

EXTRACTION OF KNOWLEDGE GRAPH OF COVID-19 THROUGH MINING OF UNSTRUCTURED BIOMEDICAL CORPORA

A PROJECT REPORT

Submitted by

Athiban T - 2018103013

Syed Mohamed Asif M - 2018103612

Prathesh N - 2018103576

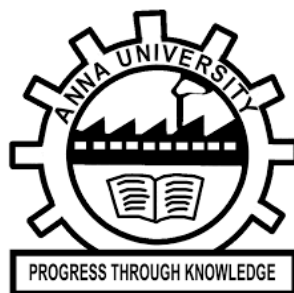
in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



**COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY :: CHENNAI 600 025**

JUNE 2022

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled “**EXTRACTION OF KNOWLEDGE GRAPH OF COVID-19 THROUGH MINING OF UNSTRUCTURED BIOMEDICAL CORPORA**” is the bonafide work of “**ATHIBAN T (2018103013), SYED MOHAMED ASIF M (2018103612), PRATHESH N (2018103576)**” who carried out the project work under my supervision, for the fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering.

Place: Chennai

Date:

Dr. G. Sudhakaran

SUPERVISOR

Teaching Fellow,

Department of Computer Science and Engineering,

Anna University, Chennai - 600 025.

COUNTERSIGNED

Dr. Valli S

Head of the Department,

Department of Computer Science and Engineering,

Anna University, Chennai - 600 025.

ACKNOWLEDGEMENT

We express our deep gratitude to our guide, **Dr. G. Sudhakaran**, Teaching Fellow, Department of Computer Science and Engineering, for guiding us through every phase of the project. We appreciate his thoroughness, tolerance and ability to share his knowledge with us. Apart from adding his own input, he has encouraged us to think on our own and give form to our thoughts. We owe him for harnessing our potential and bringing out the best in us.

We are extremely grateful to Dr. S. Valli, Professor and Head of the Department of Computer Science and Engineering, Anna University, Chennai - 25, for extending the facilities of the Department towards our project and for her unstinting support.

We express our thanks to the panel of reviewers **Dr. S. Chitrakala**, Professor, Department of Computer Science and Engineering, **Dr. V. Vetriselvi**, Professor, Department of Computer Science and Engineering and **Dr. G. S. Mahalakshmi**, Associate Professor, Department of Computer Science and Engineering for their valuable suggestions and critical reviews throughout the course of our project.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

ATHIBAN T

SYED MOHAMED ASIF M

PRATHESH N

ABSTRACT

The number of biomedical articles published is increasing rapidly over the years. Currently there are over 30 million publications in PubMed and over 25 million references in Medline. Among these concepts, Biomedical Named Entity Recognition (BioNER) and Biomedical Relation Extraction (BioRD) are the most important. Graphs are practical resources for defining relationships and are applicable in real-world scenarios. In the biomedical domain, Knowledge Graph is used to visualize the relationships between various entities such as proteins, chemicals and diseases. The system defines a biomedical knowledge graph as the following: a resource that integrates one or more expert-derived sources of information into a graph where nodes represent biomedical entities and edges represent relationships between two entities. The system uses Named Entity Recognition models for disease recognition, chemical recognition and protein recognition. Then the system uses the Chemical - Disease Relation Extraction and Chemical - Protein Relation Extraction models. And the system extracts the entities and relations from the CORD-19 dataset using the models. The system then creates a Knowledge Graph for the extracted relations and entities. The system performs Representation Learning on this KG to get the embeddings of all entities and get the top related diseases, chemicals and proteins with respect to COVID-19.

திட்டப்பணிச்சுருக்கம்

பல ஆண்டுகளாக வெளியிடப்பட்ட உயிரியல் மருத்துவக் கட்டுரைகளின் எண்ணிக்கை வேகமாக அதிகரித்து வருகிறது. தற்போது PubMed இணையதளத்தில் 30 மில்லியனுக்கும் அதிகமான வெளியீடுகளும், MEDLINE இணையதளத்தில் 25 மில்லியனுக்கும் அதிகமான குறிப்புகளும் உள்ளன. இந்த கருத்துக்களில், உயிரியல் மருத்துவத்தில் பெயரிடப்பட்ட நிறுவன அங்கீகாரம் மற்றும் உயிரியல் மருத்துவத்தில் உறவு பிரித்தெடுத்தல் ஆகியவை மிக முக்கியமானவை. வரைபடங்கள் உறவுகளை வரையறுப்பதற்கான நடைமுறை ஆதாரங்கள் மற்றும் அவை நிஜ உலக சூழ்நிலைகளில் பொருந்தும். உயிரியல் மருத்துவத்தில், புரதங்கள், இரசாயனங்கள் மற்றும் நோய்கள் போன்ற பல்வேறு நிறுவனங்களுக்கு இடையேயான உறவுகளைக் காட்சிப்படுத்த அறிவு வரைபடம் பயன்படுத்தப்படுகிறது. ஒரு உயிரியல் மருத்துவ அறிவு வரைபடத்தை கணினி பின்வருமாறு வரையறுக்கிறது: ஒன்று அல்லது அதற்கு மேற்பட்ட நிபுணரால் பெறப்பட்ட தகவல் ஆதாரங்களை ஒரு வரைபடத்தில் ஒருங்கிணைக்கும் ஒரு ஆதாரம், இதில் கணுக்கள் உயிரியல் மருத்துவ நிறுவனங்களைக் குறிக்கின்றன மற்றும் விளிம்புகள் இரண்டு நிறுவனங்களுக்கு இடையிலான உறவுகளைக் குறிக்கின்றன. நோய் கண்டறிதல், இரசாயன அங்கீகாரம் மற்றும் புரத அங்கீகாரம் ஆகியவற்றிற்கு அமைப்பு

பெயரிடப்பட்ட நிறுவன அங்கீகார மாதிரிகளைப் பயன்படுத்துகிறது. பின்னர் கணினி இரசாயன - நோய் தொடர்பு பிரித்தெடுத்தல் மற்றும் இரசாயன - புரத உறவு பிரித்தெடுத்தல் மாதிரிகளைப் பயன்படுத்துகிறது. CORD-19 தரவுத்தொகுப்பில் இருந்து அமைப்புகளையும் உறவுகளையும் கணினி பிரித்தெடுக்கிறது. அமைப்பு பின்னர் பிரித்தெடுக்கப்பட்ட உறவுகள் மற்றும் நிறுவனங்களுக்கான அறிவு வரைபடத்தை உருவாக்குகிறது. அனைத்து நிறுவனங்களின் உட்பொதிவுகளைப் பெறுவதற்கும், கொரோனா வைரஸ் தொடர்பான முக்கிய நோய்கள், இரசாயனங்கள் மற்றும் புரதங்களைப் பெறுவதற்கும் இந்த அறிவு வரைபடத்தில் பிரதிநிதித்துவக் கற்றல் அமைப்பு செய்கிறது.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT – ENGLISH	iv
	ABSTRACT – TAMIL	v
	LIST OF TABLES	ix
	LIST OF FIGURES	x
	LIST OF ABBREVIATIONS	xiii
1	INTRODUCTION	1
	1.1 PROBLEM STATEMENT	4
	1.2 OBJECTIVE	4
	1.3 ORGANIZATION OF THE THESIS	4
2	RELATED WORK	6
	2.1 CONTEXTUAL WORD EMBEDDING	6
	2.2 NAMED ENTITY RECOGNITION	7
	2.3 RELATION EXTRACTION	8
	2.4 KNOWLEDGE GRAPH	10
	2.5 REPRESENTATION LEARNING	11
	2.6 SUMMARY	11
3	SYSTEM DESIGN	12
	3.1 OVERALL ARCHITECTURE DIAGRAM	12
	3.2 PREPROCESSING MODULE	14
	3.3 FEATURE EXTRACTION MODULE	15
	3.4 NAMED ENTITY RECOGNITION MODULE	17
	3.5 RELATION EXTRACTION MODULE	20

	3.6 GRAPH CONSTRUCTION MODULE	23
	3.7 REPRESENTATION LEARNING MODULE	25
4	DATASET	29
	4.1 DATASET	29
	4.2 RESULTS	34
	4.2.1 PREPROCESSING	34
	4.2.2 FEATURE EXTRACTION	37
	4.2.3 NAMED ENTITY RECOGNITION	39
	4.2.4 RELATION EXTRACTION	43
	4.2.5 GRAPH CONSTRUCTION	46
	4.2.6 REPRESENTATION LEARNING	47
	4.3 HYPERPARAMETERS	52
	4.4 PERFORMANCE METRICS	53
	4.4.1 NAMED ENTITY RECOGNITION	53
	4.4.2 RELATION EXTRACTION	54
	4.4.3 SYSTEM	55
	4.4.4 REPRESENTATION LEARNING	56
	4.5 COMPARATIVE ANALYSIS	59
	4.5.1 NAMED ENTITY RECOGNITION	59
	4.5.2 RELATION EXTRACTION	61
	4.6 TEST CASES	62
5	CONCLUSION AND FUTURE WORK	65
	REFERENCES	66

LIST OF TABLES

TABLE	PAGE NO.
3.1 Example encoding of a sentence in BIO scheme	19
4.1 NCBI Disease dataset characteristics	29
4.2 CHEMDNER dataset characteristics	30
4.3 JNLPBA dataset characteristics	31
4.4 BC5CDR dataset characteristics	31
4.5 Available relations in CHEMPROT dataset	32
4.6 CHEMPROT dataset characteristics	33
4.7 CORD-19 dataset characteristics	34
4.8 Overall entities extracted from CORD-19	43
4.9 Overall relations extracted from CORD-19	46
4.10 Hyperparameters for different NER datasets	52
4.11 Hyperparameters for Relation Extraction	52
4.12 Named Entity Recognition evaluation metrics	53
4.13 Relation Extraction evaluation metrics	55
4.14 Evidence for certain highly COVID-19 related entities	56
4.15 Comparative Analysis for NER	60
4.16 Comparative Analysis for RE	61
4.17 Test cases of the system	62

LIST OF FIGURES

FIGURE	PAGE NO.
3.1 Complete Architecture Diagram	13
3.2 Preprocessing module design diagram	14
3.3 Pseudocode for Preprocessing module	15
3.4 Feature extraction module design diagram	16
3.5 Pseudocode for Feature Extraction module	17
3.6 Named Entity Recognition module design diagram	18
3.7 Pseudocode for BERT-BiLSTM-CRF model	19
3.8 Pseudocode for the training of NER models	20
3.9 Pseudocode for entities extraction from CORD-19	20
3.10 Relation Extraction module design diagram	21
3.11 Pseudocode for RE training	22
3.12 Pseudocode for Relation extraction from CORD-19	23
3.13 Graph Construction module design diagram	23
3.14 Pseudocode for Graph construction module	24
3.15 Representation Learning module design diagram	25
3.16 Pseudocode for Representation learning module	26
4.1 CORD-19 dataset with language column	34
4.2 Processed CORD-19 abstract	35
4.3 Sample high occurrence words	35
4.4 Updated SciBERT vocabulary size	36
4.5 Input features for BERT model	36
4.6 SciBERT fine-tuning output	37
4.7 Sub word tokenized sentence	37

4.8	Input feature id for SciBERT	38
4.9	Input feature mask for SciBERT	38
4.10	Format of output of the model	38
4.11	Contextualized embeddings from SciBERT	38
4.12	BERT-BiLSTM-CRF model code	39
4.13	BERT layer output	39
4.14	BiLSTM layer output	40
4.15	Fully Connected layer output	40
4.16	CRF Layer loss output	41
4.17	SGD Optimizer	41
4.18	Word Tokenizing sample sentence	42
4.19	Input IDs for sample sentence	42
4.20	Attention mask for sample sentence	42
4.21	Extracted entities from sample sentence	43
4.22	Chemicals and Diseases from sample BC5CDR row	43
4.23	Sample BC5CDR row after transformation	44
4.24	Input ids of sample BC5CDR row	44
4.25	Attention mask for input ids	44
4.26	Token type ids for input ids	45
4.27	Encoded and tensorized label	45
4.28	Fine tuning SciBERT for BC5CDR	45
4.29	Relation extraction model output	46
4.30	Final relations after filtering of entities	46
4.31	Portion of knowledge graph focused on COVID-19	48
4.32	Top Diseases related to COVID-19	49
4.33	Top Chemicals related to COVID-19	50

4.34	Top proteins related to COVID-19	51
4.35	NER metrics graph	54
4.36	RE metrics graph	55
4.37	Comparative Analysis of different NER models	60
4.38	Comparative Analysis of different RE models	61

LIST OF ABBREVIATIONS

BioNER	Biomedical Named Entity Recognition
BioRD	Biomedical Relation Detection
COVID-19	Coronavirus Disease 2019
CORD-19	COVID-19 Open Research Dataset
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short Term Memory
CRF	Conditional Random Field
SciBERT	A BERT model for scientific text
KG	Knowledge Graph
MLE	Maximum Likelihood Estimation
ELMo	Embeddings from Language Model
MLM	Masked Language Modelling
DNA	Deoxyribonucleic acid
RNA	Ribonucleic acid
FFNN	Feed Forward Neural Networks
RNN	Recurrent Neural Networks
CNN	Convolution Neural Networks
SVM	Support Vector Machine
TF-IDF	Term Frequency – Inverse Document Frequency
LDA	Latent Dirichlet Allocation
BioBERT	Bidirectional Encoder Representations from Transformers for Biomedical Text Mining
NER	Named Entity Recognition
RE	Relation Extraction

TransE	Translating Embeddings
NCBI	National Center for Biotechnology Information
CHEMDNER	Chemical compound and Drug Name Recognition
JNLPBA	Joint Workshop on Natural Language Processing in Biomedicine and its Applications
BIO	Beginning Inside Outside
BC5CDR	BioCreative 5 Chemical Disease Relation
CHEMPROT	Chemical Protein interaction
SACEM	Structure-associated Chemical Entity Mention
GENIA	Genome Information Acquisition
MEDLINE	Medical Literature Analysis and Retrieval System Online
MeSH	Medical Subject Headings
Sars-COV-2	severe acute respiratory syndrome coronavirus 2
SGD	Stochastic Gradient Descent
NLTK	Natural Language Toolkit
CPR	Chemical Protein Relation
CID	Chemical Induced Disease
HIV	Human immunodeficiency virus infection
ARDS	Acute Respiratory Distress Syndrome
CRD	Chronic Respiratory Disease
HSV-2	herpes simplex virus – 2
PGE2	Prostaglandin E2
SER	Serine C ₃ H ₇ NO ₃
IL-1 β	Interleukin 1 beta
OTF-2	octamer transcription factor 2
[125I] T3	125I-triiodothyroacetic acid

CHAPTER 1

INTRODUCTION

With the exploding volume of data that has become available in the form of unstructured text articles, Biomedical Named Entity Recognition (BioNER) and Biomedical Relation Detection (BioRD) are becoming increasingly important for biomedical research. Currently, there are over 30 million publications in PubMed and over 25 million references in Medline. This amount makes it difficult to keep up with the literature even in more specialized fields. For this reason, the usage of BioNER and BioRD for tagging entities and extracting associations is indispensable for biomedical text mining and knowledge extraction.

