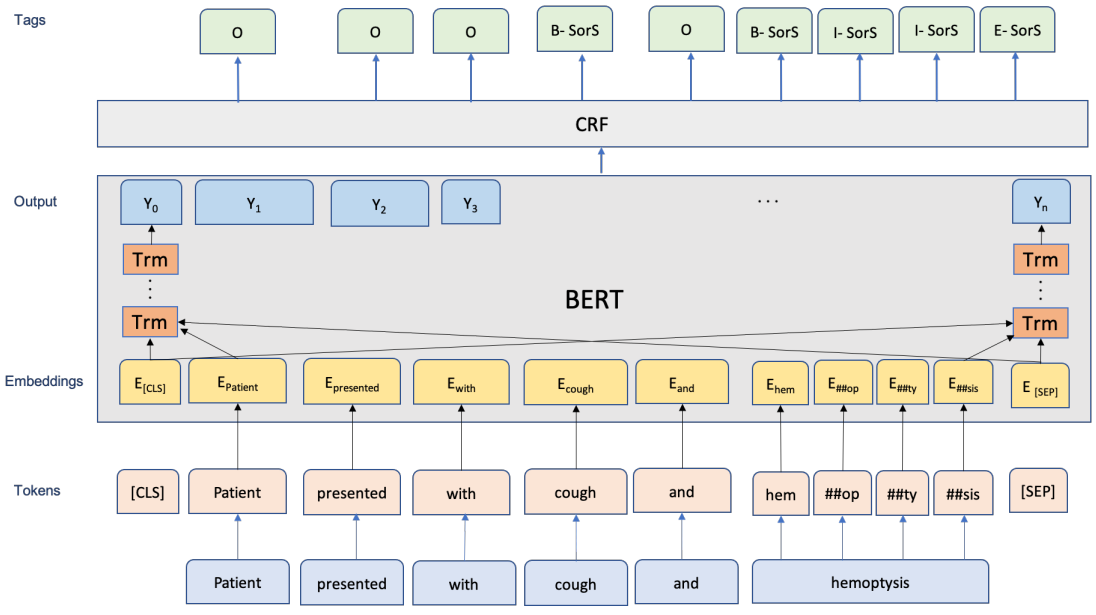



**Figure 5.10.** BERT + CRF Model Architecture.

### 5.4.2 Model Description

BERT is a language model which is different from other language models as it combines both features based and fine-tuned approach of language model.[61] BERT is the first fine-tuning based representation model that achieves state-of-the-art performance on a large suite of sentence-level and token-level tasks. Initially, sentences are tokenized by BERT tokenizer and each word is embedded by WordPiece embeddings. The masked model masks some percentage of tokens to predict by BERT Language Model which is a multi-layer bidirectional Transformer encoder. The last layer of the model contains tokens embeddings. A pre-trained BERT model is available, which is trained by google BERT and available on TensorFlow hub [62]. The model supports a maximum of 512 lengths of tokens for one sequence.

There have been feature-based and fine-tuning based approaches to practice pre-trained models. However, they consider the unidirectional strategy, that acts as a bottleneck for implementing different types of architectures while pre-training to study common language representations. BERT achieves understanding the context of the given text through a bidirectional masked language model (MLM) [63]. BERT combines the bi-directional transformer, used in MLM to foretell the vocabulary index of the randomly masked token words with the "next sentence prediction" task. BERT has also been proved to outperform many token-level as wells as sentence-level tasks.

The CRF layer computes:

$$P(y \mid x) = \frac{1}{Z(x)} \exp\left(\sum_{\mathrm{i,k}} \lambda_{\mathrm{k}} t_{\mathrm{k}}\left(y_{\mathrm{i}-1}, y_{\mathrm{i}}, x, \mathrm{i}\right) + \sum_{\mathrm{i,1}} \mu_1 s_1\left(y_{\mathrm{i}}, x, \mathrm{i}\right)\right) \tag{5.13}$$

There are two procedures implemented in BERT architecture, pre-training and fine-tuning. BERT is trained on a large plain text corpus from BooksCorpus [64], English Wikipedia, and Billion Word Benchmark [65] that makes it an unsupervised model used for downstream tasks of NLP. It deeply trains the bidirectional model by masking 15% of the tokens and predicting those tokens. It also generates a boolean value to know if two consecutive sentences are linked or independent of each other. The model comprises of twelve to twenty-four layer transformers. The model is initially set to pre-trained parameters and later updated by labeled data from downstream tasks while fine-tuning. During this process,

special tokens [CLS], to define the start of the sequence of tokens and [SEP], a separator to define the end of the sequence of tokens are added as shown in Figure 5.11.

| Sentence | Patient | presented | with | cough | and | hemoptysis | | | |
|---|---|---|---|---|---|---|---|---|---|
| Tokens | [CLS] | Patient | presented | with | cough | and | hem | ##op | ##ty | ##sis | [SEP] |

**Figure 5.11.** BERT Tokenization.

### 5.4.3 BERT Embedding

Unlike, traditional word embeddings that represent the words in vectors, dynamic word embeddings also known as language models consider the possible meanings of words such as "back pain" and "go back". The vectors in dynamic word embeddings overcome the limitations of the traditional word embeddings by understanding the context of the words. Elmo, one of the first dynamic embeddings uses a bidirectional LSTM network to analyze the context of the words in the sentence and then designates vectors to represent them. In 2018, Google introduced Bidirectional Encoder Representations from Transformer (BERT) that could outperform state-of-art models in NLP applications. It makes use of the attention mechanism of a transformer to carry forward an entire sequence of values from one layer to another instead of a sequential transfer. Combining the context embedding feature of Elmo with bidirectional transformers led to a successful dynamic word embedding model, BERT.

BERT attempts to capture the semantics and then generates dependent embeddings. So the word, "cough" does not have a specific embedding associated with it. The embedding of the word changes as per the context it is used in. All sentences in BERT begin with a [CLS] tag and end with a delimiter [SEP] tag. BERT enables the user to train and classify documents as per the user's data, enables users to use BERT embedding in user-defined models, and also has a pre-trained model to perform transfer learning. For this, BERT pre-trained model is used for embedding and classifying documents.

Figure 5.12 and Figure 5.13 shows the projection of BERT embeddings on a 2D plane, (1,0) denotes the beginning and (0,1) denotes the end of the sentence. Figure 5.12 displays how the embeddings of the words using BERT changes when tense changes and Figure 5.13

displays the change in embeddings when the punctuation change. The punctuation at the end of the statement changes the context of the statement, thus the embeddings also change.

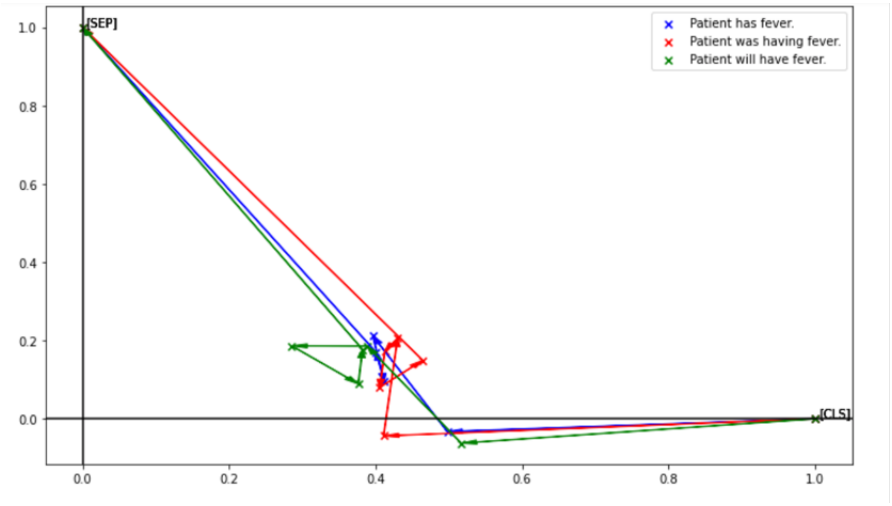



**Figure 5.12.** Example showing BERT considers the Tense used within the Sentence.

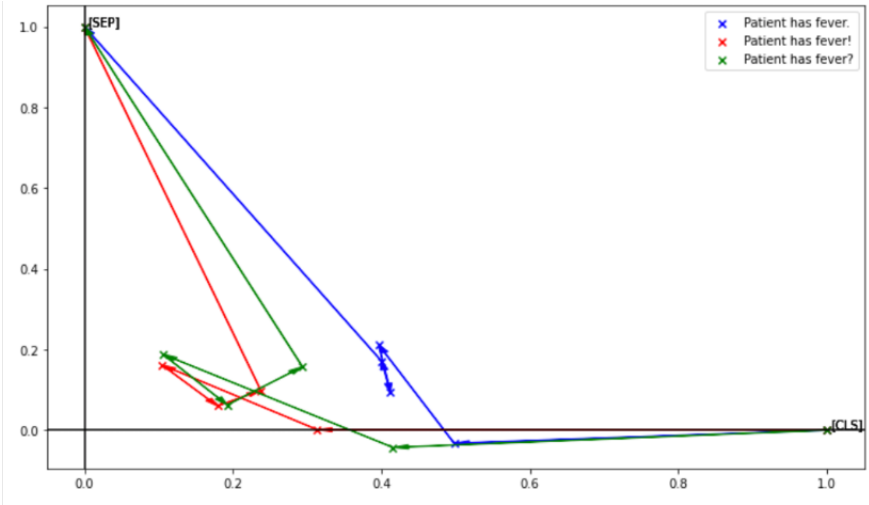



**Figure 5.13.** Example showing BERT considers the Punctuations used within the Sentence.

# 6. DATA COLLECTION

## 6.1 Medical Dataset

In recent years, because of the wide adoption of electronic medical data systems, vibrant health-related data stored in the Electronic Health Records (EHR) systems are available to use for predictive analysis. Many computational models have been developed based on these data for disease prediction, hospital readmission prediction, or mortality prediction. Research has been done on various respiratory disease prediction and analysis using different learning models, although most of the current research focused on the chronic obstructive pulmonary disease (COPD) and cancer datasets.