Graphs are practical resources for many real-world applications. They have been used in social network mining to classify nodes and create recommendation systems. They have also been used in natural language processing to interpret simple questions and use relational information to provide answers. In a biomedical setting, graphs have been used to prioritize drugs relevant to disease, perform drug repurposing and identify drug-target interactions.

Within a biomedical setting, some graphs can be considered knowledge graphs; although, precisely defining a knowledge graph is difficult because there are multiple conflicting definitions.

The paper defines a biomedical knowledge graph as the following: a resource that integrates one or more expert-derived sources of information into a graph where nodes represent biomedical entities and edges represent relationships between two entities. This definition is consistent with other definitions found in the literature.

Often relationships are considered unidirectional (e.g., a compound treats a disease, but a disease cannot treat a compound); however, there are cases where relationships can be considered bidirectional (e.g., a compound resembles another compound, or a protein interacts with a chemical).

Knowledge graphs can be constructed in many ways using resources such as pre-existing databases or text. Usually, knowledge graphs are constructed using pre-existing databases. These databases are constructed by domain experts using approaches ranging from manual curation to automated techniques, such as text mining. Manual curation is a time-consuming process that requires domain experts to read papers and annotate sentences that assert a relationship. Automated approaches rely on machine learning or natural language processing techniques to rapidly detect sentences of interest. The automated approaches are categorized into the following groups: rule-based extraction, unsupervised machine learning, and supervised machine learning and discuss examples of each type of approach while synthesizing their strengths and weaknesses.

The proposed system considers the Automated way of extracting knowledge graphs from text using Deep Learning and Natural Language Processing techniques. The advantages of this approach include quick results and not needing ground truth information. The disadvantages include not having accurate and exact results. Since some entities may be missing or wrongly classified.

COVID-19 is a global epidemic with a considerable fatality rate and a high transmission rate, affecting millions of people world-wide since its outbreak. The search for treatments and possible cures for the novel Coronavirus has led to an exponential increase in scientific publications, but the challenge lies in effectively processing, integrating and leveraging related sources of information.

Scientific publications regarding COVID-19 contain various data about related diseases, proteins, chemicals and so on. The data in such publications are vastly unstructured. Most of the articles published under the title COVID-19 are gathered under the name of CORD-19. The paper introduce a fully automated generic pipeline consisting of an Information Extraction (IE) system followed by Knowledge Graph construction.

The proposed system creates Named Entity Recognition model using BERT-BiLSTM-CRF. Then this model is trained on multiple datasets and multiple models are generated. Then these models are used to recognize diseases, proteins and chemicals in the prediction dataset. The Relation Extraction models are created using SciBERT with linear classifier. Then this model is trained on multiple datasets and multiple models are created. Then these models are used to extract relations such as Chemical – Protein relation and Chemical – Disease relation. Once the entities and relations are extracted from the prediction dataset, the system create the Knowledge graph with the entities as nodes and relations as edges.

After the Knowledge Graph has been created, then the knowledge graph is represented as a table format. Then the TransD knowledge graph embedding method is trained using this knowledge graph. It uses dynamic mapping matrix between entities to refine the embedding values. After training, the model emits the embeddings for each entity. Then these embeddings are compared with the coronavirus embedding to find the top covid related diseases, chemicals and proteins. The similarity between two embeddings is found using the cosine similarity method.

1.1 PROBLEM STATEMENT

The project focuses on the problem of Information Extraction (IE) from the unstructured biomedical text articles. The task of Information Extraction is split into two subtasks regarding the problem in hand such as Named Entity Recognition and Relation Extraction. The system also focuses on the conversion of these unstructured biomedical text articles into well-structured Knowledge Graph. The project also focuses on using the extracted Knowledge Graph into one of many possible downstream tasks to demonstrate the use case of the extracted Knowledge Graph.

1.2 OBJECTIVE

The objective of the system is to extract information regarding COVID-19 from CORD-19 in a fully autonomous way using NLP language models and techniques. Initially the named entities such as diseases, proteins and chemicals from the CORD-19 dataset are gathered using the BERT-BiLSTM-CRF models. Then the relations between entities (Chemical – Protein relation and Chemical – Disease relation) are extracted using Transformers based Language models (SciBERT). Then the entities and relations are organized in the form of a biomedical knowledge graph using which research can be done. As an example of downstream task, The system embeds the knowledge graph using Knowledge Graph embedding techniques and finding the vectors of the entities in the knowledge graph. The found vectors are used to find similarity between the entities and COVID-19. Based on similarity scores, the top entities related to COVID-19 is found and is presented.

1.3 ORGANIZATION OF THE THESIS

The Thesis is separated into five chapters. Chapter 1 contains the Introduction, problem statement and the objectives of the system. Chapter 2 contains all the related works relevant to the system and also summary of all the issues found in the existing

systems. Chapter 3 contains the complete design of the system and also detailed description of all the modules in the system. Chapter 4 contains the dataset description, results of the system, performance measures, comparative analysis and test cases of the system. Chapter 5 concludes the thesis with conclusion and also provides guidelines for future works.

CHAPTER 2

RELATED WORK

Chapter 2 contains all the related works relevant to the system in use such as Named Entity Recognition, Relation Extraction, Knowledge Graph and Representation Learning

2.1 CONTEXTUAL WORD EMBEDDING

Pre-training contextual embeddings can be divided into either unsupervised methods(e.g. language modelling and its variants) or supervised methods (e.g. machine translation and natural language inference). The prototypical way to learn distributed token embeddings is via language modelling. A language model is a probability distribution over a sequence of tokens. Language modelling uses maximum likelihood estimation (MLE), often penalized with regularization terms, to estimate model parameters. A left-to-right language model takes the left context, t_1, t_2, \dots, t_{i-1} , of t_i into account for estimating the conditional probability. Language models are usually trained using large-scale unlabelled corpora. The conditional probabilities are most commonly learned using neural networks (Bengio et al., 2003) [27], and the learned representations have been proven to be transferable to downstream natural language understanding tasks.

The ELMo model (Peters et al., 2018) [20] generalizes traditional word embeddings by extracting context-dependent representations from a bidirectional language model. A forward L-layer LSTM and a backward L-layer LSTM are applied to encode the left and right contexts, respectively. At each layer j , the contextualized representations are the concatenation of the left-to-right and right-to-

left representations, obtaining N hidden representations, $(h_{1,j}, h_{2,j}, \dots, h_{N,j})$, for a sequence of length N .

BERT proposes a masked language modelling (MLM) objective, where some of the tokens of a input sequence are randomly masked, and the objective is to predict these masked positions taking the corrupted sequence as input. BERT applies a Transformer encoder to attend to bi-directional contexts during pre-training.

2.2 NAMED ENTITY RECOGNITION

Biomedical Named Entity Recognition (BNER) is the task of identifying biomedical instances such as chemical compounds, genes, proteins, viruses, disorders, DNAs and RNAs. The key challenge behind BNER lies in the methods that would be used for extracting such entities. They are done in multiple ways as follows,

Dictionary-based methods use large databases of named-entities and possibly trigger terms of different categories as a reference to locate and tag entities in a given text. One prominent example of a dictionary-based BioNER model is in the association mining tool Polysearch (Cheng et al., 2008) [2]. Another example is Whatizit (Rebholz-Schuhmann, 2013) [23], a class-specific text annotator tool available online, with separate modules for different NE types.

Currently, the most frequently used methods for named entity recognition are machine learning approaches. The first supervised machine learning methods used were Support Vector Machines (Kazama et al., 2002) [14], Hidden Markov models (Shen et al., 2003) [4], Decision trees, and Naive Bayesian methods (Nobata et al., 1999) [3]. However, the milestone publication by Lafferty et al. [11] (2001) about Conditional Random Fields (CRF) taking the probability of contextual dependency

of words into account shifted the focus away from independence assumptions made in Bayesian inference and directed graphical models.

In the last 5 years, there is a shift in the literature toward general deep neural network models. For instance, feedforward neural networks (FFNN) (Furrer et al., 2019), recurrent neural networks (RNN), or convolution neural networks (CNN) (Zhu et al., 2017) [30] have been used for BioNER systems. Among these, frequent variations of RNNs are, e.g., Elman-type, Jordan Type, unidirectional, or bidirectional models (Li et al., 2015c) [26].

For achieving the best results, Bi-LSTM and CRFs models are combined with a word-level and character-level embedding in a structure. (Habibi et al., 2017 [25]; Wang et al., 2018a [19]; Giorgi and Bader, 2019 [8]). Here a pre-trained lookup table produces word embeddings, and a separate Bi-LSTM for each word sequence renders a character-level embedding, both of which are then combined to acquire x_1 , x_2 , ..., x_n as word representation (Habibi et al., 2017) [25].

Currently, Transformer based models such BioBERT and SciBERT are fine tuned for BioNER.

2.3 RELATION EXTRACTION

After BioNER, the identification of associations between the named entities follows. For establishing such associations, the majority of studies use one of the following techniques.

In co-occurrence-based approaches, the hypothesis is that the more frequent two entities occur together, the higher the probability that they are associated with each other. In an extension of this approach, a relationship is deemed to exist

between two (or more) entities if they share an association with a third entity acting as a reciprocal link (Percha et al., 2012) [22].

In a rule-based approach, the relationship extraction depends highly on the syntactic and semantic analysis of sentences. For instance, in Fundel et al. (2006) [7], the authors explain how syntactic parse trees can be used to break sentences into the form *NounPhase1 – AssociationVerb – NounPhrase2*, where the noun phrases are biomedical entities associated through an association verb, and therefore indicates a relationship.

The most commonly used machine learning approaches use an annotated corpus with pre-identified relations as training data to learn a model (supervised learning). Previously, the biggest obstacle for using such machine learning approaches for relation detection was acquiring the labeled training and testing data. However, data sets generated through biomedical text mining competitions such as BioCreative and BioNLP have moderated this problem significantly.

One of the earliest studies using an SVM. In contrast to this, the latter study used a similar SVM model, however, for identifying the polarity of food-disease associations. In Jensen et al. (2014) [12], a Naive-Bayes classifier has been used for identifying food-phytochemical and food-disease associations based on TF-IDF (term frequency-inverse document frequency) features. Whereas, in Quan and Ren (2014), a Max-entropy based classifier with Latent Dirichlet Allocation (LDA) was used for inferring gene-disease associations, and a CRF was used for both NER and relation detection, for identifying disease-treatment and gene-disease associations.

Due to the state-of-the-art performance and less need for complicated feature processing, deep learning (DL) methods are becoming increasingly popular for relation extraction in the last five years. The most commonly used DL approaches

include convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrids of CNN and RNN (Jettakul et al., 2019 [13]). The feature inputs to DL models may include sentence level, word-level, and lexical-level features represented as vectors (Zeng et al., 2014) [28], positions of the related entities, and the class label of the relation type.

Recently, the Transformer based models such as BioBERT and SciBERT are used here.

2.4 KNOWLEDGE GRAPH

In the paper PharmKG: a dedicated knowledge graph benchmark for biomedical data mining briefings in bioinformatics [29], they extract knowledge graphs from drugs, diseases and genes databases and also their relations. Its advantage is that a large knowledge graph is obtained. But the data is highly generic and only takes structured data as input.

In the paper COVID-19 Knowledge Graph: a computable, multimodal, cause-and-effect knowledge model of COVID-19 pathophysiology [5], their evidence text from the prioritized corpus was manually encoded as a triple (source-relation-target). Its advantage is that extracted information is mostly correct apart from Human errors. But only a small knowledge graph is obtained and manual curation is time-consuming.

In the paper Extraction and Representation of Financial Entities from Text [24], they extract knowledge graphs from financial text corpus using NER and RE tasks. Its advantage is that finding financial entities is comparatively easier with rule-based approaches.

In the paper Deep Learning-based Knowledge Graph Generation for COVID-19 [16], they find entities related to COVID-19 from dictionaries and extract their relations from text corpus. Its advantage is that the Unsupervised method is devised to find entities and relations. But results vary massively in unsupervised methods.

2.5 REPRESENTATION LEARNING

The large-scale knowledge graph is also faced with a serious problem of data sparsity, which makes the calculation of semantic or inferential relations of entities extremely inaccurate. To counter this problem, knowledge graph embedding has been developed as it has the capability of providing dense and low-dimensional feature space and helps in efficiently calculating the semantic relations between entities with low computational quality.

Translation based models is based on the idea that a triple (**head, relation, tail**) can be represented as a geometric principle such as $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ (TransE) [9].

Tensor Factorization-Based Models is based on the idea that all triples can be transformed to 3D binary tensor and this tensor is converted into entities and relations embeddings using Dimensionality Reduction.

2.6 SUMMARY

Some of the problems that we found in the analysis of literature survey are Manual curation of entities and relations takes very long time and the results count will be low, Lack of ground truth information for correct and easy evaluation which results in lower quality knowledge graphs, Rule based information extraction requires manual rule setting which differs based on the information we need, Usage of External Knowledge Base which may not be available for the concerned task and Lack of structured data which will be easier to extract.

CHAPTER 3

SYSTEM DESIGN

Chapter 3 contains the description of the system architecture. It also contains the detailed description of all the modules in the system along with module design diagram and pseudocode.

3.1 OVERALL ARCHITECTURE DIAGRAM

The overall architecture diagram for the proposed system is shown in figure 3.1. The proposed work is split into 6 modules namely, Preprocessing, Feature extraction, Named Entity Recognition, Relation Extraction, Graph construction and Representation learning. The final result is the COVID-19 knowledge graph.

In the Preprocessing module, the COVID-19 abstracts are taken and is transformed to input to the Language Model and the BERT model is finetuned to get better results for entities extraction. In the Feature extraction module, the Named Entity Recognition datasets are passed through to the BERT model to get the features from these datasets. In the Named Entity Recognition module, the BERT-BiLSTM-CRF model is used to train on the NER datasets and the models are used to predict on COVID-19. In the Relation Extraction module, the Relations datasets are used to train the BERT model and these models are used to predict whether a relation exists between the extracted entities from the COVID-19. In the Graph Construction module, the Knowledge graph is constructed. In the Representation Learning module, the KG is used to train the TransD model to find embeddings of entities which are used to find similarity with COVID-19. Figure 3.1 shows the Complete Architecture diagram.

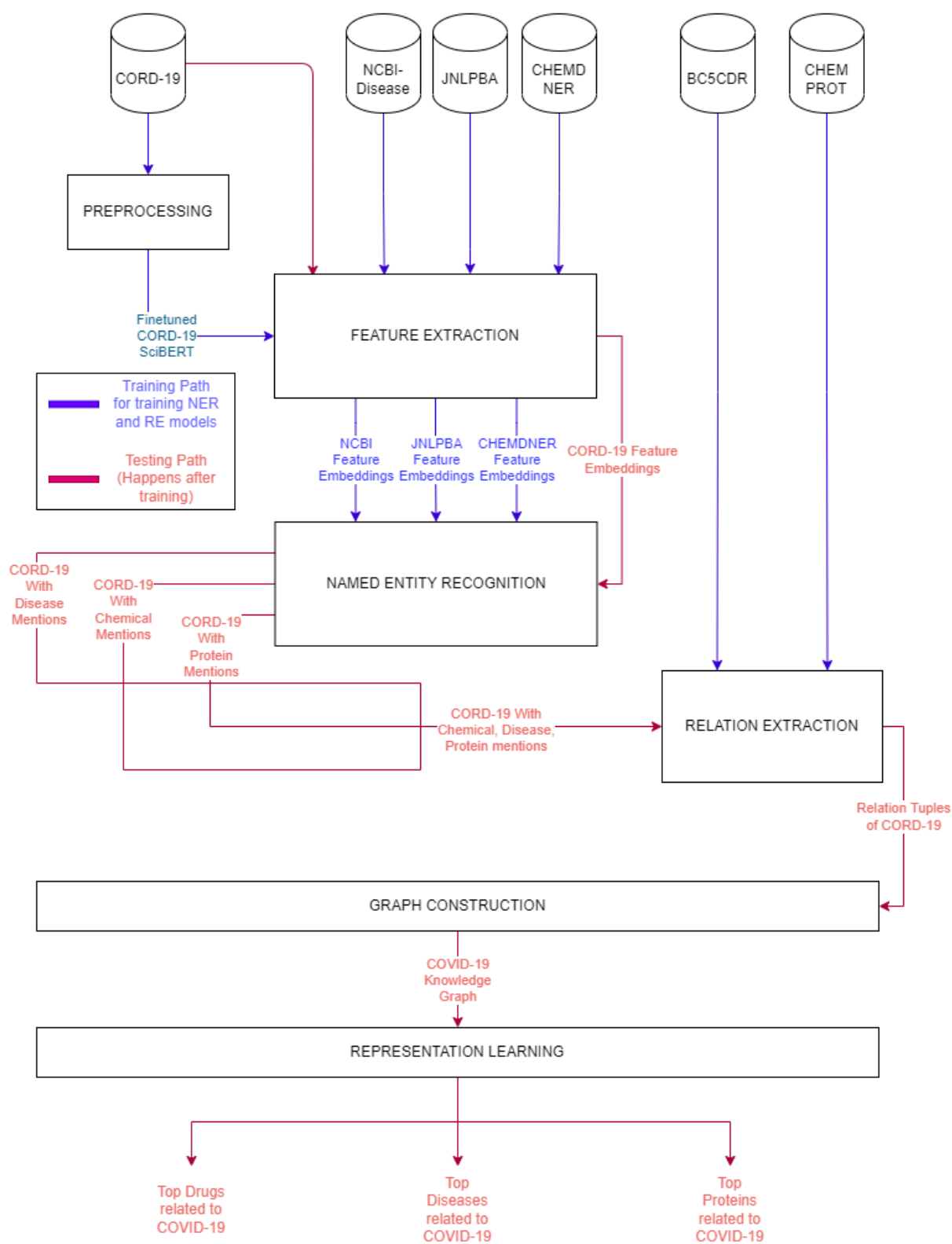


Figure 3.1 Complete Architecture Diagram

3.2 PREPROCESSING MODULE

Figure 3.2 shows the complete design of the preprocessing module. In this module, initially the CORD-19 dataset is taken and is filtered by language using python package and only English text is considered afterwards. Then the title and abstracts are loaded from the file system. Then only the CORD-19 abstracts are taken. There are about 350,000 abstracts. They are all cleaned and the final version contains only letters, numbers and some symbols. Then all the abstracts are word tokenized and stored in an array. Then the occurrence count of all the words is found. Then the words with occurrence count greater than 450 are taken. There are about 6000 words in this stage.

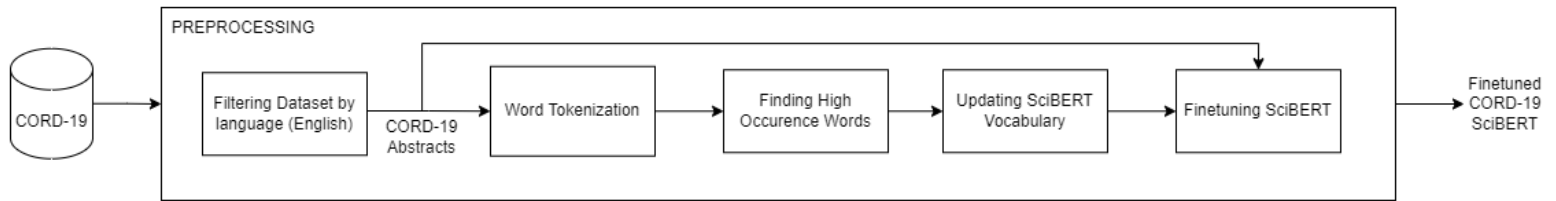


Figure 3.2 Preprocessing module design diagram

Then the **SciBERT** model is taken and the vocabulary of the model is updated. Initially there are 31090 words in the SciBERT vocabulary. After updating the vocabulary, the new vocabulary size is 32056. Almost 1000 new words are added to the SciBERT model. This updated vocabulary leads to updated sub word tokenizer which will be used for fine tuning. Then the CORD-19 abstracts are taken and are sub word tokenized using the new tokenizer and the sub words are converted into input ids and are padded to the length of 256. This input ids are used to finetune the SciBERT model and the final CORD-19 SciBERT model is obtained which will be used for Feature extraction.

The module takes the **CORD-19 dataset** as the input. The dataset consists of files related to COVID-19. The module returns the **finetuned SciBERT model** that contains updated vocabulary and updated weights according to CORD-19 dataset. Figure 3.3 shows the pseudocode for the preprocessing module.

```
def Preprocessing(cord_19_dataset):
    dataset = cord_19_dataset.where(language == "english")
    words = tokenize(dataset["abstracts"])
    counter = Counter(words)
    scibert = BERT("SciBERT")
    for word in words:
        if counter[word] > 450:
            scibert.update_vocabulary(word)
    return scibert.train(dataset["abstracts"], epochs=5)
```

Figure 3.3 Pseudocode for Preprocessing module

3.3 FEATURE EXTRACTION MODULE

Figure 3.4 shows the complete design diagram of Feature extraction module. Here the input dataset is first loaded. In the case of CORD-19, the title, abstract and full-text are all loaded. The dataset is loaded in the form of sentences and their NER tags. Then the dataset is sentence tokenized. Here the custom tokenizer obtained from preprocessing is used to tokenize individual sentences. Here the special token [CLS] is added to the start of the sentence and [SEP] is added to the end of the sentence. The tokenizer basically tokenizes common words as individual tokens and more rare words into meaningful sub tokens. Then the sentences are truncated or padded to accommodate the max length (256) for the BERT model. Then the mask is defined such that only words are marked as 1 and padded words are marked otherwise.

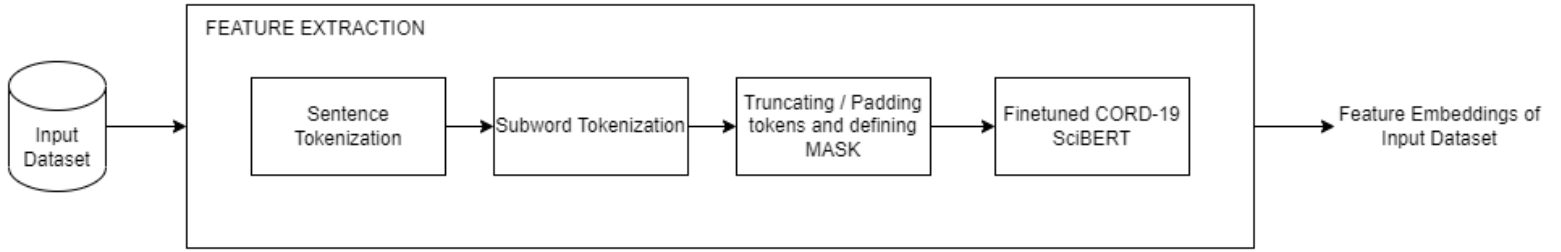


Figure 3.4 Feature extraction module design diagram

Finally, the individual sentence of size (256) is passed to the fine-tuned CORD-19 SciBERT model and the last hidden layer output is taken from the BERT model which is of the dimension 768. The output produced by the BERT model is of the shape (1,256,768). Here the 1 denotes the number of input sentences. The 256 denotes the sequence length. Each sub word in the sentence is represented by a vector of 768 dimension which depends not only on the sub word but also the surrounding words. Hence this type of embeddings is called Contextualized word embedding. This output is passed onto the Named Entity Recognition layers which detect the entities based on the word embedding of words in the sentence.

The module takes the **finetuned CORD-19 SciBERT model** as the input which is used to produce the feature embeddings. The module returns the **contextualized word embeddings of the input dataset** regardless of whether it is a NER dataset or CORD-19 dataset. Figure 3.5 shows the pseudo code for the feature extraction module.

```

def FeatureExtraction(dataset):
    data = read(dataset)
    sentences = sentence_tokenize(data)
    scibert = BERT("SciBERT")
    embeddings = []
    for sentence in sentences:
        tokens = scibert.subword_tokenize(sentence)
        tokens, mask = padOrTruncate(tokens, 256)
        embeddings.append(scibert(tokens, mask))
    return embeddings

```

Figure 3.5 Pseudocode for the Feature Extraction module

3.4 NAMED ENTITY RECOGNITION MODULE

Figure 3.6 shows the complete architecture diagram of the Named Entity Recognition module. In this module, the NCBI-Disease dataset is used for recognition of diseases. CHEMDNER dataset is used for recognition of Chemicals. JNLPBA dataset is used for recognition of Proteins.

The individual sentence features such as input_ids, attention_mask, token_type_ids are taken and are passed to the CORD-19 SciBERT. Then the final hidden layer of size 768 of each token in the sentence is taken as the contextualized word embedding and is passed to the Bidirectional LSTM layer and the output is of size 1024 (512 for forward LSTM and 512 for backward LSTM). Then dropout regularisation of 30% is applied to avoid overfitting. Then a fully connected layer that takes the input of size 1024 and outputs the vector of size equal to the total number of labels in the training dataset. Then the vector is given to the Conditional Random Field (CRF) layer which learns the transitional probabilities from the input dataset and finds the best possible tag sequence for the sentence. The model uses

Negative log-likelihood loss to optimise the model. It is modelled as a minimization problem.

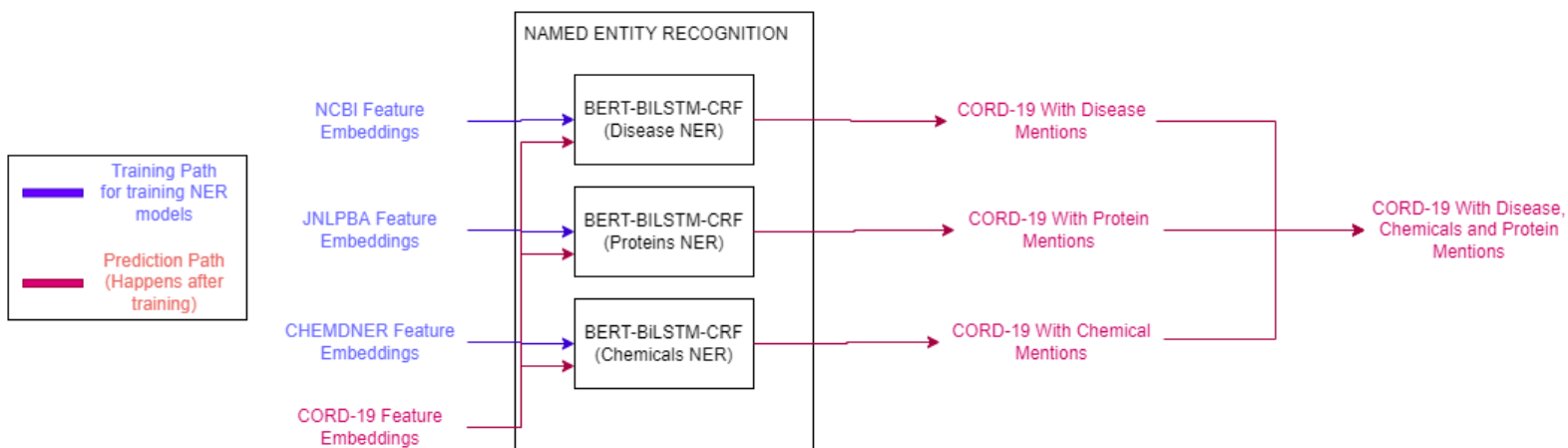


Figure 3.6 Named Entity Recognition module design diagram

Now there are 3 models, which are capable of finding diseases, drugs and proteins respectively. Now the Processed CORD-19 is fed into each model and the entity tags of CORD-19 dataset are found. The tags are for each word token that are encoded in BIO Scheme. Here B-Entity refers to the beginning of the entity, I-Entity refers to the inside of the entity and O refers to the outside of the entity. Table 3.1 shows sample encoding of a sentence in BIO scheme. These Tags are combined so that the final output contains CORD-19 dataset with all the diseases, chemicals and proteins mentioned.

The module takes the **contextualized word embeddings** of the NER training datasets and CORD-19 dataset as input. Then the module returns the **extracted diseases, chemicals and proteins** from the CORD-19 dataset as output. Figure 3.7, Figure 3.8 and Figure 3.9 shows the pseudocode for the BERT-BiLSTM-CRF model, training of the NER models on input dataset and extraction of entities from the CORD-19 dataset respectively.

Table 3.1 Example encoding of a sentence in BIO scheme

Token	Encoding in BIO
Management	O
of	O
critically	O
ill	O
patients	O
with	O
Severe	B-Disease
Acute	I-Disease
Respiratory	I-Disease
Syndrome	I-Disease
.	O

```
def BERT_BiLSTM_CRF(sentence, tags):
    cord_scibert = BERT("finetuned-cord-scibert")
    embeddings = cord_scibert(sentence)
    lstm_output = BiLSTM(embeddings, 512)
    dropout_output = Dropout(lstm_output)
    linear_output = Linear(dropout_output, len(dataset.labels))
    tag, loss = CRF(linear_output, tags)
    return tag, loss
```

Figure 3.7 Pseudocode for the BERT-BiLSTM-CRF model

```

def NER_Training(dataset):
    model = BERT_BiLSTM_CRF()
    for data in dataset:
        sentence, original_tag = data
        tag, loss = model(sentence, original_tag)
        loss.backward()
    return loss

```

Figure 3.8 Pseudocode for the training of NER models

```

def Extract_Entities_From_CORD_19(cord_19_dataset):
    diseaseNER = NER_Training(ncbi_dataset)
    chemicalNER = NER_Training(chemdner_dataset)
    proteinNER = NER_Training(jnlpba_dataset)
    entities = []
    for sentence in cord_19_dataset:
        disease_tag_sequence = diseaseNER(sentence)
        chemical_tag_sequence = chemicalNER(sentence)
        protein_tag_sequence = proteinNER(sentence)
        sentence_entities = get_entities(
            disease_tag_sequence,
            chemical_tag_sequence,
            protein_tag_sequence
        )
        entities += sentence_entities
    return entities

```

Figure 3.9 Pseudocode for entities extraction from CORD-19

3.5 RELATION EXTRACTION MODULE

Figure 3.10 shows the complete design diagram of the Relation extraction module. In this module, BC5CDR dataset is used for extraction of chemical induced disease relations. CHEMPROT dataset is used for extraction of chemical-protein relations. The 2 datasets are preprocessed where the tokens are associated with its

entities, and each sentence in the dataset is modified into the 2 sentences where the first sentence contains the two entities whose relation is in question separated by a space and the second sentence contains the actual sentence from the dataset.