### 6.1.1 Chronic Cough

Chronic cough, or cough lasting more than eight weeks, affects approximately 10% of adults and is a common outpatient complaint. Affected individuals can cough hundreds or even thousands of times per day [66], severely impairing their quality of life [67]. The underlying reason for cough in an individual is often multifactorial [66], with coughing persisting in some cases for years [66]. Chronic cough is often treated according to one or a combination of the common causes. Since more than one underlying condition may cause chronic cough, many individuals with chronic cough do not respond to treatment [68], highlighting the need to identify such individuals for both prospective and retrospective study. Weiner et al. [69] developed a rule-based algorithm to identify chronic cough. The sensitivity gain by the rule-based algorithm is high based on a validation of a small set, however, the specificity is unknown. Unlike most other diseases, there is no ICD diagnosis code for chronic cough, which makes it even difficult to identify and analyze the population with this chronic disease. For this research, on a random basis, 570 out of 2654 patient records were selected that had clinical notes about their illness.

### 6.1.2 Breast Cancer and Colorectal Cancer

Cancer patients commonly experience symptoms such as pain, depression, and fatigue as a consequence of undergoing chemotherapy treatment, and these symptoms may persist, or develop, even after the chemotherapy ends. These symptoms add to the patient's distress and functional impairment if left untreated. The literature shows individual differences that have associations with the symptoms and patient's experience [70][71]. The symptoms could be gastrointestinal symptoms including nausea, vomiting, lack of appetite, or psychoneurological symptoms including depressive symptoms, anxiety, or other types. The study cohort consists of patients with a primary diagnosis of breast cancer (BC) or colorectal cancer (CRC) who have electronic medical records in the EHR system. BC and CRC patients are identified using the International Classification of Diseases (ICD). Through these ICD codes, BC and CRC cases were identified that have received chemotherapy within the ten years of 2007-2017. For this research, on a random basis, 570 out of 3458 patient records from the BC dataset and 570 out of 1694 patient records from the CRC dataset were selected that had clinical notes about their illness.

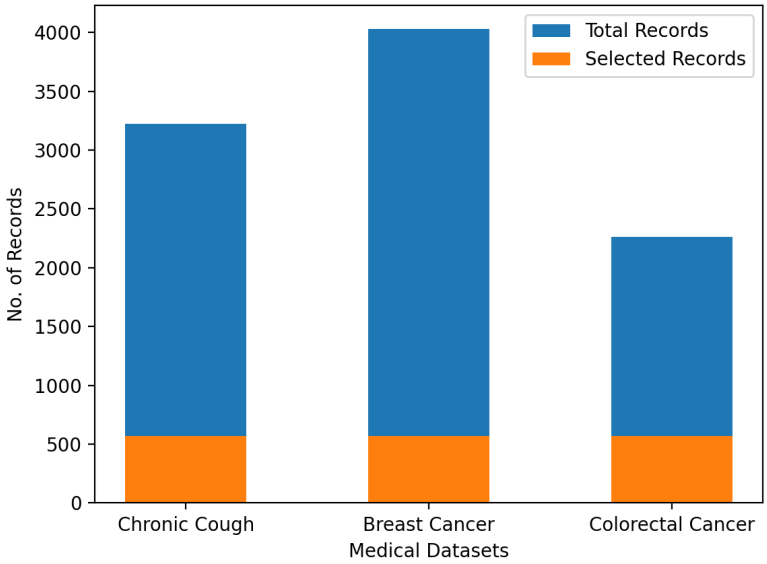



**Figure 6.1.** Medical Dataset Count.

## 6.2 Materials

In this research, the motive is to prove that IE is a powerful tool for extracting symptoms of known and unknown diseases. IE can be used to extract medications, treatments, medical reasoning, and many more in the medical field. The goal of this work is to extract symptoms from unstructured text. The dataset comprises of positive as well as negative statements. Positive statements include, "James caught cold 3 days ago and suffering from shortness of breath" while the negative statements include "patient had no chest pain or cough." All the negative statements in the dataset were not tagged by the clinicians as symptoms and were fed into different models to evaluate if the models can recognize and eliminate negative statements.

## 6.3 Annotation

The dataset used to train the models was manually tagged by clinicians. Annotating every recording involved recognizing symptoms. Since it was not feasible to manually annotate entire clinical notes, the notes were processed and only the recordings within the patient illness section were retrieved. Taking into consideration the limitation of BERT tokens of 500, the records with word count lesser than 500 were selected. Any symptom that was discontiguous in time was not labeled as a symptom in the dataset. For example, "Two years ago, the patient was suffering from shortness of breath which was cured. The patient is now showing symptoms of SOB", in this recording, "shortness of breath" was not tagged as a symptom. The recordings were tagged and then fed into different models. Figure 6.2 shows the distribution of human-annotated symptoms with respect to the length of the symptom phrases.

For this research, $BIOE$ chunk tagging a variant of inside-outside-beginning ($IOB$) tagging [72] for NER was used. The $B$ tag is used to show the beginning of the chunk and the $E$ tag is used to show the end of the chunk. Anything that is between the chunk delimiters is set to $I$ tag. If the word token does not belong to any chunk then it is indicated by an $O$ tag. Any single chunk is represented by a $B$ tag. Figure 6.3 shows a set of records where $BIOE$ tagging is used.

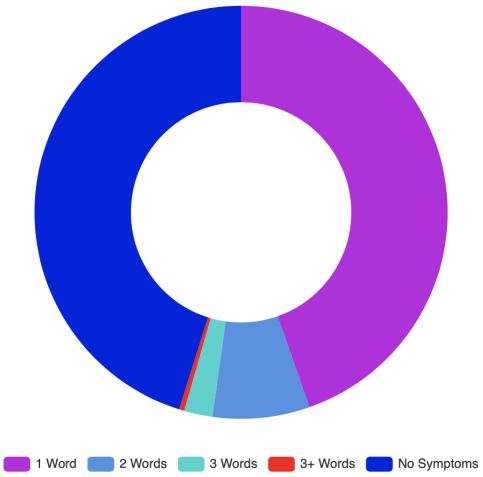



**Figure 6.2.** Distrbution of Human Labeled Dataset with respect to Symptom Phrase Length.

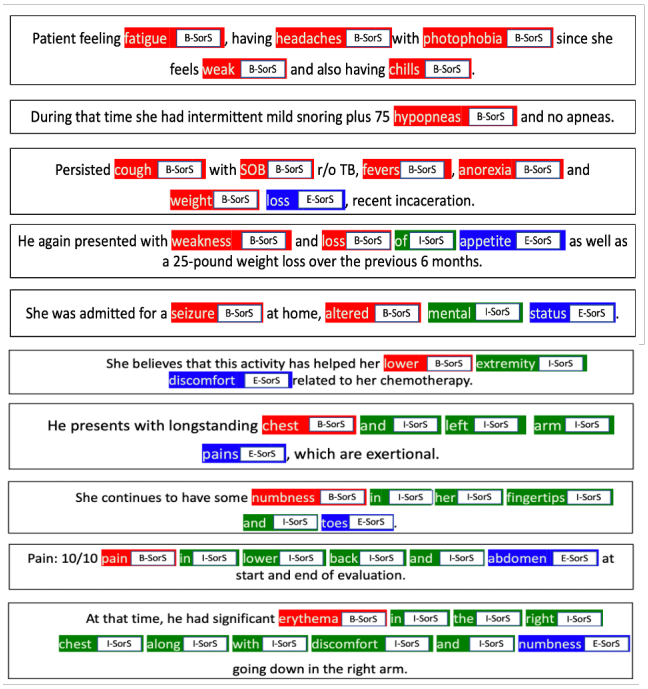



**Figure 6.3.** Named Entity Recognition BIOE Tagging.

## 6.4 Social Media Networks

Twitter is a leading application to share information. Twitter enables its user to microblog and shares its content online. People in the community interact with others through short messages called tweets. Through tweets, people share emergency alerts, breaking news, and research developments. In recent years, the social web has been increasingly used for health information seeking, sharing, and subsequent health-related research. The use of social media as an information-seeking tool increased significantly. Social media has become a popular tool that enables users' creation and exchange of information. Social media allows users to form groups or online communities to provide information and emotional support to peers. In recent years, the social web has been increasingly used for health information seeking, sharing, and subsequent health-related research [73].

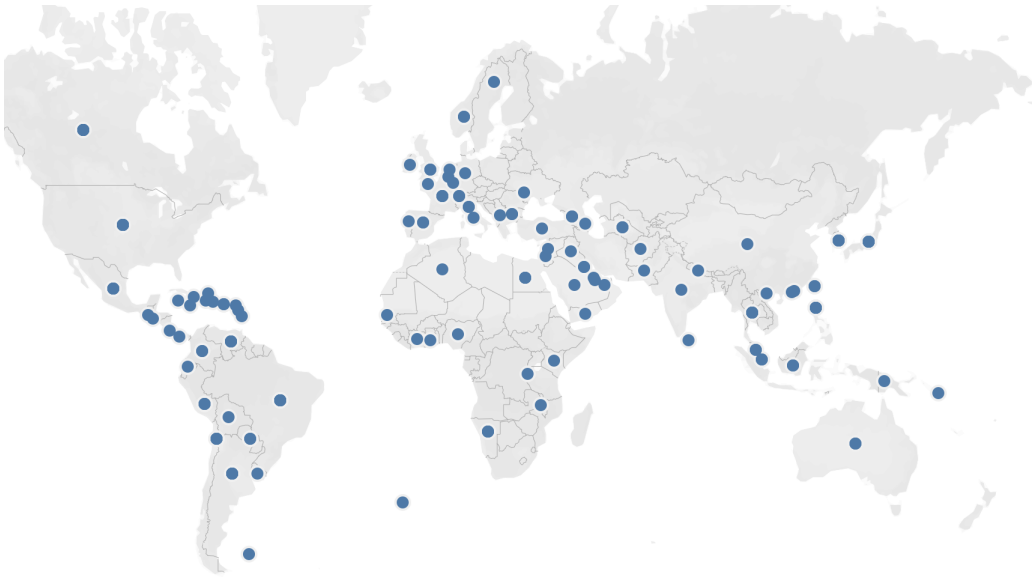


**Figure 6.4.** Twitter User Base.

Twitter was developed in 2006 as an interactive platform for users to communicate and now used by researchers to communicate with people of similar interests and mine information. In 2017, a survey suggested that a total of 5 billion tweets were used by 137 health research projects. More than half of those research projects were based on examining the content of the tweets. WHO has stated evaluating and monitoring of the health of people

in danger to recognize health problems as one of the functions of public health [74]. For a long time, researchers made use of the available datasets about the health conditions. With an increase in the use of social platforms, they have started leveraging social mediums like Twitter, Instagram, and Facebook to extract data about the community's health and concerns [75][76]. Out of all the other social networks, Twitter acts as one of the major sources of data for the researchers as it avails data to its massive base of users. On average, 350000 tweets are tweeted every minute on Twitter, which corresponds to 500 million tweets per day [77]. Figure 6.4 shows live data users around the world using Twitter. However, there are few countries where Twitter is banned.