Example transformation,

Epidermal growth factor receptor gefitinib\t. Epidermal growth factor receptor inhibitors currently under investigation include the small molecules gefitinib (Iressa, ZD1839) and erlotinib (Tarceva, OSI-774), as well as monoclonal antibodies such as cetuximab (IMC-225, Erbitux).

The input_ids, attention_mask and token_type_ids of the input sentence/text is taken. The sentence/text is passed into the SciBERT and the output embedding is taken. The token embedding of the special token “[CLS]” is used for classification of the sentence/text to one of the relations. The 768-dimensional vector of “[CLS]” is then passed into the fully connected layer which takes this as input and produces output of size equal to the total number of types of relations in the dataset. AdamW optimizer is used to compile the model.

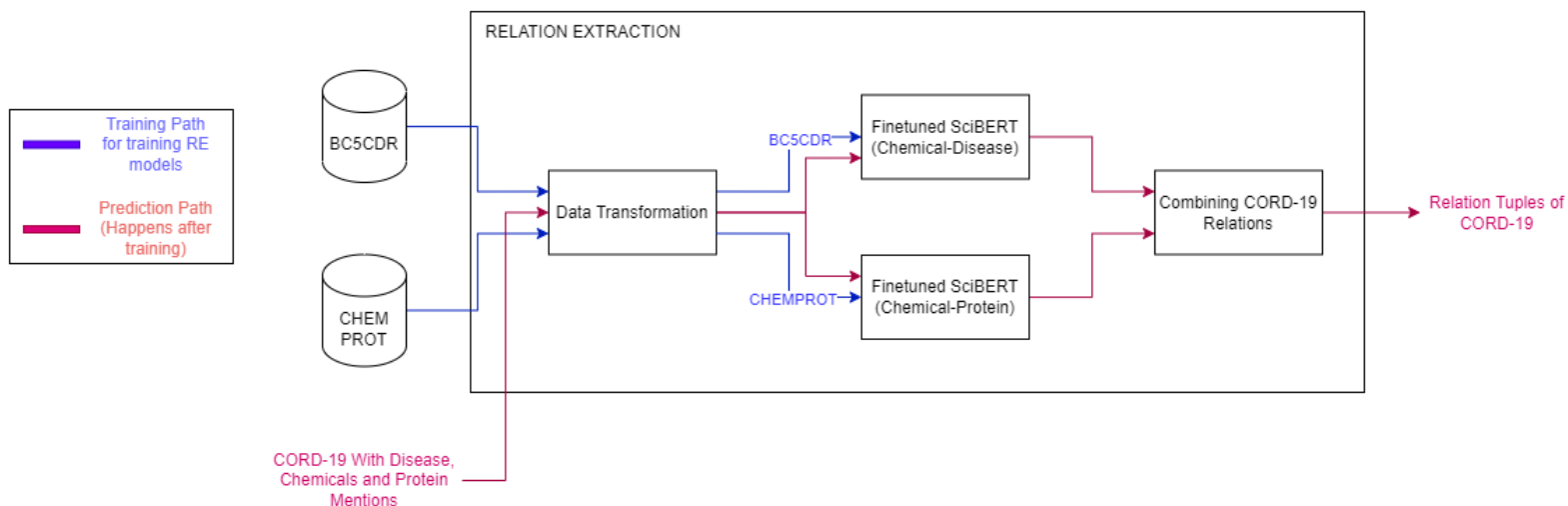


Figure 3.10 Relation extraction module design diagram

The BC5CDR dataset is fed into the SciBERT model which produces the Chemical Induced Disease Relation. The CHEMPROT dataset is fed into an individual SciBERT model which will be finetuned for finding relations between Chemicals and Proteins. Now the CORD-19 dataset with entity mentions is preprocessed where only the sentences with two or more entities are forwarded into the model. The cell line, cell type entities are not considered for this count. And based on the type of entities, the sentence is fed to one of two models and the model predicts whether a relation exists between two entities or not. Finally, the two models' outputs are combined and tuples are generated of the form (Entity1, Entity2, Relation).

The module takes the **relation datasets** as input to train the SciBERT models. The module returns the **relations between entities** extracted from the CORD-19 dataset. Figure 3.11 and Figure 3.12 shows the pseudocode for the RE training and relation extraction from CORD-19 respectively.

```
def RE_Training(dataset, number_of_relations):
    bert_model = BERT("SciBERT")
    for row in dataset:
        text = row.entity_1+" "+row.entity_2+"\t."+row.text
        features = bert_model.sub_word_tokenizer(text)
        bert_output = bert_model(features)
        cls_token_output=bert_output["[CLS]"]
        pred_relation = Linear(cls_token_output, number_of_relations)
        loss = CrossEntropyLoss(pred_relation, row.relation)
        loss.backpropagate()
    return bert_model
```

Figure 3.11 Pseudocode for RE training

```

def RelationExtraction(cord_19_with_entities):
    chem_dis_re = RE_Training(bc5cdr, len(bc5cdr.relations))
    chem_prot_re = RE_Training(chemprot, len(chemprot.relations))
    relations = []
    for sentence, entity_1, entity_2 in cord_with_entities:
        relations.append(chem_dis_re(sentence, entity_1, entity_2))
        relations.append(chem_prot_re(sentence, entity_1, entity_2))
    return relations

```

Figure 3.12 Pseudocode for Relation extraction from CORD-19

3.6 GRAPH CONSTRUCTION MODULE

Figure 3.13 shows the complete design diagram of Graph construction module. The module takes the relation types in which each tuple is of the type (“Entity_1, Relation, Entity_2”). Then from the tuples the occurrence count of all the entities is found. Then only the tuples with either of the entities having occurrence count greater than 5 are considered for the next step. This helps in reducing the noise in the resultant Knowledge Graph.

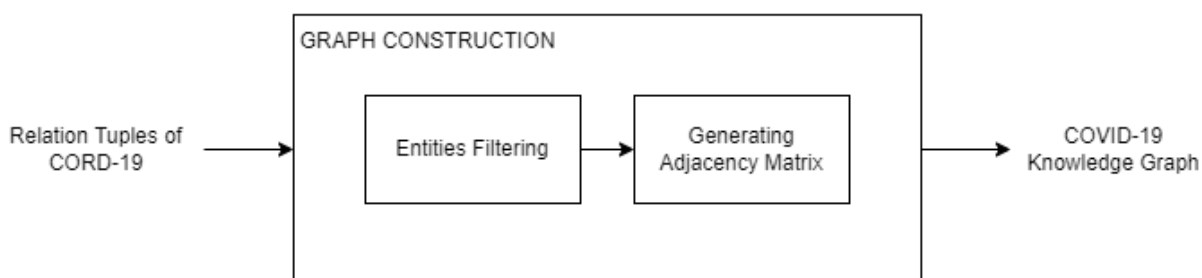


Figure 3.13 Graph Construction module design diagram

Then a Neo4j Graph Database instance is created and all the entities from the filtered tuples are created as nodes with names and their types. Then the relations are created as directed edges from Chemical-to-Disease for Chemical Induced Disease relation and Chemical-to-Protein for Chemical Protein relation.

Mathematically, the Knowledge Graph is defined as follows,

E : A set of nodes representing disease / protein / chemical.

R : A set of labels representing chemical-disease relation and chemical-protein relation.

$G \subseteq E \times R \times E$: A set of edges that represent facts connecting entity pairs.

The model takes the **relation tuples** extracted from the CORD-19 dataset as input. The model returns the **Knowledge Graph of COVID-19** which is stored in a Neo4j Graph Database. Figure 3.14 shows the pseudocode for the graph construction module.

```
def GraphConstruction(relation_tuples):
    all_entities = relation_tuples["entity_1"] + relation_tuples["entity_2"]
    entitiesCounter = Counter(all_entities)
    database = Neo4j("covid_knowledge_graph")
    valid_tuples = []

    for relation in relation_tuples:
        entity1C = entitiesCounter[relation["entity_1"]]
        entity2C = entitiesCounter[relation["entity_2"]]
        if entity1C >= 5 or entity2C >= 5:
            valid_tuples.append(relation)

    for relation in valid_tuples:
        database.add_node(relation["entity_1"])
        database.add_node(relation["entity_2"])
        database.add_edge(relation["entity_1"], relation["entity_2"])

    return database
```

Figure 3.14 Pseudocode for the Graph construction module

3.7 REPRESENTATION LEARNING MODULE

Figure 3.15 shows the complete design diagram of the Representation learning module. Initially the COVID-19 Knowledge Graph is taken and is converted into triplets (head entity, relation, tail entity). Since the Knowledge Graph contains only positive relations, negative relations are generated by taking two random entities and if there is no relation between them, then these two entities are taken as negative relation.

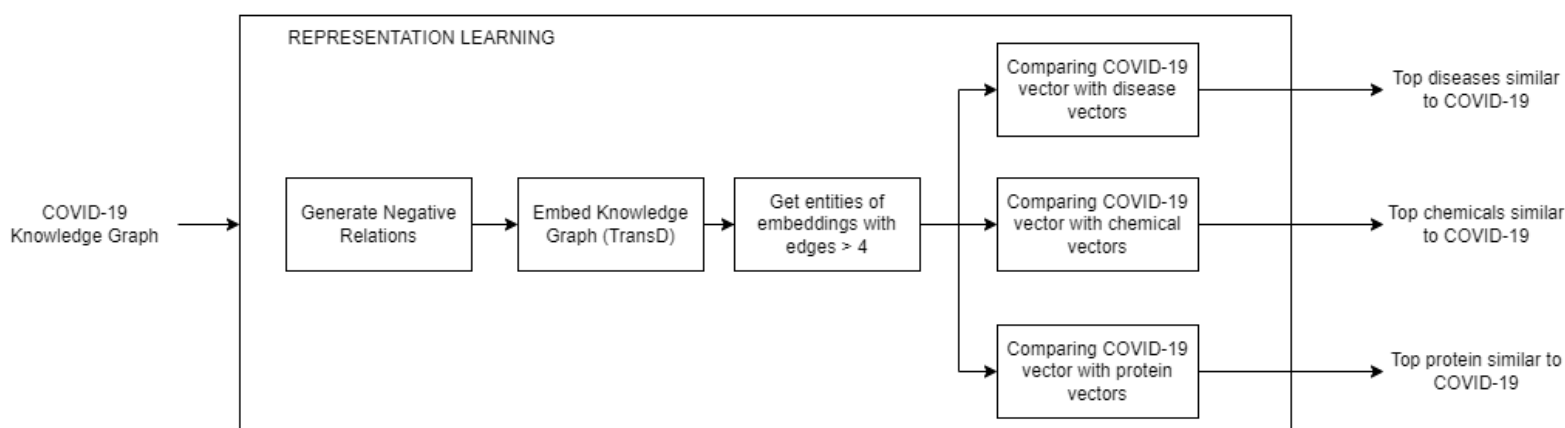


Figure 3.15 Representation Learning module design diagram

Then these triplets are passed to train the TransD model. In TransD model, each symbol (entities, relations) is represented by two vectors. The first one captures the meaning of entity / relation, the other one is used to construct mapping matrices. Once the TransD model is trained, the embeddings of all entities and relations are taken from the model parameters. Then only the embeddings of entities whose edges count is greater than or equal to 5 are taken for comparison. Then the COVID-19 embedding is taken from the model, and is compared with all the entities from above using cosine similarity score. Then the top 25 diseases / chemicals / proteins with highest cosine similarity score is taken and is produced as output.

The model takes the **COVID-19 Knowledge Graph** as input in the triplet's format. The model returns the list of **top 25 diseases / chemicals / proteins related to COVID-19** along with their cosine similarity score. Figure 3.16 shows the pseudocode for the representation learning module.

```
def RepresentationLearning(covid_19_knowledge_graph):
    relations = covid_19_knowledge_graph.to_relations()
    entitiesCounter = Counter(relations["entity_1"] + relations["entity_2"])
    neg_relations = []
    for i in range(len(relations)):
        e1, e2 = get_random_entities(relations["entity_1"], relations["entity_2"])
        neg_relations.append((e1, e2, "NEGATIVE"))
    model = TransD(relations + neg_relations)
    valid_emb = {"disease": [], "chemical" : [], "protein" : []}
    covid_emb = model("coronavirus")

    for entity in entitiesCounter:
        if entitiesCounter[entity] >= 5:
            valid_emb[entity.type].append(model(entity))

    diseases = sorted(cosine_similarity(valid_emb["disease"], covid_emb))
    chemicals = sorted(cosine_similarity(valid_emb["chemical"], covid_emb))
    proteins = sorted(cosine_similarity(valid_emb["protein"], covid_emb))
    return diseases[:25], chemicals[:25], proteins[:25]
```

Figure 3.16 Pseudocode for Representation learning module

CHAPTER 4

IMPLEMENTATION DETAILS AND RESULTS

Chapter 4 contains the description of all the datasets used in the system, the complete working of the system along with intermediate results and performance of the system. Then the performance is compared with existing systems and all the test cases of the system are also listed.

4.1 DATASET

NCBI-DISEASE

The NCBI disease corpus is fully annotated at the mention and concept level to serve as a research resource for the biomedical natural language processing community. Two-annotators are assigned per document (randomly paired) and annotations are checked for corpus-wide consistency of annotations. Table 4.1 shows the characteristics of the dataset. The available tags are **B-Disease**, **I-Disease**, **O**.

Table 4.1 NCBI Disease dataset characteristics

Corpus Characteristics	Training Set	Development Set	Test Set	Whole Corpus
PubMed citations	593	100	100	793
Total disease mentions	5145	787	960	6892
Unique disease mentions	1710	368	427	2136
Unique concept ID	670	176	203	790

CHEMDNER

The abstracts of the CHEMDNER corpus were selected to be representative for all major chemical disciplines. Each of the chemical entity mentions was manually labeled according to its structure-associated chemical entity mention (SACEM) class: abbreviation, family, formula, identifier, multiple, systematic and trivial. Table 4.2 shows the characteristics of the dataset. The available tags are **B-Chemical**, **I-Chemical** and **O**.

Table 4.2 CHEMDNER dataset characteristics

Corpus Characteristics	Training Set	Development Set	Test Set	Whole Corpus
Abstracts	3500	3500	3000	10000
Nr. Characters	4,883,753	4,864,558	4,199,068	13,947,379
Nr. Tokens	770,855	766,331	662,571	2,199,757
Abstracts with SACEM	2,916	2,907	2,478	8,301
Nr. Mentions	29,478	29,526	25,351	84,355
Nr. Chemicals	8,520	8,677	7,563	19,805
Nr. Journals	193	188	188	203

JNLPBA

The data came from the GENIA version 3.02 corpus (Kim et al., 2003). This was formed from a controlled search on MEDLINE using the MeSH terms human, blood cells and transcription factors. Table 4.3 denotes the characteristics of the

dataset. The available tags are **B-Protein, I-Protein, B-DNA, I-DNA, B-RNA, I-RNA, B-cell_line, I-cell_line, B-cell_type, I-cell_type, O.**

Table 4.3 JNLPBA dataset characteristics

Corpus Characteristics	Training Set	Test Set	Whole corpus
Abstracts	2000	404	2,404
Sentences	20,546	4,260	24,806
Words	472,006	96,780	568,786
Entities	51,291	8,662	59,953

BC5CDR

Chemicals, diseases, and their relations are among the most searched topics by PubMed users worldwide (1-3) as they play central roles in many areas of biomedical research and healthcare such as drug discovery and safety surveillance. Although the ultimate goal in drug discovery is to develop chemicals for therapeutics, recognition of adverse drug reactions between chemicals and diseases. Table 4.4 denotes the characteristics of the dataset. The only relation available is **Chemical-Induced Disease**.

Table 4.4 BC5CDR dataset characteristics

Corpus Characteristics	Training Set	Testing Set	Whole Corpus
No. of Chosen Abstracts	1,000	500	1,500
No. of Chemical Mentions	10,550	5,385	15,935

Chemical Unique Mentions	2,973	1,435	4,408
No. of Disease Mentions	8,426	4,424	12,850
Disease Unique Mentions	3,829	1,988	5,817
Chemical Induced Disease Relations	2,050	1,066	3,116

CHEMPROT

It is a manually annotated corpus, the CHEMPROT corpus, where domain experts have exhaustively labeled:(a) all chemical and gene mentions, and (b) all binary relationships between them corresponding to a specific set of biologically relevant relation types (CHEMPROT relation classes). The aim of the CHEMPROT track is to promote the development of systems able to extract chemical-protein interactions of relevance for precision medicine, drug discovery as well as basic biomedical research. Table 4.5 shows all the available relations in the CHEMPROT dataset and Table 4.6 shows the corpus characteristics of the dataset.

Table 4.5 Available relations in CHEMPROT dataset

Group	CHEMPROT relations belonging to this group
CPR:1	PART_OF
CPR:2	REGULATOR DIRECT_REGULATOR INDIRECT_REGULATOR
CPR:3	UPREGULATOR ACTIVATOR INDIRECT_UPREGULATOR
CPR:4	DOWNREGULATOR INHIBITOR INDIRECT_DOWNREGULATOR
CPR:5	AGONIST AGONIST-ACTIVATOR AGONIST-INHIBITOR

CPR:6	ANTAGONIST
CPR:7	MODULATOR MODULATOR-REGULATOR MODULATOR-INHIBITOR
CPR:8	COFACTOR
CPR:9	SUBSTRATE PRODUCT_OF SUBSTRATE_PRODUCT_OF
CPR:10	NOT

Table 4.6 CHEMPROT dataset characteristics

Corpus Characteristics	Training Set	Development Set	Test set	Whole Corpus
Document	1,020	612	800	2,432
Chemical	13,017	8,004	10,810	31,831
Protein	12,752	7,567	10,019	30,338
Positive Relation	4,157	2,416	3,458	10,031
Positive relation in one sentence	4,122	2,412	3,444	9,978

CORD-19

In response to the COVID-19 pandemic, the White House and a coalition of leading research groups have prepared the COVID-19 Open Research Dataset (CORD-19). CORD-19 is a resource of over 1,000,000 scholarly articles, including over 350,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses. This freely available dataset is provided to the global research

community to apply recent advances in AI techniques to generate new insights in support of the ongoing fight against this infectious disease. Table 4.7 shows the characteristics of the dataset.

Table 4.7 CORD-19 dataset characteristics

Subfield	Count	% Of Corpus
Virology	20,116	42.3%
Immunology	9,875	20.7%
Molecular biology	6,040	12.7%
Genetics	3,783	8.0%
Intensive care medicine	3,204	6.7%
Other	4,595	9.6%

4.2 RESULTS

4.2.1 PREPROCESSING

Initially the CORD-19 dataset articles are loaded along with their metadata. Then the language of each row in the dataset is found using the langdetect python package and is added as an additional column in the dataset. Figure 4.1 shows the CORD-19 dataset rows with language column.

metadata	title	abstract	body_text	lang
{'title': 'Dexmedetomidine improved renal func...	Dexmedetomidine improved renal function in pat...	Background: Dexmedetomidine has been reported ...	Dexmedetomidine is a sedative drug that has a ...	en
{'title': 'Aortic volume determines global end...	Aortic volume determines global end- diastolic...	Background: Global end-diastolic volume (GEDV)...	Transpulmonary thermomodulation is commonly used...	en
{'title': 'Whole genome sequencing and phyloge...	Whole genome sequencing and phylogenetic analy...	Background: Human metapneumovirus (HMPV) is an...	Human metapneumovirus (HMPV) is a single-stran...	en

Figure 4.1 CORD-19 dataset with language column

Then all the rows with languages other than English are removed, the dataset rows before and after filtering are 2,29,777 and 2,21,520 respectively. This helps in reducing the noise in the CORD-19 dataset with other language rows and also helps in increasing accuracy of predictions.

Then the abstracts from the CORD-19 dataset is taken and is cleaned of all punctuations and is word tokenized. Figure 4.2 shows the output after cleaning.

```
[ 'objective', 'this', 'retrospective', 'chart', 'review', 'describes', 'the', 'epidemiology', 'and', 'clinical', 'features', 'of', '40', 'patients', 'with', 'culture-proven', 'mycoplasma', 'pneumoniae', 'infections', 'at', 'king', 'abdulaziz', 'university', 'hospital', 'jeddah', 'saudi', 'arabia', 'methods', 'patients', 'with', 'positive', 'm', 'pneumoniae', 'cultures', 'from', 'respiratory', 'specimens', 'from', 'january', '1997', 'through', 'december', '1998', 'were', 'identified', 'through', 'the', 'microbiology', 'records', 'charts', 'of', 'patients', 'were', 'reviewed', 'results', '40', 'patients', 'were', 'identified', '33', '82', '5', 'of', 'whom', 'required', 'admission', 'most', 'infections', '92', '5', 'were', 'community-acquired', 'the', 'infection', 'affected', 'all', 'age', 'groups', 'but', 'was', 'most', 'common', 'in', 'infants', '32', '5', 'and', 'pre-school', 'children', '22', '5', 'it', 'occurred', 'year-round', 'but', 'was', 'most', 'common', 'in', 'the', 'fall', '35', 'and', 'spring', '30', 'more', 'than', 'three-quarters', 'of', 'patients', '77', '5', 'had', 'comorbidities', 'twenty-four', 'isolates', '60', 'were', 'associated', 'with', 'pneumonia', '14', '35', 'with', 'upper', 'respiratory', 'tract', 'infections', 'and', '2', '5', 'with', 'bronchiolitis', 'cough', '82', '5', 'fever', '75', 'and', 'malaise', '58', '8', 'were', 'the', 'most', 'common', 'symptoms', 'and', 'crepitations', '60', 'and', 'wheezes', '40', 'were', 'the', 'most', 'common', 'signs', 'most', 'patients', 'with', 'pneumonia', 'had', 'crepitations', '79', '2', 'but', 'only', '25', 'had', 'bronchial', 'breathing', 'immunocompromised', 'patients', 'were', 'more', 'likely', 'than', 'non-immunocompromised', 'patients', 'to', 'present', 'with', 'pneumonia', '8', '9', 'versus', '16', '31', 'p', '0', '05', 'of', 'the', '24', 'patients', 'with', 'pneumonia', '14', '58', '3', 'had', 'uneventful', 'recovery', '4', '16', '7', 'recovered', 'following', 'some', 'complications', '3', '12', '5', 'died', 'because', 'of', 'm', 'pneumoniae', 'infection', 'and', '3', '12', '5', 'died', 'due', 'to', 'underlying', 'comorbidities', 'the', '3', 'patients', 'who', 'died', 'of', 'm', 'pneumoniae', 'pneumonia', 'had', 'other', 'comorbidities', 'conclusion', 'our', 'results', 'were', 'similar', 'to', 'published', 'data', 'except', 'for', 'the', 'finding', 'that', 'infections', 'were', 'more', 'common', 'in', 'infants', 'and', 'preschool', 'children', 'and', 'that', 'the', 'mortality', 'rate', 'of', 'pneumonia', 'in', 'patients', 'with', 'comorbidities', 'was', 'high']
```

Figure 4.2 Processed CORD-19 abstract

Then the occurrence count of all words in the abstracts are found and the words with occurrence count greater than 450 will be taken. Figure 4.3 shows some of the highly used words in the CORD-19 abstracts.

```
The length of the vocabulary is 6968
['chart', 'describes', 'epidemiology', 'mycoplasma', 'pneumoniae', 'university', 'saudi', 'ology', 'records', 'charts', 'reviewed', '33', '82', 'whom', '92', 'community-acquired', 'rbidities', 'isolates', 'upper', 'tract']
```

Figure 4.3 Sample high occurrence words

The **allenai/scibert_scivocab_uncased** BERT model is downloaded. Then the vocabulary of the SciBERT is updated with the new words. Figure 4.4 shows the updation of the vocabulary of SciBERT along with their old and new vocabulary length.

```
tokenizer.add_tokens(vocab)
model.resize_token_embeddings(len(tokenizer))
print("New vocabulary length : ",len(tokenizer))
# del vocab
```

```
Old vocabulary length : 31090
New vocabulary length : 32056
```

Figure 4.4 Updated SciBERT vocabulary size

Some of the newly added vocabulary words are as follows, **covid19**, **coronavirus-2**, **betacoronavirus**, **antivirals** etc., These words are added to both SciBERT model vocabulary and also the SciBERT word tokenizer.

The abstracts are taken and are sub word tokenized using SciBERT tokenizer and the abstract is truncated or padded to 256 block size and the input to BERT is created. Figure 4.5 shows the input ids which corresponds to the sub words in the CORD-19 abstract.

```
{'input_ids': tensor([ 102, 3201, 238, 8759, 11791, 1579, 5223, 111, 11061, 137,
326, 30109, 31926, 152, 1882, 131, 1921, 568, 190, 2343,
579, 7865, 31090, 17119, 5352, 235, 7516, 12378, 2883, 5889,
30143, 1224, 2278, 11204, 2526, 30117, 23065, 27738, 1045, 568,
190, 1532, 127, 17119, 5238, 263, 31415, 316, 5977, 263,
5376, 10812, 833, 5854, 9555, 267, 1887, 833, 111, 12423,
5934, 18609, 131, 568, 267, 6329, 545, 1921, 568, 267,
1887, 3307, 8707, 305, 131, 7861, 1761, 7512, 755, 5352,
8698, 305, 267, 31091, 111, 2486, 3407, 355, 1407, 1302,
```

Figure 4.5 Input features for BERT model

The SciBERT fine tuning is done by Masked Language Modeling (MLM). The input tokens are masked with a probability of 15%. Then the model is asked to predict what that masked word is. The model produces the softmax activated output of all token words. Then the softmax output is compared with the one-hot encoded value of the original word and the model is trained. Cross-entropy loss function is

used. AdamW Optimizer is used to fine tune the hyperparameters. Figure 4.6 shows the training output of the SciBERT fine tuning process. Then the CORD-SciBERT model is obtained which will be used for the feature extraction module.

```
[35]: TrainOutput(global_step=115195, training_loss=0.44445790873203456, metrics=
      {'train_runtime': 19158.325, 'train_samples_per_second': 96.203, 'train_steps
      _per_second': 6.013, 'total_flos': 2.425140841039442e+17, 'train_loss': 0.444
      45790873203456, 'epoch': 5.0})
```

Figure 4.6 SciBERT fine-tuning output

4.2.2 FEATURE EXTRACTION

The NER dataset is loaded from the files. The sentences and their tags are loaded from the dataset. They are converted into the (word, tag) format. In case of CORD-19 dataset, the dataset needs to sentence tokenized first and in other datasets, the sentences are loaded. Then each sentence from the input dataset is taken and is sub word tokenized using the custom CORD-SciBERT sub word tokenizer. Figure 4.7 shows the sub word tokenized output of a sample sentence from the input dataset.



```
["Chronic", "Administration", "of", "haloperidol", "increased", "Dpp6", "expression", "in", "mouse",
"brains", "."]
["chronic", "administration", "of", "halo", "##per", "##idol", "increased", "dpp", "##6", "expression",
"in", "mouse", "brains", "."]
```

Figure 4.7 Sub word tokenized sentence

Then the sentence is added with [CLS] token in the beginning and [SEP] in the end. Then the sentence is padded or truncated to max length of 256. And a mask is generated for the padded and original words. If the word is a padded token, it is marked as 1 otherwise it is 0. Then the label tag is encoded to integer and the feature embedding for the input is generated. Figure 4.8, Figure 4.9 and Figure 4.10 shows input ids, attention mask and target tags respectively.

```
[ 'Chronic', 'administration', 'of', 'haloperidol', 'increased', 'Dpp6', 'expression', 'in', 'mouse', 'brains', '.']
{'ids': tensor([ 102, 3164, 3762, 131, 17988, 291, 25453, 1175, 21375, 30142,
                940, 121, 3475, 15433, 205, 103, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Figure 4.8 Input feature id for SciBERT

```
'mask': tensor([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Figure 4.9 Input feature mask for SciBERT

```
'target_tag': tensor([2, 2, 2, 2, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Figure 4.10 Format of output of the model

Then the dataset row is passed to the CORD-19 SciBERT model and the embedding of the input is generated. Figure 4.11 shows the contextualized word embeddings of sample sentence from CORD-19 SciBERT.