### 6.4.1 COVID-19 Dataset

In December 2019, an illness called COVID-19 caused by SARS-CoV-2, a strain of coronavirus, that led to a dreadful global outbreak [78]. It spreads through droplet or person to person contact transmission. An individual could have an asymptomatic or symptomatic illness. An individual could have no or mild symptoms or in an extreme case could have a severe illness. An infected person can come in contact with several people and transmit the virus. The patient density makes it challenging to manage the illness. For this reason, it is very important to diagnose patients suffering from this illness.

Twitter informational collection about COVID-19-related online discussions can be used to research about the pandemic across the world. Individuals have been using the public platform, Twitter to share their views and opinions. Twitter's open application programming interface (API), has demonstrated to be an important asset for considering a wide run of subjects. For a long time now, Twitter has been used to the dynamics perceptible on the internet be it distribution of information [79][80] or the influence of bots [81]. A study suggested that health researchers were able to convey messages about Ebola and H1N1 across a wider audience through Twitter [82][83]. During the pandemic outbreak of COVID-19, Twitter has been a useful resource for the researchers as well as people to understand the global health crisis [84][85][86].

Twitter tweets appear in the language written by the author. Recently, Twitter tested automatic language translation of the tweets in the language used by the user account.

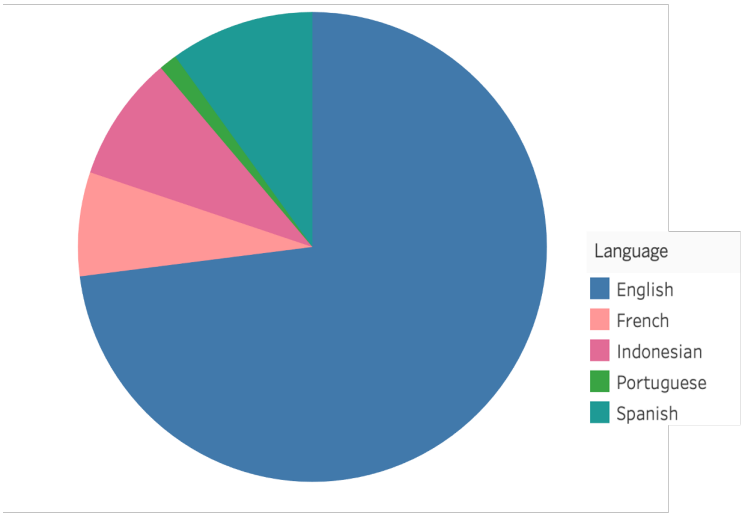



**Figure 6.5.** COVID-19 Tweets in Top 5 Languages.

However looking at the statistics, it was observed that the majority of the COVID-19 related tweets and the retweets were in English. There are 34 languages supported by Twitter but covid related tweets are dominated by English. Figure 6.5 shows the dominance of covid related tweets in English on Twitter. Figure 6.4 and Figure 6.5 are generated using live Twitter data accessed through Tableau's web data connector. COVID-19 related tweets from the date 14 March 2020 to 30 April 2020 in English were extracted through the Twitter's developer account. CDC announced COVID-19 as a global pandemic on 11 March 2020. Thus tweets from the later dates were retrieved to extract symptoms of the patients suffering from COVID-19 across the world. Figure 6.6 shows the total number of tweets extracted for each day. There was a spike in the number of tweets about COVID-19 in April. Countries across the world were declaring nationwide lockdowns, and people were panicking about the virus. People were also taking more precautionary measures and shared their concerns during this time.

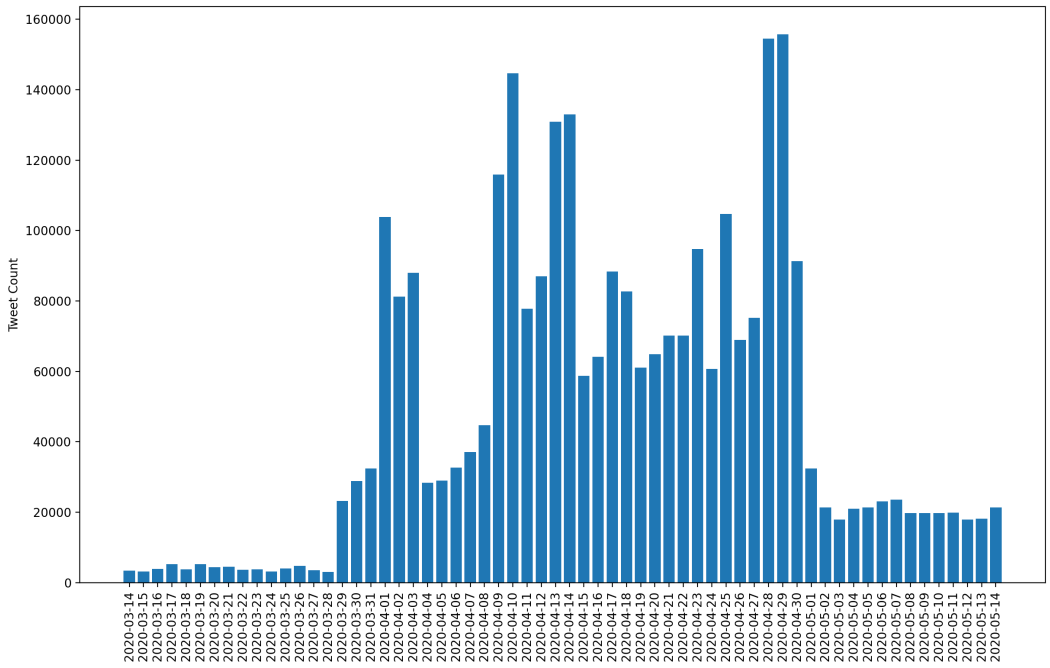



**Figure 6.6.** COVID-19 Tweet Count.

# 7. EVALUATION

To evaluate a NER model evaluation metrics are different from standardized ML models. Typically to model such systems precision, recall, and F1 score are calculated at the token level. For analyzing the performance of the models, various measures can be calculated. In this section, these evaluation metrics are described and the results of those metrics on the models are shown. Every sentence in the NER model is converted into a sequence of tokens. And each token in this sequence has a predicted tag that is compared to the actual tag. The possible outcome of the matches are :

1. Model matches string and entity
2. Model hypothesized an entity
3. Model drops an entity
4. Model tags the boundaries of the string incorrectly

**Table 7.1.** Model matches string and entity.

| **Golden Standard** | | **Model Prediction** | |
|---|---|---|---|
| **String** | **Entity Tag** | **String** | **Entity Tag** |
| Patient | O | Patient | O |
| is | O | is | O |
| suffering | O | suffering | O |
| from | O | from | O |
| cough | B-SorS | cough | B-SorS |
| and | O | and | O |
| shortness | B-SorS | shortness | B-SorS |
| of | I-SorS | of | I-SorS |
| breath | E-SorS | breath | E-SorS |

## 7.1 Evaluation Metrics

It is not enough to evaluate the models in NLP through one metric. To get insights into how well the model works when the testing dataset is fed into the network, various measures

**Table 7.2.** Model hypothesized an entity.

| **Golden Standard** | | **Model Prediction** | |
|---|---|---|---|
| **String** | **Entity Tag** | **String** | **Entity Tag** |
| Patient | O | Patient | O |
| is | O | is | O |
| suffering | O | suffering | B-SorS |
| from | O | from | O |
| cough | B-SorS | cough | B-SorS |
| and | O | and | O |
| shortness | B-SorS | shortness | B-SorS |
| of | I-SorS | of | I-SorS |
| breath | E-SorS | breath | E-SorS |

**Table 7.3.** Model drops an entity.

| **Golden Standard** | | **Model Prediction** | |
|---|---|---|---|
| **String** | **Entity Tag** | **String** | **Entity Tag** |
| Patient | O | Patient | O |
| is | O | is | O |
| suffering | O | suffering | O |
| from | O | from | O |
| cough | B-SorS | cough | O |
| and | O | and | O |
| shortness | B-SorS | shortness | B-SorS |
| of | I-SorS | of | I-SorS |
| breath | E-SorS | breath | E-SorS |

**Table 7.4.** Model tags the boundaries of the string incorrectly.

| **Golden Standard** | | **Model Prediction** | |
|---|---|---|---|
| **String** | **Entity Tag** | **String** | **Entity Tag** |
| Patient | O | Patient | O |
| is | O | is | O |
| suffering | O | suffering | O |
| from | O | from | O |
| cough | B-SorS | cough | B-SorS |
| and | O | and | I-SorS |
| shortness | B-SorS | shortness | I-SorS |
| of | I-SorS | of | I-SorS |
| breath | E-SorS | breath | E-SorS |

are used. The model's performance is not based on the only training set but also measured when unseen data is fed into the network. All the metrics used to evaluate the models are based on four values:

1. **True positives** are those cases where the model correctly predicts labeled symptoms as a symptom.

2. **True negatives** are those cases where the model correctly predicts labeled *O* as no symptom.

3. **False positives** are those cases where the model incorrectly predicts the *O* tag as a symptom.

4. **False negatives** are those cases in the dataset where the model incorrectly predicts symptom tag as the *O* tag.

**Table 7.5.** Confusion Matrix.

| | **Golden Standard** | | |
|---|---|---|---|
| **Model Prediction** | | **Symptom** | **No Symptom** |
| | **Symptom** | True Positive | False Positive |
| | **No Symptom** | False Negative | True Negative |

In the dataset, more words are tagged as *O* as compared to symptoms. The sentences might or might not contain symptoms. Therefore, to evaluate the reliability of the models more metrics are calculated from the true positive, true negative, false positive, and false negative cases of the models. Precision shows the reliability of the model by calculating the ratio of the positive predicted values to the total of positive cases. While recall calculates the ratio of the number of positive values to the total positives cases of the gold standard. The F1 score calculates the weighted average of precision and recall by considering both false-positive and false-negative cases.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{7.1}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{7.2}$$

$$F1\ Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{7.3}$$

At times, precision and recall value might have a significant difference. If there is a tradeoff between the two metrics then the priority should be given to the metric based on the context. For instance, in this research a high cost is associated with false negatives, hence Recall should be prioritized over precision. If the patient observes symptoms and the model incorrectly predicts it as a no symptom then the doctor might not attend or miss the possibility to examine the patient for the respected disease. For the wrong diagnosis, a high price would have to be paid by the patient. Therefore, the objective is to minimize false negatives.