```
["Chronic", "Administration", "of", "haloperidol", "increased", "Dpp6", "expression", "in", "mouse",
"brains", "."]
["chronic", "administration", "of", "halo", "##per", "##idol", "increased", "dpp", "##6", "expression",
"in", "mouse", "brains", "."]
tensor([[[[-1.0014, -0.4739, 0.2519, ..., -0.7363, -0.5779, 0.5717],
          [-0.7640, -0.7936, -0.4506, ..., -1.7523, -1.6091, 0.3277],
          [-0.9958, -0.7964, -0.3578, ..., -0.6214, -0.2014, 0.2999],
          ...,
          [-0.6479, -1.5951, 0.0292, ..., -0.6617, -0.5882, 0.4145],
          [-0.4898, -0.4360, -0.0209, ..., -0.8788, -1.3707, 0.7188],
          [-0.3240, -0.0856, 0.8726, ..., -1.2036, -2.3080, -0.0252]]],
        grad_fn=<NativeLayerNormBackward>])
torch.Size([1, 256, 768])
```

Figure 4.11 Contextualized embeddings from SciBERT

4.2.3 NAMED ENTITY RECOGNITION

```
class BERT_BiLSTM_CRF:
    def forward(self, ids, mask, token_type_ids, target_tag):
        x = BERT(ids, attention_mask=mask, token_type_ids=token_type_ids)
        h = BiLSTM(x)
        o_tag = Dropout(h)
        tag = Linear(o_tag)
        mask = torch.where(mask==1, True, False)
        loss=CRF(tag,target_tag,mask=mask, reduction='token_mean')
        return (-loss)
```

Figure 4.12 BERT-BiLSTM-CRF model code

The system defines a BERT-BiLSTM-CRF model for Named Entity Recognition. For the BERT layer, The already created CORD-SciBERT. Figure 4.12 shows the sample code of the construction of the BERT-BiLSTM-CRF model.

Initially, the CORD-19 SciBERT model is taken and the individual sentence's features such as input_ids, attention_mask, token_type_ids are fed into the BERT layer and the final hidden layer of the BERT model is taken as output. The output is of size (sentence_length[256], 768). Figure 4.13 shows the BERT layer output.

```
tensor([[[[-1.0014, -0.4739,  0.2519, ..., -0.7363, -0.5779,  0.5717],
          [-0.7640, -0.7936, -0.4506, ..., -1.7523, -1.6091,  0.3277],
          [-0.9958, -0.7964, -0.3578, ..., -0.6214, -0.2014,  0.2999],
          ...,
          [-0.6479, -1.5951,  0.0292, ..., -0.6617, -0.5882,  0.4145],
          [-0.4898, -0.4360, -0.0209, ..., -0.8788, -1.3707,  0.7188],
          [-0.3240, -0.0856,  0.8726, ..., -1.2036, -2.3080, -0.0252]]]],
        grad_fn=<NativeLayerNormBackward>)
torch.Size([1, 256, 768])
```

Figure 4.13 BERT layer output

Then the output of the BERT layer is fed into the BiLSTM layer. It takes vectors of size 768 and output vectors of size 1024 for each word in the sentence.

Since it is a bidirectional LSTM, 512 of the output is from the forward LSTM and other 512 is from the backward LSTM. (As shown in Figure 4.14)

```

...,
[-0.4123,  0.3271,  0.0852, ...,  0.0368, -0.2237,  0.0669],
[-0.3811,  0.2669,  0.1604, ...,  0.0597, -0.1699,  0.0505],
[-0.1611, -0.0041, -0.0299, ...,  0.2727, -0.2350,  0.0907]],
[[-0.1727,  0.0663,  0.1615, ..., -0.1408, -0.3058,  0.0710],
 [-0.1506,  0.2151,  0.0056, ..., -0.2915, -0.2896,  0.0583],
 [-0.3210,  0.2138,  0.2690, ..., -0.2086, -0.2853,  0.0485],
 ...,
 [-0.2137,  0.0427,  0.0975, ...,  0.1209, -0.2319,  0.0315],
 [-0.1747,  0.0309,  0.0789, ...,  0.1383, -0.2505,  0.0775],
 [-0.1932, -0.0604, -0.0602, ...,  0.2187, -0.1940,  0.1869]]],
grad_fn=<TransposeBackward0>)
torch.Size([8, 256, 1024])

```

Figure 4.14 BiLSTM layer output

Then, the LSTM layer output is passed to the Dropout layer. Then the dropout layer output is passed to the fully connected layer which for each word in the sentence maps the 1024 input vector into the vector of size equal to the number of types of labels in the NER dataset (3 for NCBI, As shown in Figure 4.15)

```

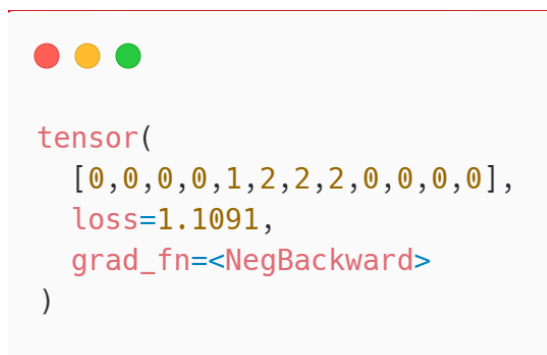
[[-0.0636, -0.1357, -0.1430],
 [-0.0649,  0.1567, -0.1360],
 [-0.0280,  0.1010, -0.1456],
 ...,
 [ 0.0712,  0.0821, -0.0846],
 [-0.0316,  0.1367, -0.0552],
 [ 0.0290,  0.0448,  0.0273]]], grad_fn=<AddBackward0>)
torch.Size([8, 256, 3])

```

Figure 4.15 Fully Connected layer output

Then the output from the fully connected layer for each word is passed to the Conditional Random Field (CRF) layer which finds the negative log-likelihood loss and provides it as output. This loss is negated and is used as the loss function for the

model and is back propagated. Figure 4.16 shows the CRF layer tag output along with the loss value.



```

tensor(
  [0,0,0,0,1,2,2,2,0,0,0],
  loss=1.1091,
  grad_fn=<NegBackward>
)

```

Figure 4.16 CRF Layer loss output

Multiple optimizers are tried and the optimizer that is finalised is Stochastic Gradient Descent (SGD). Figure 4.17 shows the SGD optimizer formulae. The system uses momentum to avoid oscillations and to converge faster than traditional gradient descent algorithms.

$$w_t = w_{t-1} - \eta V_{dw_t}$$

$$\text{where } V_{dw_t} = \beta V_{dw_{t-1}} + (1 - \beta) \frac{\partial L}{\partial w_{t-1}}$$

$$b_t = b_{t-1} - \eta V_{db_t}$$

$$\text{where } V_{db_t} = \beta V_{db_{t-1}} + (1 - \beta) \frac{\partial L}{\partial b_{t-1}}$$

Figure 4.17 SGD Optimizer

Initially all the three models for 3 datasets (NCBI-Disease, CHEMDNER, JNLPBA) are loaded into the GPU memory. Then the label encoders corresponding to each model are also loaded. Then the CORD-19 dataset is loaded into the memory. Then the title, abstract and full text corresponding to a single row of the CORD-19 dataset is taken and is merged to form the input text. Then the text is sentence

tokenized. Then the sentence is word tokenized using the NLTK word tokenizer package to split words and also punctuations. (As shown in Figure 4.18). Then the words are passed to the SciBERT sub word feature extraction to extract the input_ids, attention_mask and token_type_ids for the sentence. Figure 4.19, 4.20 shows input ids and attention mask respectively.

```
[ '-1', 'Programmed', 'ribosomal', 'frameshifting', '(', 'PRF', ')', 'in', 'synthesizing', 'the', 'gag-pro', 'precursor', 'polyprotein', 'of', 'Simian', 'retrovirus', 'type-1', '(', 'SRV-1', ')', 'is', 'stimulated', 'by', 'a', 'classical', 'H-type', 'pseudoknot', 'which', 'forms', 'an', 'extended', 'triple', 'helix', 'involving', 'base-base', 'and', 'base-sugar', 'interactions', 'between', 'loop', 'and', 'stem', 'nucleotides', '.']
```

Figure 4.18 Word Tokenizing sample sentence

```
tensor([[ 102,   101, 16403, 16786, 31108,   145,   492, 30122,   546,   121,
         20439, 4681,   111, 17032,   579,   178, 9571, 31261,   131,   462,
         1026, 8543, 12833, 1211,   579,   158,   145, 4193, 30129,   579,
         158,   546,   165, 8157,   214,   106, 5109,   151,   579, 1211,
        8953, 8326,   184,   334, 3444,   130, 3956, 8023, 14217, 5005,
        2971,   101, 2971,   137, 2971,   101, 10499, 2697,   467, 4472,
         137, 4151, 14100,   205,   103,    0,    0,    0,    0,    0
        ]])
```

Figure 4.19 Input IDs for sample sentence

```
tensor([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
         1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
         1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
        ]])
```

Figure 4.20 Attention mask for sample sentence

Then the individual sentence features are passed to the three models and based on the predicted tags the entities are extracted by matching the tags with the words. Figure 4.21 shows the extracted entities from the sample sentence. After running the above procedure for all the sentences, the extracted entities are stored in a MongoDB database for future reference. Table 4.8 shows the complete statistics of the entities found from CORD-19.

```

Diseases : [{ 'word': 'ian retrovirus type', 'type': 'Disease' }]
Chemicals: [{ 'word': 'sugar', 'type': 'Chemical' }, { 'word': 'nucleotides', 'type': 'Chemical' }]
Proteins : [{ 'word': 'gag-pro precursor polyprotein', 'type': 'protein' }]

```

Figure 4.21 Extracted entities from sample sentence

Table 4.8 Overall entities extracted from CORD-19

Entity Type	Total no. of instances	Unique no. of instances
Disease	1,00,441	17,672
Chemical	66,212	7,841
Protein	3,66,963	1,29,972

4.2.4 RELATION EXTRACTION

In the BC5CDR dataset, only positive relations are given. To create negative relations to equal the dataset for better training of the model. Two random entities from the text that are not in relation are taken and are marked as negative relations. For example, in abstract 227508, the found entities are shown in Figure 4.22. Here only D007022 and D008750 are in positive relation. So, the negative relation is created by combining D007022 and D003000 or D006973 or D009270 and so on. In CHEMPROT dataset classes CPR:8 and CPR:0 is left out because there are not enough samples to evaluate them correctly in the development and test dataset.

```

{'D009270': 'naloxone', 'D003000': 'clonidine', '-1': '[3h]-dihydroergocryptine', 'D008750': 'alpha-methyl dopa'}
{'D006973': 'hypertensive', 'D007022': 'hypotensive'}
*****

```

Figure 4.22 Chemicals and Diseases from sample BC5CDR row

The dataset is taken and is read into the memory. It is converted into the formation of [text, entity1, entity2, relation type]. Then the entity1 and entity2 of the relation is concatenated using space and \t is added to the end to mark a single

from the BC5CDR SciBERT model. All the extracted relations from the CORD-19 are stored in the MongoDB database. Table 4.9 shows the statistics of relations extracted from CORD-19.

```
{
  "entity_1": "cytidine",
  "entity_2": "plv",
  "relation": "CID"
}
```

Figure 4.29 Relation extraction model output

Table 4.9 Overall relations extracted from CORD-19

Relation Type	Total no. of Instances	Unique no. of Instances
Chemical Induced Disease	5425	4071
Chemical Protein Relation	87,794	72,891

4.2.5 GRAPH CONSTRUCTION

The relations extracted from the CORD-19 dataset is taken from the MongoDB database and is converted to a csv file with entity_1, entity_2, relation column. Only the relations whose entities occurs more than 5 times are taken into consideration. Figure 4.30 shows the filtering of entities and relations.