Message Understanding Conference (MUC) introduced scoring categories to calculate precision, recall, and F1 score. Table 7.6 shows the categories and their description. To cope with the different possible outcomes on the matches, the MUC scoring category is used. It takes into consideration all cases where the model matches string and entity. the model hypothesizes an entity, model drops an entity, and if boundary tags do not match. With the help of these values, better insights into the models can be obtained.

**Table 7.6.** Message Understanding Conference (MUC) scoring categories.

| **Scoring Category** | **Definition** | **Representation** | **Example** |
|---|---|---|---|
| Correct(COR) | Golden Standard and Model Prediction match. | $response = key$ | $cough = cough$ |
| Incorrect (INC) | Golden Standard and Model Prediction do not match. | $response \neq key$ | suffering $\neq$ cough |
| Missing (MIS) | Golden Standard string not tagged by model. | *response is blank and key has a value* | response=“”; key=nausea |
| Spurious (SPU) | Golden Standard does not contain the string tagged by the model. | *response has a value and key is blank* | response=vomiting; key=“” |

## 7.2 Evaluation Schemes

### 7.2.1 Exact Match Evaluation

CoNLL-2003 introduced the exact match technique to evaluate NER models. In this evaluation scheme, only if the model prediction is an exact match with the golden standard it is considered to be correct. This scheme could be used to analyze how well the model performs while extracting complete phrases from the dataset. To calculate precision and recall, MUC readings were used.

$$Precision = \frac{COR}{COR + INC + SPU} \tag{7.4}$$

$$Recall = \frac{COR}{COR + INC + MIS} \tag{7.5}$$

**Table 7.7.** Example of Exact Match.

| Golden Standard | Model Prediction | Precision | Recall | F1 Score |
|---|---|---|---|---|
| cough | cough | 1.0 | 1.0 | 1.0 |
| nausea, vomiting | nausea | 1.0 | 0.5 | 0.67 |
| pain | fever, pain | 0.5 | 1.0 | 0.67 |
| joint pain | pain | 0.0 | 0.0 | 0.0 |
| back pain | lower back pain | 0.0 | 0.0 | 0.0 |

It was later noticed that a lot of relevant data could be lost by this technique, specifically in the medical domain. For instance, the golden standard contains "difficulty in walking" but the model extracts "difficulty walking" then it would be considered as incorrect, or if the golden standard contains "cough" as a symptom but the model extracts a phrase "suffering from cough" as a symptom, it would be considered as a mismatch. There could be several cases like these. This would make the evaluation metric unreliable.

### 7.2.2 Relaxed Match Evaluation

To overcome the problem observed in an exact match of elimination of partially matched symptoms, a relaxed match technique was calculated. In this scheme, the golden standard

and model prediction extracted phrases of symptoms are converted into tokens. Every token is then checked for its exact match to calculate precision, recall, and f1 score.

**Table 7.8.** Example of Relaxed Match.

| Golden Standard | Model Prediction | Precision | Recall | F1 Score |
|---|---|---|---|---|
| cough | cough | 1.0 | 1.0 | 1.0 |
| nausea, vomiting | nausea | 1.0 | 0.5 | 0.67 |
| pain | fever, pain | 0.5 | 1.0 | 0.67 |
| joint, pain | pain | 1.0 | 0.5 | 0.67 |
| back, pain | lower, back, pain | 0.67 | 1.0 | 0.8 |

### 7.2.3 N-Gram Evaluation

This n-gram evaluation scheme, a variant of the BLEU metric that is commonly used to evaluate sentence extraction [87]. Instead of sentence extraction, phrase extraction is evaluated. This scheme gives the evaluation based on the length of the symptoms extracted. Generally, the longer the phrase, the difficult it is for the model to extract it. To get insights on how well the model works, this scheme is used. The precision and recall calculated in an exact and relaxed match could be biased. It shows the overall evaluation. The model could be efficient in extracting one-word symptoms but that does not necessarily mean that model could extract more than one symptom. For instance, "shortness of breath" is a symptom, only "breath" is extracted. For an exact match, it would not consider this as a symptom at all and for a relaxed match, it would still give the calculated precision and recall. These values would be considered to evaluate the overall performance. Another advantage of this evaluation scheme is that it makes it possible to get insights into the model by knowing what makes the model give high or low precision and recall values. The n-gram precision and recall values are calculated by considering true positives, true negatives, false positives, and false negatives of the respected length of symptoms as shown in Table 7.9.

**Table 7.9.** Example of 1 Gram Evaluation Scheme.

| Golden Standard | Model Prediction | 1 Gram Model Prediction | 1 Gram Precision | 1 Gram Recall |
|---|---|---|---|---|
| fever, shortness of breath, pain | shortness of breath, pain | fever, pain | 1 | 0.5 |
| odynophagia, chest pain | odynophagia, chest pain | odynophagia | 1 | 1 |
| cough, sob, fevers, anorexia, weight loss | cough, SOB, fevers, weight loss | cough, sob, fevers | 1 | 0.75 |
| fatigue, malaise, weight gain, | tired, fatigue, malaise, weight gain | tired, fatigue, malaise | 0.67 | 1 |
| weight loss, early satiety, pain | weight loss, pain | pain | 1 | 1 |
| difficulty in completing project, pain | difficulty in completing project | | 0 | 0 |

**Table 7.10.** Example of 2 Gram Evaluation Scheme.

| Golden Standard | Model Prediction | 2 Gram Model Prediction | 2 Gram Precision | 2 Gram Recall |
|---|---|---|---|---|
| fever, shortness of breath, pain | shortness of breath, pain | | | |
| odynophagia, chest pain | odynophagia, chest pain | chest pain | 1 | 1 |
| cough, sob, fevers, anorexia, weight loss | cough, SOB, fevers, weight loss | weight loss | 1 | 1 |
| fatigue, malaise, weight gain, | tired, fatigue, malaise, weight gain | | 1 | 1 |
| weight loss, early satiety, pain | weight loss, pain | weight loss | 1 | 0.5 |
| difficulty in completing project, pain | difficulty in completing project | | | |

**Table 7.11.** Example of 3 Gram Evaluation Scheme.

| Golden Standard | Model Prediction | 3 Gram Model Prediction | 3 Gram Precision | 3 Gram Recall |
|---|---|---|---|---|
| fever, shortness of breath, pain | shortness of breath, pain | shortness of breath | 1 | 1 |
| odynophagia, chest pain | odynophagia, chest pain | | | |
| cough, sob, fevers, anorexia, weight loss | cough, SOB, fevers, weight loss | | | |
| fatigue, malaise, weight gain, | tired, fatigue, malaise, weight gain | | | |
| weight loss, early satiety, pain | weight loss, pain | | | |
| difficulty in completing project, pain | difficulty in completing project | | | |

**Table 7.12.** Example of 3+ Gram Evaluation Scheme.

| Golden Standard | Model Prediction | 3+ Gram | 3+ Gram Precision | 3+ Gram Recall |
|---|---|---|---|---|
| fever, shortness of breath, pain | shortness of breath, pain | | | |
| odynophagia, chest pain | odynophagia, chest pain | | | |
| cough, sob, fevers, anorexia, weight loss | cough, SOB, fevers, weight loss | | | |
| fatigue, malaise, weight gain, | tired, fatigue, malaise, weight gain | | | |
| weight loss, early satiety, pain | weight loss, pain | | | |
| difficulty in completing project, pain | difficulty in completing project | difficulty in completing project | 1 | 1 |

## 7.3 Model Evaluations

UMLS Metamap consists of several concepts and CUIs. Extracting concepts like "Sign or Symptom", "Physiologic" and "Mental or Behaviour" were not enough to extract all the symptoms from the dataset. If other concepts were included then more noise within the data was also extracted. The tradeoff between these led the count of false positives and false negatives to increase that resulted in low precision and low recall. Table 7.13 and Figure 7.14 show the overall performance of UMLS Metamap for the exact match and relaxed match was least compared to other methods. UMLS Metamap was able to capture the common symptoms but failed to extract new and long phrases as seen in Table 7.15.

**Table 7.13.** Results of Exact Match Evaluation.

| | | **Training Results** | **Testing Results** |
|---|---|---|---|
| **UMLS Metamap** | Precision | 0.51 | 0.46 |
| | Recall | 0.48 | 0.44 |
| | F1 | 0.5 | 0.5 |
| **StanfordNLP + DNN** | Precision | 0.83 | 0.75 |
| | Recall | 0.82 | 0.76 |
| | F1 | 0.82 | 0.75 |
| **BERT + CRF** | Precision | 0.82 | 0.71 |
| | Recall | 0.83 | 0.72 |
| | F1 | 0.82 | 0.71 |
| **BiLSTM +CRF** | Precision | 0.85 | 0.8 |
| | Recall | 0.84 | 0.79 |
| | F1 | 0.84 | 0.8 |
| **BioBERT+ CRF** | Precision | 0.9 | 0.85 |
| | Recall | 0.91 | 0.85 |
| | F1 | 0.9 | 0.9 |

Stanford CoreNLP with DNN showed promising performance when evaluated with an exact match and relaxed match. However, while analyzing the output data through n-gram evaluation it was observed that this model was able to extract one-word symptoms and long phrases but could not extract 2 and 3 words symptoms. Due to the model's architecture, it could capture long dependencies such as "erythema in right chest and discomfort and numbness in right arm" but failed to capture adjacent word dependency like "weight loss" and "altered mental status".

BERT with CRF showed good performance with the training set but when the unseen test model was fed into the model the testing results significantly decreased. The main objective of these models is to perform well with unseen data so it can be used to automate the annotation process. With the testing results obtained by exact and relaxed evaluation, this model can not be reliable to be deployed in the real-world. Furthermore, this model showcased high precision and recall only for 1 Gram evaluation. The model was not able to capture dependencies between the other words in the sentence. BiLSTM with CRF using the BioWord2Vec embeddings exhibited good performance with all evaluations - exact match, relaxed match and n-gram. The embeddings helped the model to capture unseen rare symptoms like "hypopneas" and "photophobia". The forward and backward pass of BiLSTM

**Table 7.14.** Results of Relaxed Match Evaluation.

| | | **Training Results** | **Testing Results** |
|---|---|---|---|
| **UMLS Metamap** | Precision | 0.54 | 0.51 |
| | Recall | 0.51 | 0.49 |
| | F1 | 0.52 | 0.5 |
| **StanfordNLP + DNN** | Precision | 0.86 | 0.84 |
| | Recall | 0.83 | 0.84 |
| | F1 | 0.84 | 0.84 |
| **BERT + CRF** | Precision | 0.86 | 0.77 |
| | Recall | 0.86 | 0.76 |
| | F1 | 0.86 | 0.77 |
| **BiLSTM +CRF** | Precision | 0.89 | 0.87 |
| | Recall | 0.87 | 0.82 |
| | F1 | 0.88 | 0.84 |
| **BioBERT+ CRF** | Precision | 0.93 | 0.87 |
| | Recall | 0.92 | 0.86 |
| | F1 | 0.92 | 0.86 |

with CRF captured long dependencies like "lower extremity discomfort" and "difficulty in completing project". However, the longer the phrases get, the less likely it is for the model to recognize.

**Table 7.15.** Results of n-Gram Evaluation.

| | | **Training Results** | | | | **Testing Results** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **1 Gram** | **2 Gram** | **3 Gram** | **3+ Gram** | **1 Gram** | **2 Gram** | **3 Gram** | **3+ Gram** |
| **UMLS Metamap** | Precision | 0.21 | 0.2 | 0.1 | 0 | 0.2 | 0.15 | 0 | 0 |
| | Recall | 0.17 | 0.2 | 0.1 | 0 | 0.2 | 0.1 | 0 | 0 |
| | F1 Score | 0.19 | 0.2 | 0.1 | 0 | 0.2 | 0.12 | 0 | 0 |
| **StanfordNLP + DNN** | Precision | 0.73 | 0.36 | 0.32 | 0.23 | 0.62 | 0.47 | 0.4 | 0 |
| | Recall | 0.73 | 0.34 | 0.32 | 0.23 | 0.69 | 0.45 | 0.4 | 0 |
| | F1 Score | 0.73 | 0.35 | 0.32 | 0.23 | 0.65 | 0.46 | 0.4 | 0 |
| **BERT + CRF** | Precision | 0.7 | 0.59 | 0.43 | 0.07 | 0.57 | 0.36 | 0.2 | 0 |
| | Recall | 0.73 | 0.6 | 0.43 | 0.07 | 0.61 | 0.32 | 0.2 | 0 |
| | F1 Score | 0.71 | 0.6 | 0.43 | 0.07 | 0.59 | 0.34 | 0.2 | 0 |
| **BiLSTM + CRF** | Precision | 0.75 | 0.61 | 0.33 | 0.14 | 0.65 | 0.57 | 0 | 0 |
| | Recall | 0.76 | 0.6 | 0.33 | 0.14 | 0.67 | 0.57 | 0 | 0 |
| | F1 Score | 0.75 | 0.6 | 0.33 | 0.14 | 0.66 | 0.57 | 0 | 0 |
| **BioBERT + CRF** | Precision | 0.84 | 0.67 | 0.57 | 0.27 | 0.76 | 0.57 | 0.8 | 0 |
| | Recall | 0.85 | 0.67 | 0.57 | 0.27 | 0.77 | 0.55 | 0.8 | 0 |
| | F1 Score | 0.84 | 0.67 | 0.57 | 0.27 | 0.76 | 0.56 | 0.8 | 0 |

BioBERT with CRF outperformed other models in extracting symptoms. With a low count of false positives and false negatives, the model captured the majority of symptoms irrespective of its length and obtained maximum precision and recall. The model recognized

the symptoms from the complexly structured sentences and also extracted rare symptoms observed. With the context-based embeddings, the maximum count of long phrased symptoms was also extracted compared to other models. Some of the long phrased symptoms like "erythema in right chest and discomfort and numbness in right arm", "pain in lower back and abdomen", "numbness in fingertips and toes", etc. This model displayed significantly good performance compared to other models for the dataset with n-gram evaluation.

**Table 7.16.** Analysis of rare symptom recognition by the models.

| | UMLS Metamap | StanfordNLP +DNN | BERT + CRF | BiLSTM +CRF | BioBERT + CRF |
|---|---|---|---|---|---|
| photophobia | | | | ✓ | ✓ |
| hypopneas | | ✓ | ✓ | ✓ | ✓ |
| weight loss | | | ✓ | ✓ | ✓ |
| loss of appetite | | ✓ | ✓ | ✓ | ✓ |
| altered metal status | ✓ | | ✓ | ✓ | ✓ |
| lower extremity discomfort | | | | | ✓ |
| chest and left arm pains | ✓ | ✓ | | ✓ | ✓ |
| numbness in fingertips and toes | | | | | ✓ |
| pain in lower back and abdomen | | ✓ | | | ✓ |
| erythema in right chest and discomfort and numbness in right arm | | ✓ | | | ✓ |

## 7.4 COVID-19 Results

To analyze the performance of the BioBERT with CRF the model was further tested on COVID-19 Twitter tweets. The model was used to extract the symptoms observed by COVID-19 patients. The medical research on the COVID-19 virus is in its development phase. Currently, there is very little information known about the virus. This work would help the researchers to know more about the virus and the common symptoms faced by the patients throughout the world. The previously trained model was fed in with the tweets to investigate if it can capture known and any new symptoms.

Figure 7.1 shows few symptoms that were tweeted the most in March. It was the initial time when COVID-19 started spreading across the world, and several countries reported a few cases. People were scared and shared the initial symptoms that they observed. Most of the tweets included symptoms like "fever" and "cough". A lot of patients suffered from breathing problems and tweeted about "shortness of breath", "difficulty breathing", "short of breath", "trouble breathing", etc. These were the common symptoms observed among all patients and released by the CDC. People were panicking and shared their concerns. People also posted about them suffering from common symptoms like "headache" and "sneezing"

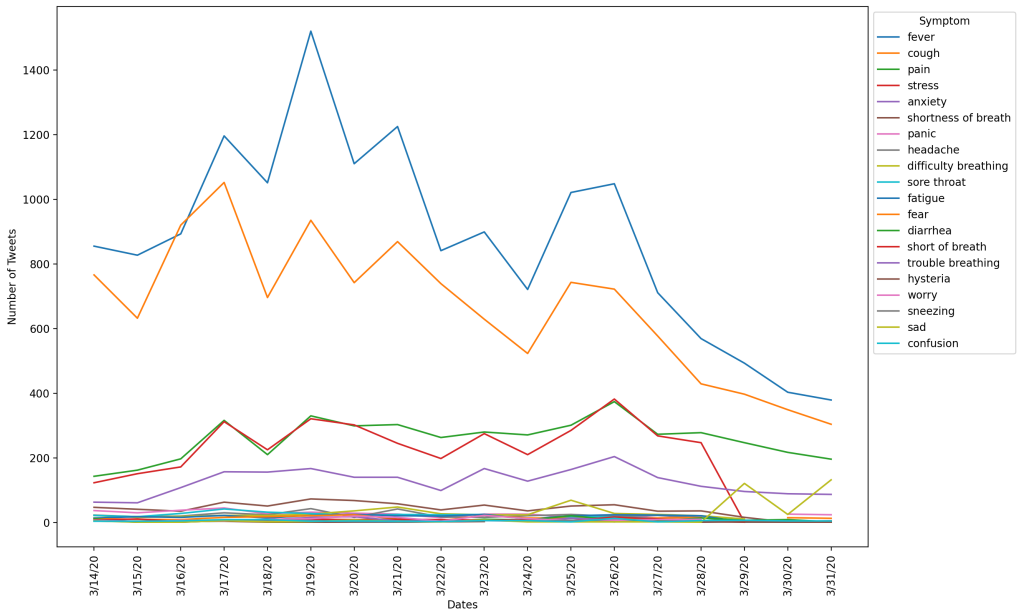



**Figure 7.1.** Top 20 COVID-19 Symptoms Extracted from Tweets in March.

during the global pandemic. One of the symptoms extracted is "diarrhea", this symptom was much later added to the CDC COVID-19 symptom list. Among all the symptoms major concerns were regarding mental health issues. Several people tweeted about "stress", "anxiety", "panic", "feat", "hysteria", etc.

With nationwide lockdowns across the world, eventually more people started tweeting about mental health. In April, the number of tweets regarding COVID-19 increased significantly but fewer people tweeted about the symptoms observed. However, new symptoms were extracted other than the CDC listed as shown in Figure 7.2. People were aware of the common symptoms thus tweeted about other illnesses. With the common symptoms like "fever", "cough", "sob", etc people posted about their psychological problems. These problems include symptoms like "sad", "tired", "fear", etc. People have also complained about "anxiety" and "depression" disorders.