```
Initial no. of relations from the database : 76962
No. of entities which occurs more than 5 times : 5237
No. of relations which contains atleast one 5-occurrence entity : 69782
```

Figure 4.30 Final relations after filtering of entities

The final list of relations is used to create the Knowledge graph. A Neo4j graph database is initiated. The nodes are entities and the edges are relations. And all the relations are inserted into the database and the knowledge graph is created.

Figure 4.31 shows a portion of Knowledge Graph focused on coronavirus. Here Red denotes Disease, Yellow for Chemical and Purple for Protein.

4.2.6 REPRESENTATION LEARNING

Initially the relations list is taken and from them negative relations are created by taking random entities from different relations and combining them. Then the TransD model is trained using the hyperparameters `batch_size = 100`, `entities_dimension_size = 400`, `loss_function = MarginLoss`, `epochs = 700`. Then the embeddings of entities are taken and only the entities with edges greater than 5 are taken and is compared with coronavirus. And the diseases, chemicals and proteins with the highest cosine similarity score with coronavirus taken and is produced as final output. Figure 4.32, Figure 4.33 and Figure 4.34 shows the top diseases, chemicals and proteins related to COVID-19 respectively. In these figures thicker edges and darker colours indicates strong relationship and vice versa.

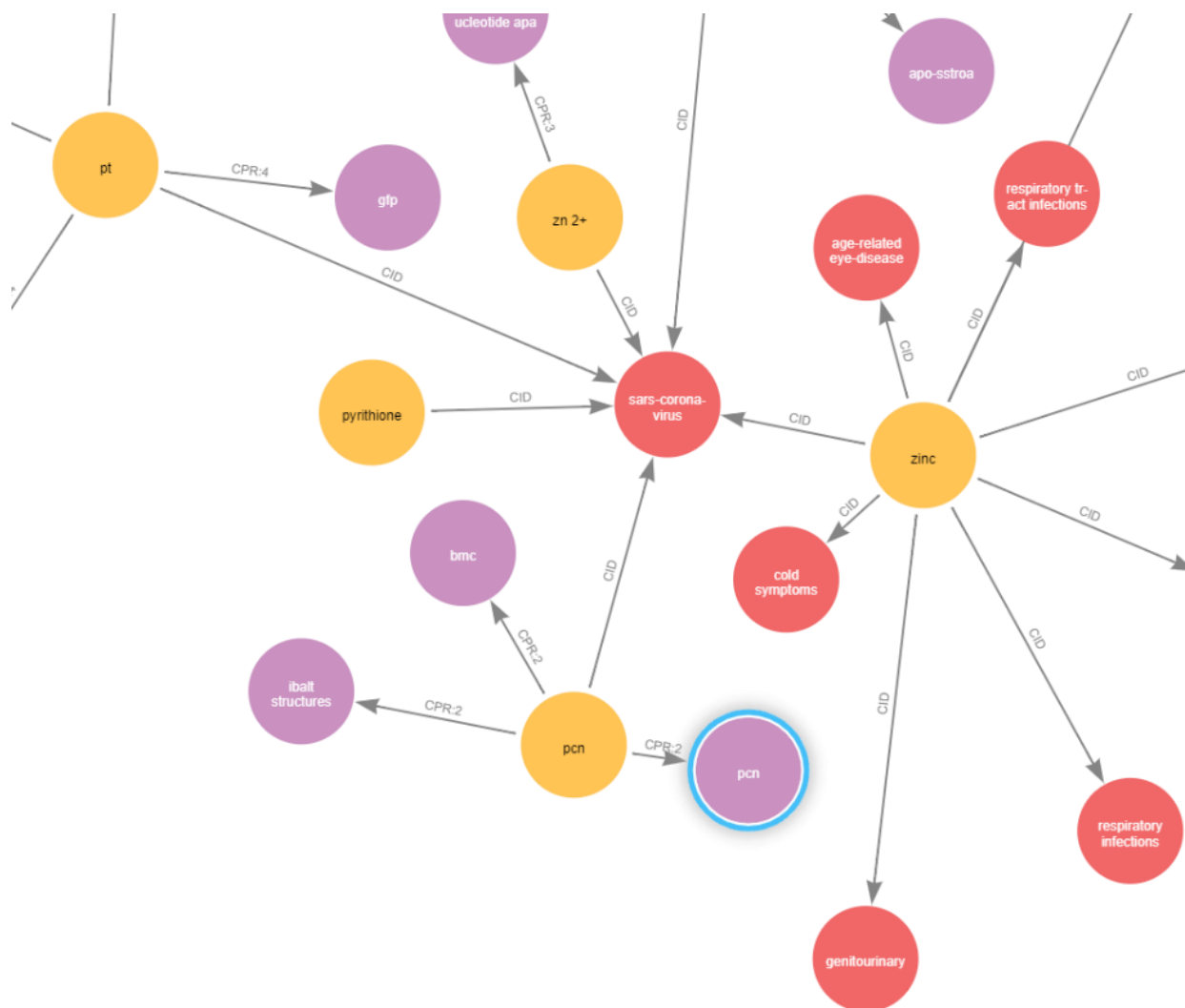


Figure 4.31 Portion of knowledge graph focused on COVID-19



Figure 4.32 Top Diseases related to COVID-19

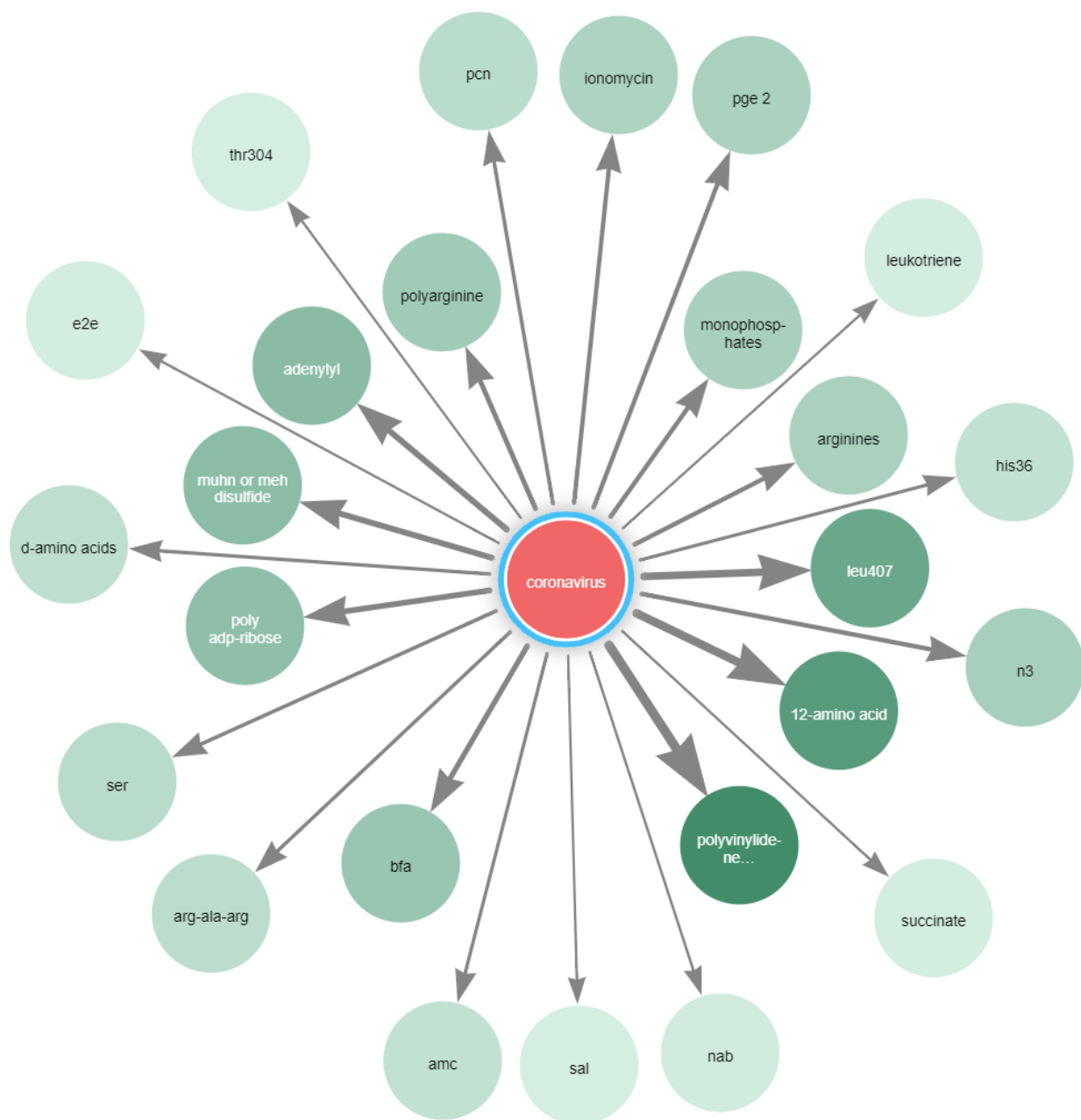


Figure 4.33 Top chemicals related to COVID-19



Figure 4.34 Top proteins related to COVID-19

4.3 HYPERPARAMETERS

For Named Entity Recognition, by training the BERT-BiLSTM-CRF model using various hyperparameters the best possible hyperparameters for the model are found which differs for the NCBI-Disease, CHEMDNER and JNLPBA dataset. Table 4.10 shows the Hyperparameters used for different datasets.

Table 4.10 Hyperparameters for Different NER datasets

Dataset	No. of Labels	Epochs	Batch Size	Sequence Length
NCBI-Disease	3	10	16	256
CHEMDNER	3	6	16	256
JNLPBA	11	4	16	256

For Relation Extraction, By training the SciBERT model using various hyperparameters the best possible hyperparameters for model are found which differs for the BC5CDR and CHEMPROT dataset. Table 4.11 shows the Hyperparameters used for different datasets.

Table 4.11 Hyperparameters for Relation Extraction

Dataset	No. of Relations	Sequence Length	Epoch
BC5CDR	2	512	3
CHEMPROT	9	512	4

4.4 PERFORMANCE METRICS

4.4.1 NAMED ENTITY RECOGNITION

Here the precision, recall and F1 are the metrics used which are calculated based on whether the entire phrase is correctly detected or not. Table 4.12 shows the evaluation metrics for the Named Entity Recognition model by testing the model on test dataset for 5 times and averaging the output. Figure 4.35 shows the performance of the system in graphical format. The definition of evaluation metrics in the domain of NER are as follows,

Precision : Percentage of named entities found by the algorithm that are correct.

Recall : Percentage of named entities defined in the corpus that were found.

F1 : $2 * \text{Precision} * \text{Recall} / (\text{Recall} + \text{Precision})$ (4.1)

Table 4.12 Named Entity Recognition evaluation metrics

Dataset	Precision	Recall	F1
NCBI-Disease	88.49	89.02	88.76
JNLPBA	71.25	81.5	76.06
CHEMDNER	90.88	92.25	91.56

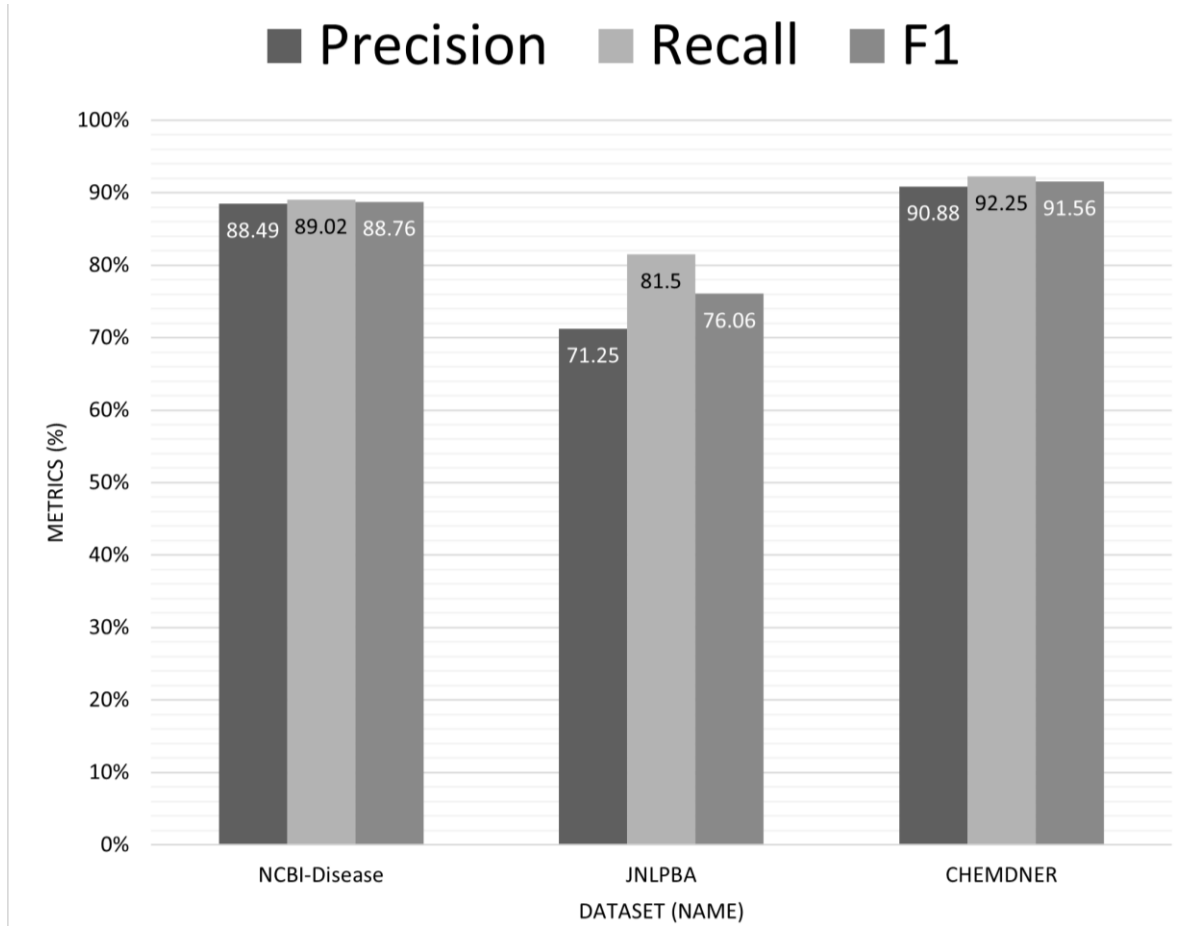


Figure 4.35 NER Metrics Graph

4.4.2 RELATION EXTRACTION

Relation extraction is a classification problem. Hence Precision, Recall and F1 are used to evaluate the model. Table 4.13 denotes the Relation extraction evaluation metrics of the system on both the datasets. Figure 4.36 shows the performance of the system in graphical format. The definition are as follows,

$$\text{Precision} : TP / (TP + FP) \quad (4.2)$$

$$\text{Recall} : TP / (TP + FN) \quad (4.3)$$

$$\text{F1} : 2 * \text{Precision} * \text{Recall} / (\text{Recall} + \text{Precision}) \quad (4.4)$$

Here TP denotes true positive, FP denotes false positive, FN denotes false negative in the confusion matrix.

Table 4.13 Relation Extraction evaluation metrics

Dataset	Precision	Recall	F1
BC5CDR	74.00	73.00	73.00
CHEMPROT	72.00	71.00	71.00

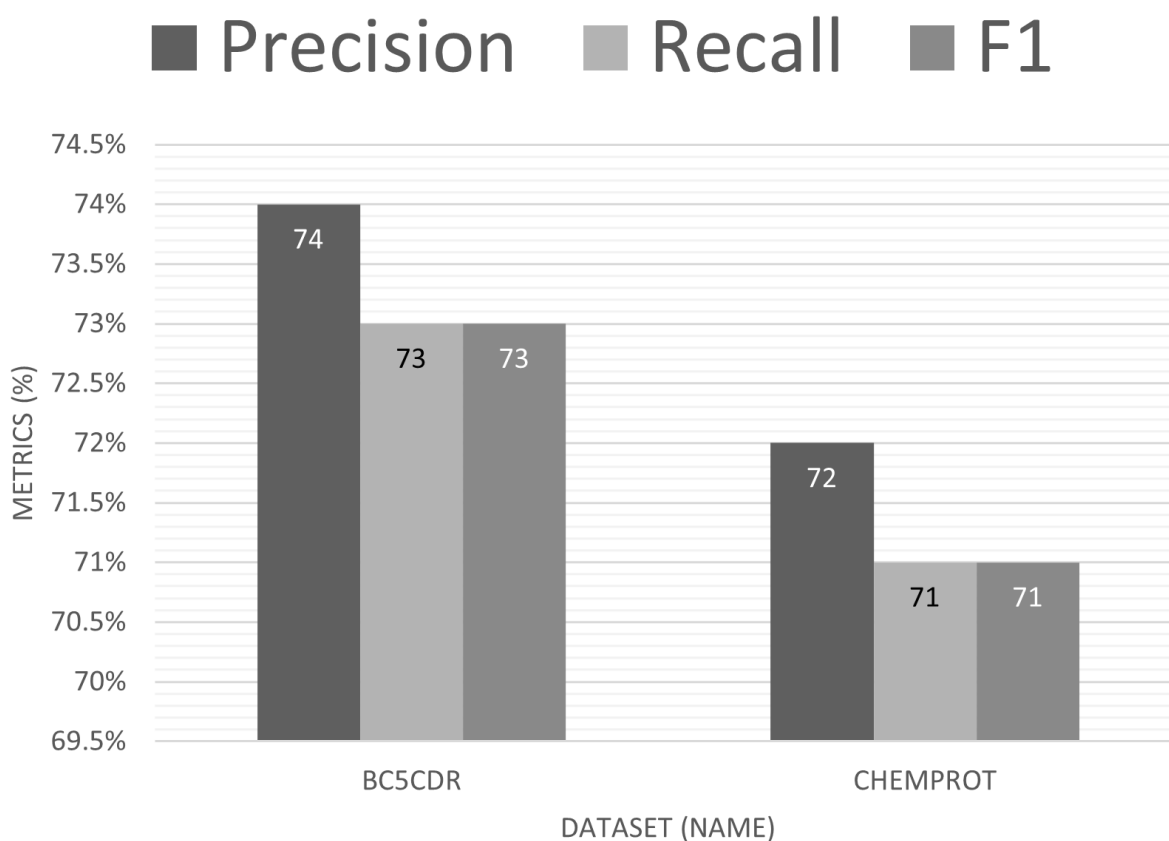


Figure 4.36 RE Metrics Graph

4.4.3 SYSTEM

Since there is no gold standard data available on the CORD-19 dataset. The system uses the Relation Extraction metrics as the output of the system. To ensure

the validity of the formed Knowledge Graph by performing one of the downstream tasks of representation learning and the output of the representation learning is verified manually by checking the relation between the mentioned entity and COVID-19.

4.4.4 REPRESENTATION LEARNING

The relation between entities outputted by the representation learning module needs to be verified manually. Table 4.14 showcases some of the evidences.

Table 4.14 Evidences for certain highly COVID-19 related entities

Entity	Entity Type	Evidence
HIV	Disease	Illanes-Álvarez, Francisco et al. “Similarities and differences between HIV and SARS-CoV-2.” <i>International journal of medical sciences</i> vol. 18,3 846-851. 1 Jan. 2021, doi:10.7150/ijms.50133
ARDS	Disease	Aslan, A., Aslan, C., Zolbanin, N.M. et al. acute respiratory distress syndrome in COVID-19: possible mechanisms and therapeutic management. <i>Pneumonia</i> 13, 14 (2021). https://doi.org/10.1186/s41479-021-00092-9
CRD	Disease	Guillaume Beltramo, Jonathan Cottenet, Anne-Sophie Mariet, Marjolaine Georges, Lionel Piroth, Pascale Tubert-Bitter, Philippe Bonniaud, Catherine Quantin European Respiratory Journal 2021; DOI: 10.1183/13993003.04474-2020

Hepatitis C	Disease	Ronderos, Diana et al. "Chronic hepatitis-C infection in COVID-19 patients is associated with in-hospital mortality." <i>World journal of clinical cases</i> vol. 9,29 (2021): 8749-8762. doi:10.12998/wjcc.v9.i29.8749
HSV-2	Disease	Shanshal, Mohammed, and Hayder Saad Ahmed. "COVID-19 and Herpes Simplex Virus Infection: A Cross-Sectional Study." <i>Cureus</i> vol. 13,9 e18022. 16 Sep. 2021, doi:10.7759/cureus.18022
PGE2	Chemical	Ricke-Hoch M, Stelling E, Lasswitz L, Gunesch AP, Kasten M, et al. (2021) Impaired immune response mediated by prostaglandin E2 promotes severe COVID-19 disease. <i>PLOS ONE</i> 16(8): e0255335. https://doi.org/10.1371/journal.pone.0255335
Polyvinylidene fluoride	Chemical	Zinc Oxide Nanoparticle-Loaded Electrospun Polyvinylidene Fluoride Nanofibers as a Potential Face Protector against Respiratory Viral Infections Hassan Nageh, Merna H. Emam, Fedaa Ali, Nasra F. Abdel Fattah, Mohamed Taha, Rehab Amin, Elbadawy A. Kamoun, Samah A. Loutfy, and Amal Kasry <i>ACS Omega</i> 2022 7 (17), 14887-14896 DOI: 10.1021/acsomega.2c00458