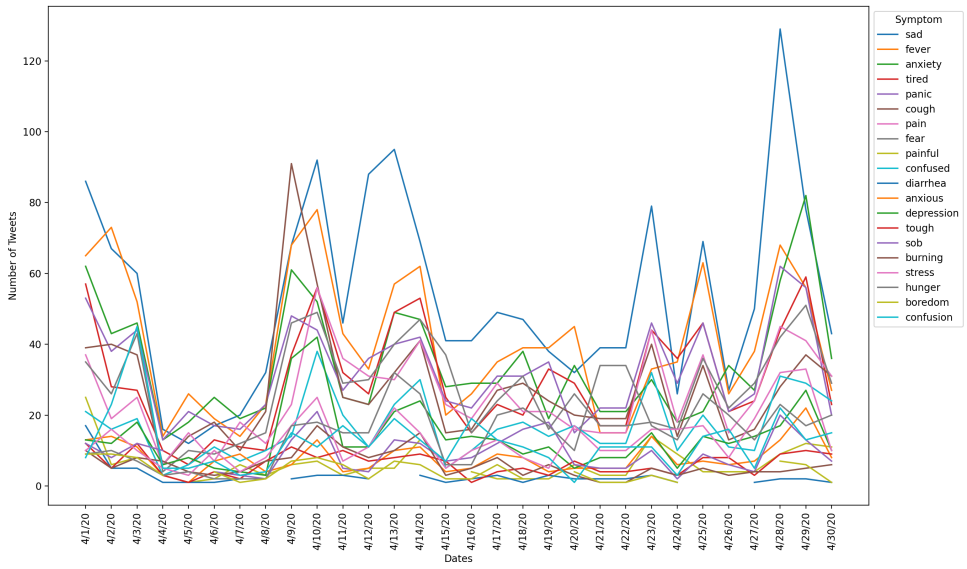



**Figure 7.2.** Top 20 COVID-19 Symptoms Extracted from Tweets in April.

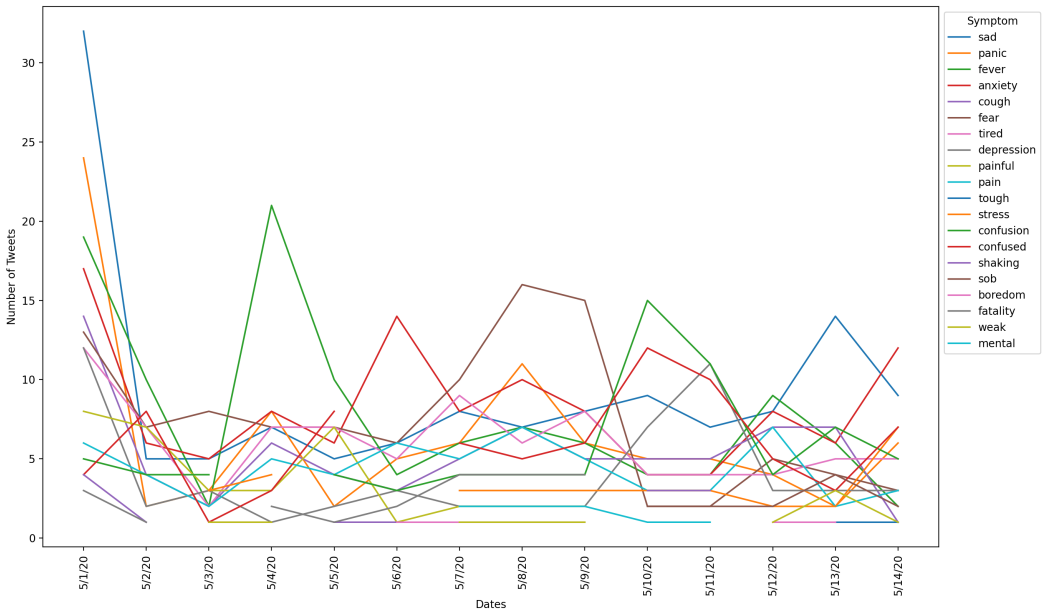



**Figure 7.3.** Top 20 COVID-19 Symptoms Extracted from Tweets in May.

A recent study shows that COVID-19 illness can be classified into six categories as per the symptoms observed by the patients. Table 7.17 shows that the model was able to extract all listed symptoms of all six classes besides the negated values like “no cough” and “no fever”. The model implemented in this work can also be used to analyze the symptoms observed by the patients and suggest the severity of COVID-19 illness. The symptoms can be extracted from the narratives of tweets, complaints, medical notes, etc. If the patient suffers from “headache”, “loss of smell”, “muscle pain”, “cough”, “sore throat”, and “chest pain” then the patient might be suffering from class 1, flu-like with no fever COVID-19 illness. Table 7.17 also shows the number of people tweeted about these symptoms in March, April, and May.

**Table 7.17.** Model Extracted Symptoms based on 6 Classes of COVID-19 Symptoms

| | Symptom | Model Extracted Symptoms | Count from March Tweets | Count from April Tweets | Count from May Tweets |
|---|---|---|---|---|---|
| Class 1: Flu-like with no fever | headache | ✓ | 355 | 120 | 5 |
| | loss of smell | ✓ | 6 | 22 | |
| | muscle pain | ✓ | 8 | 14 | |
| | cough | ✓ | 12024 | 836 | 75 |
| | sore throat | ✓ | 334 | 52 | |
| | chest pain | ✓ | 158 | 16 | |
| | no fever | | | | |
| Class 2: Flu-like with fever & headache | headache | ✓ | 355 | 120 | 5 |
| | loss of smell | ✓ | 6 | 22 | |
| | sore throat | ✓ | 334 | 52 | |
| | loss of appetite | ✓ | 7 | 4 | |
| | cough | ✓ | 12024 | 836 | 75 |
| | hoarseness | ✓ | 2 | 1 | 2 |
| | fever | ✓ | 15762 | 1207 | 110 |
| Class 3: Gastrointestinal & headache | headache | ✓ | 355 | 120 | 5 |
| | loss of smell | ✓ | 6 | 22 | |
| | loss of appetite | ✓ | 7 | 4 | |
| | diarrhea | ✓ | 140 | 71 | 10 |
| | sore throat | ✓ | 334 | 52 | |
| | chest pain | ✓ | 158 | 16 | |
| | no cough | | | | |
| Class 4: Severe level one, fatigue & headache | headache | ✓ | 355 | 120 | 5 |
| | loss of smell | ✓ | 6 | 22 | |
| | fever | ✓ | 15762 | 1207 | 110 |
| | hoarseness | ✓ | 2 | 1 | 2 |
| | chest pain | ✓ | 158 | 16 | |
| | cough | ✓ | 12024 | 836 | 75 |
| | fatigue | ✓ | 300 | 203 | 29 |
| Class 5: Severe level two, confusion | headache | ✓ | 355 | 120 | 5 |
| | loss of smell | ✓ | 6 | 22 | |
| | cough | ✓ | 12024 | 836 | 75 |
| | fever | ✓ | 15762 | 1207 | 110 |
| | confusion | ✓ | 99 | 366 | 74 |
| | loss of appetite | ✓ | 7 | 4 | |
| | hoarseness | ✓ | 2 | 1 | 2 |
| | sore throat | ✓ | 334 | 52 | |
| | chest pain | ✓ | 158 | 16 | |
| | fatigue | ✓ | 300 | 203 | 29 |
| | muscle pain | ✓ | 8 | 14 | |
| Class 6: Severe level three, abdominal and respiratory | headache | ✓ | 355 | 120 | 5 |
| | loss of smell | ✓ | 6 | 22 | |
| | loss of appetite | ✓ | 7 | 4 | |
| | cough | ✓ | 12024 | 836 | 75 |
| | fever | ✓ | 15762 | 1207 | 110 |
| | hoarseness | ✓ | 2 | 1 | 2 |
| | sore throat | ✓ | 334 | 52 | |
| | chest pain | ✓ | 158 | 16 | |
| | fatigue | ✓ | 300 | 203 | 29 |
| | confusion | ✓ | 99 | 366 | 74 |
| | muscle pain | ✓ | 8 | 14 | |
| | diarrhea | ✓ | 140 | 71 | 10 |
| | shortness of breath | ✓ | 761 | 49 | 18 |
| | abdominal pain | ✓ | 69 | 6 | |

# 8. CONCLUSION

In this research, different models to extract symptoms from the biomedical text are evaluated. Different types of word embeddings are used to convert the words into the vectors and are then fed into the models. The results of all models are compared and a model is used to extract symptoms of the COVID-19 virus.

The essential data extracted about the patient's illness in an unstructured format, as discussed in Chapter 2, makes it difficult for a human to annotate symptoms. Chapter 3 discusses the concepts used to extract the symptoms from the biomedical text. Named Entity Recognition (Section 3.1) is used to annotate the symptoms within the text, Conditional Random Field (Section 3.2) is used to find the dependency among the annotations, and word embeddings (Section 3.3) are used to convert words into vectors. Chapter 4 discusses the related work done in this domain.

Chapter 5 discusses the details about the model architectures used to extract symptoms. The UMLS MetaMap (Section 5.1) is a natural language processing tool that uses various sources to categorize the phrases or terms in the text to different semantic types, can be used to extract information. Stanford CoreNLP (Section 5.2), a dependency parser, is used to find the syntactical associations between the words of a sentence.

Bidirectional Long Short Term Memory due to its architecture (Section 5.3) can co-relate information between segments that are separated in the input. To capture the hard constraints in output tags a CRF layer is added on the top. BioWord2Vec word embeddings, trained on the medical corpus, are used to generate word vectors. To produce word embeddings based on the context, BERT and BioBERT are used. To compare the dependencies of the output tags, a CRF layer is added to the top of the models, as discussed in Section 5.4.

The details about the different data sources are discussed in Chapter 6. A subset of the dataset is created for the clinicians to annotate the symptoms. $BIOE$ tagging is used to annotate the symptoms. COVID-19 Twitter tweets were retrieved to evaluate the model performance. The performance of the models is evaluated through different evaluation schemes, discussed in Section 7.2.1. The exact match evaluation scheme considers the match is correct only if the model prediction is identical to the golden standard. The relaxed match breaks

the model predictions and golden standards into a sequence of tokens and then performs an exact match evaluation. Therefore, partially extracted symptoms are also given some weightage. The n-gram evaluation scheme measures the performance based on the length of symptom phrases.

The results are promising as discussed in Section 7.3 and show BioBERT with a CRF layer comparatively performs better than compared to other models. All evaluation schemes showed that it can extract human-labeled as well as new and rare symptoms. Section 7.4 shows the results of the proposed BioBERT with a CRF layer model implemented on COVID-19 tweets. In March, a lot of people tweeted about the COVID-19 symptoms. Many of those symptoms were present in the list of COVID-19 symptoms posted by the CDC. In April and May, many people shared their concerns about their mental health issues.

In conclusion, the work presented as a part of this research could automate the process of symptom annotation and also be used to extract symptoms of known as well as unknown illnesses.

# REFERENCES


[1] S. Singh, *Natural language processing for information extraction*, 2018. arXiv: 1807.02383 [cs.CL].

[2] V. Ehrenstein, *Obtaining data from electronic health records*. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK551878/.

[3] A. K. Jha, M. F. Burke, C. DesRoches, M. S. Joshi, P. D. Kralovec, E. G. Campbell, and M. B. Buntin, "Progress toward meaningful use: Hospitals' adoption of electronic health records," *The American journal of managed care*, vol. 17, no. 12 Spec No. SP117–24, Dec. 2011, ISSN: 1088-0224. [Online]. Available: http://europepmc.org/abstract/MED/22216770.

[4] Office of the National Coordinator for Health Information Technology (ONC), Department of Health and Human Services, "Health information technology: Standards, implementation specifications, and certification criteria for electronic health record technology, 2014 edition; revisions to the permanent certification program for health information technology. final rule," *Federal register*, vol. 77, no. 171, pp. 54 163–54 292, Sep. 2012, ISSN: 0097-6326. [Online]. Available: http://europepmc.org/abstract/MED/22946139.

[5] C. for Drug Evaluation and Research, *National drug code directory*. [Online]. Available: https://www.fda.gov/drugs/drug-approvals-and-databases/national-drug-code-directory.

[6] *Rxnorm*. [Online]. Available: https://www.nlm.nih.gov/research/umls/rxnorm/.

[7] *Snomed home page*. [Online]. Available: https://www.snomed.org/.

[8] *The anatomical therapeutic chemical classification system with defined daily doses (atc/ddd)*, Dec. 2010. [Online]. Available: http://www.who.int/classifications/atcddd/en/.

[9] *Icd - icd-10-cm - international classification of diseases, tenth revision, clinical modification*, Jul. 2020. [Online]. Available: https://www.cdc.gov/nchs/icd/icd10cm.htm.

[10] *Cpt® (current procedural terminology)*. [Online]. Available: https://www.ama-assn.org/amaone/cpt-current-procedural-terminology.

[11] *Hcpcs - general information*. [Online]. Available: https://www.cms.gov/Medicare/Coding/MedHCPCSGenInfo/index.html.

[12] *Home*, Sep. 2020. [Online]. Available: https://loinc.org/.

[13] R. E. Gliklich, *Registries for evaluating patient outcomes*. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK208616/.

[14] I. of Medicine (US) Committee on Data Standards for Patient Safety. (Jan. 1970). Key capabilities of an electronic health record system: Letter report, [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK551878/#.

[15] A. al-Aiad, R. Duwairi, and M. Fraihat, "Survey: Deep learning concepts and techniques for electronic health record," in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, 2018, pp. 1–5.

[16] I. Spasic and G. Nenadic, "Clinical text data in machine learning: Systematic review," *JMIR Med Inform*, vol. 8, no. 3, e17984, Mar. 2020, ISSN: 2291-9694. DOI: 10.2196/17984. [Online]. Available: https://doi.org/10.2196/17984.

[17] W.-H. Weng, K. Wagholikar, A. McCray, P. Szolovits, and H. Chueh, "Medical subdomain classification of clinical notes using a machine learning-based natural language processing approach," *BMC Medical Informatics and Decision Making*, vol. 17, p. 155, Dec. 2017. DOI: 10.1186/s12911-017-0556-8.

[18] B. Jin, C. Che, Z. Liu, S. Zhang, X. Yin, and X. Wei, "Predicting the risk of heart failure with ehr sequential data modeling," *Ieee Access*, vol. 6, pp. 9256–9261, 2018.

[19] G. Maragatham and S. Devi, "Lstm model for prediction of heart failure in big data," *Journal of medical systems*, vol. 43, no. 5, p. 111, 2019.

[20] T. Garske, "Using deep learning on ehr data to predict diabetes," PhD thesis, Ph. D. Thesis, University of Colorado, Denver, CO, USA, 2018.[Google Scholar], 2018.

[21] Y.-H. Wang, P. A. Nguyen, M. M. Islam, Y.-C. Li, and H.-C. Yang, "Development of deep learning algorithm for detection of colorectal cancer in ehr data.," in *MedInfo*, 2019, pp. 438–441.

[22] H. Guo, "Accelerated continuous conditional random fields for load forecasting," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, 2016, pp. 1492–1493.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, 2013. arXiv: 1301.3781 [cs.CL].

[24] V. Taslimitehrani, G. Dong, N. L. Pereira, M. Panahiazar, and J. Pathak, "Developing ehr-driven heart failure risk prediction models using cpxr (log) with the probabilistic loss function," *Journal of biomedical informatics*, vol. 60, pp. 260–269, 2016.

[25] T. Zheng, W. Xie, L. Xu, X. He, Y. Zhang, M. You, G. Yang, and Y. Chen, "A machine learning-based framework to identify type 2 diabetes through electronic health records," *International journal of medical informatics*, vol. 97, pp. 120–127, 2017.

[26] M. Panahiazar, V. Taslimitehrani, N. Pereira, and J. Pathak, "Using ehrs and machine learning for heart failure survival analysis," *Studies in health technology and informatics*, vol. 216, p. 40, 2015.

[27] R. Miotto, L. Li, B. A. Kidd, and J. T. Dudley, "Deep patient: An unsupervised representation to predict the future of patients from the electronic health records," *Scientific reports*, vol. 6, no. 1, pp. 1–10, 2016.

[28] L. Wang, W. Zhang, X. He, and H. Zha, "Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2447–2456.

[29] Y. Cheng, F. Wang, P. Zhang, and J. Hu, "Risk prediction with electronic health records: A deep learning approach," in *Proceedings of the 2016 SIAM International Conference on Data Mining*, SIAM, 2016, pp. 432–440.

[30] K. Wagstaff, R. Francis, T. Gowda, Y. Lu, E. Riloff, K. Singh, and N. L. Lanza, "Mars target encyclopedia: Rock and soil composition extracted from the literature," in *AAAI*, 2018.

[31] B. Tang, D. Jiang, Q. Chen, X. Wang, J. Yan, and Y. Shen, "De-identification of clinical text via bi-lstm-crf with neural language models," *AMIA ... Annual Symposium proceedings. AMIA Symposium*, vol. 2019, pp. 857–863, 2019, ISSN: 1942-597X. [Online]. Available: https://europepmc.org/articles/PMC7153082.

[32] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, "Joint extraction of entities and relations based on a novel tagging scheme," *CoRR*, vol. abs/1706.05075, 2017. arXiv: 1706.05075. [Online]. Available: http://arxiv.org/abs/1706.05075.

[33] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 2335–2344. [Online]. Available: https://www.aclweb.org/anthology/C14-1220.

[34] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 207–212. DOI: 10.18653/v1/P16-2034. [Online]. Available: https://www.aclweb.org/anthology/P16-2034.

[35] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 2124–2133. DOI: 10.18653/v1/P16-1200. [Online]. Available: https://www.aclweb.org/anthology/P16-1200.

[36] M. Miwa and M. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures," *CoRR*, vol. abs/1601.00770, 2016. arXiv: 1601.00770. [Online]. Available: http://arxiv.org/abs/1601.00770.

[37] S. Moen and T. S. S. Ananiadou, "Distributional semantics resources for biomedical text processing," in *Proceedings of the 5th International Symposium on Languages in Biology and Medicine, Tokyo, Japan*, 2013, pp. 39–43.

[38] S. Tulkens, S. Suster, and W. Daelemans, "Using distributed representations to disambiguate biomedical and clinical concepts," in *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, 2016.

[39] Y. Zhu, E. Yan, and F. Wang, "Semantic relatedness and similarity of biomedical terms: Examining the effects of recency, size, and section of biomedical publications on the performance of word2vec," *BMC Medical Informatics and Decision Making*, vol. 17, pp. 95–103, 2017.

[40] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa, "Natural language processing (almost) from scratch," *CoRR*, vol. abs/1103.0398, 2011. arXiv: 1103.0398. [Online]. Available: http://arxiv.org/abs/1103.0398.

[41] V. Yadav and S. Bethard, "A survey on recent advances in named entity recognition from deep learning models," in *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 2145–2158. [Online]. Available: https://www.aclweb.org/anthology/C18-1182.

[42] J. Santos, J. Terra, B. S. Consoli, and R. Vieira, "Multidomain contextual embeddings for named entity recognition," in *IberLEF@SEPLN*, 2019.

[43] T. H. Nguyen, A. Sil, G. Dinu, and R. Florian, *Toward mention detection robustness with recurrent neural networks*, 2016. arXiv: 1602.07749 [cs.CL].

[44] Z. Huang, W. Xu, and K. Yu, *Bidirectional lstm-crf models for sequence tagging*, 2015. arXiv: 1508.01991 [cs.CL].

[45] E. Strubell, P. Verga, D. Belanger, and A. McCallum, *Fast and accurate entity recognition with iterated dilated convolutions*, 2017. arXiv: 1702.02098 [cs.CL].

[46] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[47] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, *Neural architectures for named entity recognition*, 2016. arXiv: 1603.01360 [cs.CL].

[48] P. V. Q. de Castro, N. F. F. da Silva, and A. da Silva Soares, "Portuguese named entity recognition using lstm-crf," in *PROPOR*, 2018.

[49] D. Demner-Fushman, W. W. Chapman, and C. J. McDonald, "What can natural language processing do for clinical decision support?" *Journal of biomedical informatics*, vol. 42, no. 5, pp. 760–772, 2009.

[50] J. Gulla, P. M. Neri, D. W. Bates, and L. Samal, "User requirements for a chronic kidney disease clinical decision support tool to promote timely referral," *International journal of medical informatics*, vol. 101, pp. 50–57, 2017.

[51] L. Soldaini, A. Yates, and N. Goharian, "Denoising clinical notes for medical literature retrieval with convolutional neural model," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 2307–2310.

[52] P. Gandhi, X. Luo, S. Storey, Z. Zhang, Z. Han, and K. Huang, "Identifying symptom clusters in breast cancer and colorectal cancer patients using ehr data," in *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, ser. BCB '19, Niagara Falls, NY, USA: Association for Computing Machinery, 2019, pp. 405–413, ISBN: 9781450366663. DOI: 10.1145/3307339.3342164. [Online]. Available: https://doi.org/10.1145/3307339.3342164.

[53] *Unified medical language system (umls)*. [Online]. Available: https://www.nlm.nih.gov/research/umls/index.html.

[54] P. Gandhi, X. Luo, and R. Tian, "Modeling vehicle-pedestrian encountering risks in the natural driving environment using machine learning algorithms," in. Jun. 2019, pp. 382–393, ISBN: 978-3-030-22215-4. DOI: 10.1007/978-3-030-22216-1_28.

[55] M. E. Peters, W. Ammar, C. Bhagavatula, and R. Power, *Semi-supervised sequence tagging with bidirectional language models*, 2017. arXiv: 1705.00108 [cs.CL].

[56] Y. Zhang, Q. Chen, Z. Yang, H. Lin, and Z. Lu, "Biowordvec, improving biomedical word embeddings with subword information and mesh," *Scientific data*, vol. 6, no. 1, pp. 1–9, 2019.

[57] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[58] B. Chiu, G. Crichton, A. Korhonen, and S. Pyysalo, "How to train good word embeddings for biomedical nlp," in *Proceedings of the 15th workshop on biomedical natural language processing*, 2016, pp. 166–174.

[59] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, p. 160 035, 2016.

[60] F. Souza, R. Nogueira, and R. Lotufo, *Portuguese named entity recognition using bert-crf*, 2020. arXiv: 1909.10649 [cs.CL].

[61] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. [Online]. Available: https://arxiv.org/abs/1810.04805.

[62] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Pre-trained bert using tensorflow hub*, 2018. [Online]. Available: https://tfhub.dev/google/bert_uncased_L-12_H-768_A-12/1.

[63] Y. Liao, X. Jiang, and Q. Liu, *Probabilistically masked language model capable of autoregressive generation in arbitrary word order*, 2020. arXiv: 2004.11579 [cs.CL].

[64] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, *Aligning books and movies: Towards story-like visual explanations by watching movies and reading books*, 2015. arXiv: 1506.06724 [cs.CV].

[65] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson, *One billion word benchmark for measuring progress in statistical language modeling*, 2014. arXiv: 1312.3005 [cs.CL].

[66] J. A. Smith and A. Woodcock, "Chronic cough," *New England Journal of Medicine*, vol. 375, no. 16, pp. 1544–1551, 2016.

[67] C. L. French, R. S. Irwin, F. J. Curley, and C. J. Krikorian, "Impact of chronic cough on quality of life," *Archives of internal medicine*, vol. 158, no. 15, pp. 1657–1661, 1998.

[68] A. H. Morice, A. D. Jakes, S. Faruqi, S. S. Birring, L. McGarvey, B. Canning, J. A. Smith, S. M. Parker, K. F. Chung, K. Lai, *et al.*, "A worldwide survey of chronic cough: A manifestation of enhanced somatosensory response," *European Respiratory Journal*, vol. 44, no. 5, pp. 1149–1155, 2014.

[69] M. Weiner, J. Weaver, P. Dexter, A. Roberts, Z. Liu, S. Hui, A. Church, I. Doshi, and K. Heithoff, "A semi-automated approach to identifying chronic cough in electronic health records," *Annals of Allergy, Asthma & Immunology*, vol. 121, no. 5, S57, 2018.

[70] C. Miaskowski, B. A. Cooper, S. M. Paul, M. Dodd, K. Lee, B. E. Aouizerat, C. West, M. Cho, and A. Bank, "Subgroups of patients with cancer with different symptom experiences and quality-of-life outcomes: A cluster analysis.," in *Oncology nursing forum*, vol. 33, 2006.

[71] C. Miaskowski, L. Dunn, C. Ritchie, S. M. Paul, B. Cooper, B. E. Aouizerat, K. Alexander, H. Skerman, and P. Yates, "Latent class analysis reveals distinct subgroups of patients based on symptom occurrence and demographic and clinical characteristics," *Journal of pain and symptom management*, vol. 50, no. 1, pp. 28–37, 2015.

[72] L. Ramshaw and M. Marcus, "Text chunking using transformation-based learning," in *Third Workshop on Very Large Corpora*, 1995. [Online]. Available: https://www.aclweb.org/anthology/W95-0107.

[73] J. Bian, Y. Guo, Z. He, and X. Hu, *Social Web and Health Research*. Springer, 2019.

[74] J. WHO Centre for Health Development (Kobe, *A glossary of terms for community health care and services for older persons*, 2004.

[75] D. King, D. Ramirez-Cano, F. Greaves, I. Vlaev, S. Beales, and A. Darzi, "Twitter and the health reforms in the english national health service," *Health Policy*, vol. 110, no. 2, pp. 291–297, 2013, ISSN: 0168-8510. DOI: https://doi.org/10.1016/j.healthpol.2013.02.005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168851013000456.

[76] R. Chunara, J. R. Andrews, and J. S. Brownstein, "Social and news media enable estimation of epidemiological patterns early in the 2010 haitian cholera outbreak," *The American Journal of Tropical Medicine and Hygiene*, vol. 86, no. 1, pp. 39–45, 2012, ISSN: 0002-9637. DOI: https://doi.org/10.4269/ajtmh.2012.11-0597. [Online]. Available: https://www.ajtmh.org/content/journals/10.4269/ajtmh.2012.11-0597.

[77] D. Sayce, *The number of tweets per day in 2020*, Jul. 2020. [Online]. Available: https://www.dsayce.com/social-media/tweets-day/.

[78] M. Cevik, C. Bamford, and A. Ho, "Covid-19 pandemic—a focused review for clinicians," *Clinical Microbiology and Infection*, vol. 26, no. 7, pp. 842–847, 2020, ISSN: 1198-743X. DOI: https://doi.org/10.1016/j.cmi.2020.04.023. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1198743X20302317.

[79] K. Lerman and R. Ghosh, *Information contagion: An empirical study of the spread of news on digg and twitter social networks*, 2010. arXiv: 1003.2664 [cs.CY].

[80] D. M. Romero, B. Meeder, and J. Kleinberg, "Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter," in *Proceedings of the 20th International Conference on World Wide Web*, ser. WWW '11, Hyderabad, India: Association for Computing Machinery, 2011, pp. 695–704, ISBN: 9781450306324. DOI: 10.1145/1963405.1963503. [Online]. Available: https://doi.org/10.1145/1963405.1963503.

[81] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Commun. ACM*, vol. 59, no. 7, pp. 96–104, Jun. 2016, ISSN: 0001-0782. DOI: 10.1145/2818717. [Online]. Available: https://doi.org/10.1145/2818717.

[82] H. Liang, I. Fung, Z. Tse, J. Yin, C.-H. Chan, L. Pechta, B. Smith, R. Marquez-Lameda, M. Meltzer, K. Lubell, and K.-w. Fu, "How did ebola information spread on twitter: Broadcasting or viral spreading?" *BMC Public Health*, vol. 19, Dec. 2019. DOI: 10.1186/s12889-019-6747-8.

[83] C. Chew and G. Eysenbach, "Pandemics in the age of twitter: Content analysis of tweets during the 2009 h1n1 outbreak," *PLOS ONE*, vol. 5, no. 11, pp. 1–13, Nov. 2010. DOI: 10.1371/journal.pone.0014118. [Online]. Available: https://doi.org/10.1371/journal.pone.0014118.

[84] H. W. Park, S. Park, and M. Chong, "Conversations and medical news frames on twitter: Infodemiological study on covid-19 in south korea," *J Med Internet Res*, vol. 22, no. 5, e18897, May 2020, ISSN: 1438-8871. DOI: 10.2196/18897. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/32325426.

[85] A. Abd-Alrazaq, D. Alhuwail, M. Househ, M. Hamdi, and Z. Shah, "Top concerns of tweeters during the covid-19 pandemic: Infoveillance study," *J Med Internet Res*, vol. 22, no. 4, e19016, Apr. 2020, ISSN: 1438-8871. DOI: 10.2196/19016. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/32287039.

[86] E. Ferrara, "What types of covid-19 conspiracies are populated by twitter bots?" *First Monday*, May 2020, ISSN: 1396-0466. DOI: 10.5210/fm.v25i6.10633. [Online]. Available: http://dx.doi.org/10.5210/fm.v25i6.10633.

[87] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02, Philadelphia, Pennsylvania: Association for Computational Linguistics, 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. [Online]. Available: https://doi.org/10.3115/1073083.1073135.

# PUBLICATIONS

1. Priyanka Gandhi, Xiao Luo, Renran Tian, Modeling Vehicle-Pedestrian Encountering Risks in the Natural Driving Environment Using Machine Learning Algorithms. In: Duffy V. (eds) Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. Human Body and Motion. Human Computer Interaction International (HCII) 2019. (Awarded as the Best Paper)

2. Priyanka Gandhi, Xiao Luo, Susan Storey, Zuoyi Zhang, Zhi Han, and Kun Huang. 2019. Identifying Symptom Clusters in Breast Cancer and Colorectal Cancer Patients using EHR Data. In Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '19). Association for Computing Machinery, New York, NY, USA, 405–413. (Awarded as the Best Paper)

3. Xiao Luo, Priyanka Gandhi, Wei Shao, Zuoyi Zhang, Zhi Han, Vasu Chandrasekaran, Vladimir Turzhitsky, Vishal Bali, Anna R. Roberts, Megan Metzger, Jarod Baker, Carmen La Rosa, Jessica Weaver, Paul Dexter and Kun Huang, Predicting Chronic Cough with Random Forests Based on EHR Data. ACAAI, 2020.

4. Xiao Luo, Haoran Ding, Matthew Tang, Priyanka Gandhi, Zhan Zhang, Zhe He, Attention Mechanism with BERT for Content Annotation and Categorization of Pregnancy-Related Questions on a Community Q&A Site. IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2020.

5. Susan Storey, Xiao Luo, Zuoyi Zhang, Megan Metzger, Diane Von Ah, Priyanka Rakesh Gandhi, Wei Shao, Kun Huang, Symptom Clusters in colorectal cancer survivors with and without comorbid diabetes, MASCC Annual Meeting on Supportive Care in Cancer, 2020.

6. Xiao Luo, Priyanka Gandhi, Susan Storey, Zuoyi Zhang, Zhi Han and Kun Huang, A Novel Framework to Analyze the Associations between Symptoms and Cancer Patient Attributes Post Chemotherapy using EHR data, IEEE Journal of Biomedical and Health Informatics, 2020. (Submitted)

7. Xiao Luo, Susan Storey, Priyanka Gandhi, Zuoyi Zhang, Megan Metzger and Kun Huang, Analyzing the Symptoms in Colorectal and Breast Cancer Patients with or without Type 2 Diabetes using EHR Data. Health Informatics Journal, 2020. (Submitted)

8. Xiao Luo, Priyanka Gandhi, Zuoyi Zhang, Wei Shao, Zhi Han, Vasu Chandrasekaran, Vladimir Turzhitsky, Vishal Bali, Anna R. Roberts, Megan Metzger, Jarod Baker, Carmen La Rosa, Jessica Weaver, Paul Dexter, and Kun Huang, Applying Machine Learning Models to Identify Chronic Cough Patients using EHR Data. (In Preparation)