Poly adp ribose	Chemical	Badawy AA. Immunotherapy of COVID-19 with poly (ADP-ribose) polymerase inhibitors: starting with nicotinamide. <i>Biosci Rep.</i> 2020 Oct 30;40(10): BSR20202856. doi: 10.1042/BSR20202856. PMID: 33063092; PMCID: PMC7601349.

Succinate	Chemical	Water-soluble tocopherol derivatives inhibit SARS-CoV-2 RNA-dependent RNA polymerase Hayden T. Pacl, Jennifer L. Tipper, Ritesh R. Sevalkar, Andrew Crouse, Camerron Crowder, UAB Precision Medicine Institute, Shama Ahmad, Aftab Ahmad, Gillian D. Holder, Charles J. Kuhlman, Krishna C. Chinta, Sajid Nadeem, Todd J. Green, Chad M. Petit, Adrie J.C. Steyn, Matthew Might, Kevin S. Harrod bioRxiv 2021.07.13.449251; doi: https://doi.org/10.1101/2021.07.13.449251
SER	Chemical	Rahbar Saadat, Yalda et al. "Host Serine Proteases: A Potential Targeted Therapy for COVID-19 and Influenza." Frontiers in molecular biosciences vol. 8 725528. 30 Aug. 2021, doi:10.3389/fmolb.2021.725528
Trypsin 1	Protein	Kim Y, Jang G, Lee D, Kim N, Seon JW, Kim YH, Lee C. Trypsin enhances SARS-CoV-2 infection by facilitating viral entry. Arch Virol. 2022 Feb;167(2):441-458. doi: 10.1007/s00705-021-05343-0. Epub 2022 Jan 26. PMID: 35079901; PMCID: PMC8789370.
IL-1 β	Protein	Mardi A, Meidaninikjeh S, Nikfarjam S, Majidi Zolbanin N, Jafari R. Interleukin-1 in COVID-19 Infection: Immunopathogenesis and Possible Therapeutic Perspective. Viral Immunol. 2021

		Dec;34(10):679-688. doi: 10.1089/vim.2021.0071. Epub 2021 Dec 8. PMID: 34882013.
Kinase	Protein	Pillaiyar, Thanigaimalai, and Stefan Laufer. “Kinases as Potential Therapeutic Targets for Anti-coronaviral Therapy.” Journal of medicinal chemistry vol. 65,2 (2022): 955-982. doi: 10.1021/acs.jmedchem.1c00335
DICER	Protein	Mousavi, Seyyed Reza et al. “Dysregulation of RNA interference components in COVID-19 patients.” BMC research notes vol. 14,1 401. 29 Oct. 2021, doi:10.1186/s13104-021-05816-0
Cyclo Oxygenase 2	Protein	Baghaki, Semih et al. “COX2 inhibition in the treatment of COVID-19: Review of literature to propose repositioning of celecoxib for randomized controlled studies.” International journal of infectious diseases: IJID: official publication of the International Society for Infectious Diseases vol. 101 (2020): 29-32. doi: 10.1016/j.ijid.2020.09.1466

4.5 COMPARATIVE ANALYSIS

4.5.1 NAMED ENTITY RECOGNITION

For comparison of the Proposed System with existing systems, the NCBI-Disease dataset is taken. The proposed system is BERT-BiLSTM-CRF for Named Entity Recognition. Table 4.15 and Figure 4.37 shows the comparison of proposed system with BiLSTM, LSTM-CRF, BiLSTM-CRF and SciBERT. The BERT-

BiLSTM-CRF model has better Precision and F1 value than standard SciBERT. Hence the usage of BERT-BiLSTM-CRF model is justified.

Table 4.15 Comparative Analysis for NER

Model	Precision	Recall	F1
BiLSTM + Word Embedding	84.87	74.11	79.13
LSTM-CRF + Word and Char Embedding	85.20	82.40	83.80
BiLSTM-CRF + Word Embedding	86.75	87.11	86.93
SciBERT	85.47	90.10	87.73
BERT-BiLSTM-CRF (Proposed System)	88.49	89.02	88.76

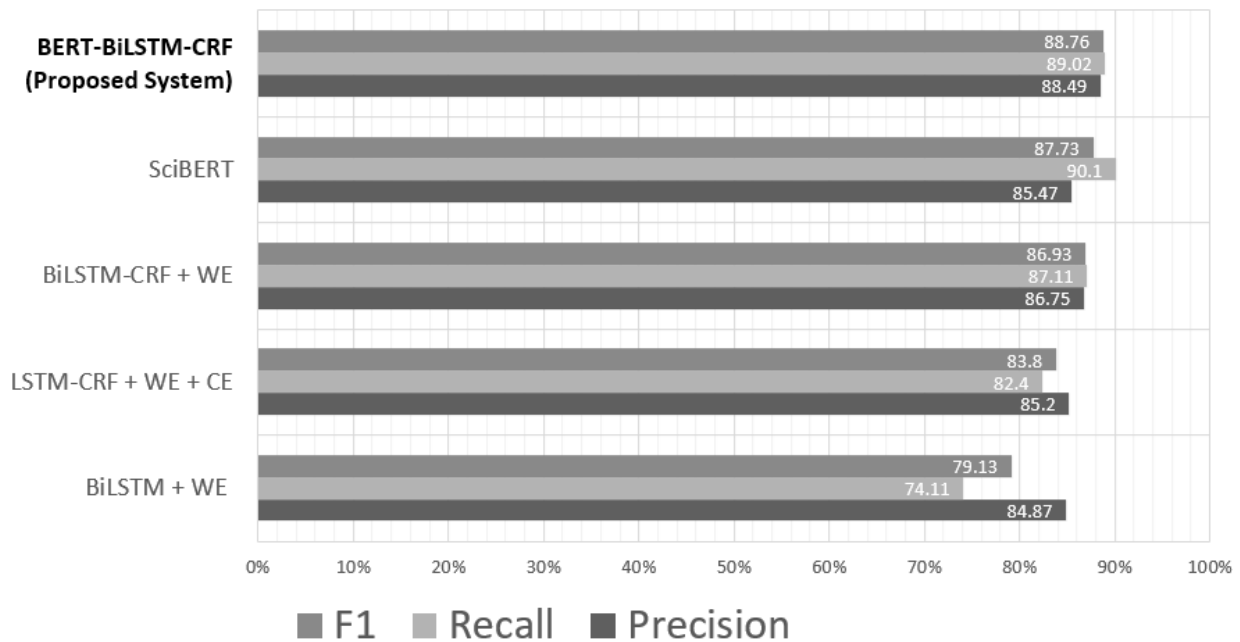


Figure 4.37 Comparative Analysis of different NER models

4.5.2 RELATION EXTRACTION

For comparison of the proposed system with existing systems, the BC5CDR dataset is taken. The Proposed System is SciBERT. Table 4.16 shows the results of the comparison analysis of SciBERT with existing systems like 1d-CNN, LSTM-SVM, LSTM-CNN, LSTM-CRF. Figure 4.38 shows the comparative analysis in graphical format.

Table 4.16 Comparative analysis for RE

Model	Precision	Recall	F1
1d-CNN + Glove	60.85	56.42	58.55
LSTM-SVM	64.90	49.30	56.00
LSTM-CNN	54.30	65.90	59.50
LSTM-CRF	60.00	67.50	63.50
SciBERT (Proposed System)	74.00	73.00	73.00

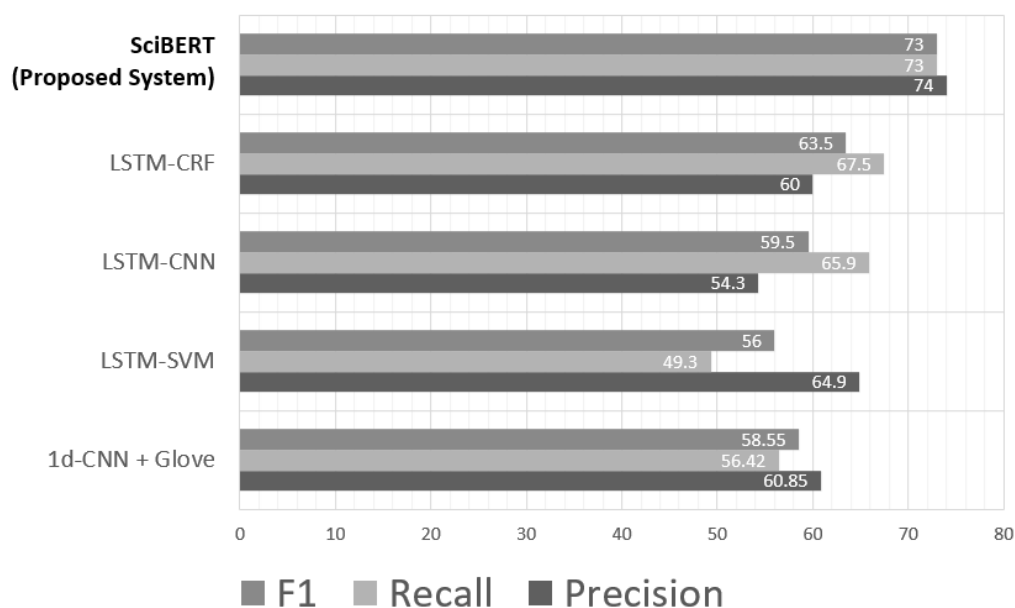


Figure 4.38 Comparative analysis of different RE models

4.6 TEST CASES

Table 4.17 Test cases of the system

Test Case	Input	Expected Output	Actual Output
Only non-English rows from CORD-19 to be removed. (Preprocessing)	Mycobacterium tuberculosis (M.tb) is responsible for more deaths globally.	Should not be removed	Removed
Similarity between unrelated entities to be smaller. (Feature Extraction)	“coronavirus” “computer”	<0.4	0.65
Tagging entities correctly. (Named Entity Recognition)	The degeneracy in sequences recognized by the OTFs may be important in widening the range over which gene expression can be modulated and in establishing cell type specificity.	“OTFs” - Protein	“OTFs” - DNA
Finding all the entities. (Named Entity Recognition)	In whole cell experiments at 37 degrees C, nuclear	“[125I] T3” - Protein	“[125I] T3” - O

Entity Recognition)	binding of [¹²⁵ I] T3 was saturable (K _d 34 +/- 6 pmol/l) and of finite capacity (approximately equal to 350 sites/cell)		
Tagging adjectives when necessary. (Named Entity Recognition)	Multiple B lineage genes	“Multiple B lineage genes” - DNA	“B lineage genes” – DNA
Tagging entities completely. (Named Entity Recognition)	Recombination of the MPC11 plasma B-cell derived NF-Y A: B: C complex with the low molecular mass protein fraction, NF-Y-associated factors (YAFs), derived from mature A20 B-cell nuclei, conferred high affinity anion exchange binding to NF-Y as an intact trimeric complex.	“B-cell derived NF-Y A: B: C complex” - Protein	“NF-Y A: B: C complex” – Protein
Not finding all relations. (Relation Extraction)	Moderate to high dose corticosteroid use is recognized as a major risk factor for src. Furthermore,	CID relation between “ssc” and “corticosteroid”	No relation

	there have been reports of thrombotic microangiopathy precipitated by cyclosporine in patients with ssc.		
Misclassifying relations (Relation Extraction)	Dimemorfan pre-treatment also attenuated the KA-induced increases in c-fos c-jun expression, activator protein-1 DNA-binding activity, and loss of cells in the CA1 and CA3 fields of the hippocampus.	CPR:4 relation between “Dimemorfan” and “c-fos”	CPR:3 relation between “Dimemorfan” and “c-fos”
Finding highly COVID-19 related entities from Knowledge Graph (Representation Learning)	Extracted Complete Knowledge Graph	Only diseases, chemicals and proteins.	“heart” (not a disease) as disease entity related to COVID-19

CHAPTER 5

CONCLUSION AND FUTURE WORK

The paper proposes a generic pipeline, for association analysis with respect to a given entity from an unstructured dataset. The part of the pipeline integrating IE and KG construction keeps human out-of-the-loop. The system also proposes creating custom SciBERT for word embeddings by fine tuning the SciBERT on the CORD-19 abstracts. The system also proposes the BERT-BiLSTM-CRF model for Biomedical Named Entity Recognition which as the results shown, performed better than SciBERT. The system also uses the SciBERT model for extracting relations between entities. Then the entities are filtered to reduce noise and the final set of relations is used to produce the Knowledge Graph.

In order to learn the latent representation of the formed Knowledge Graph, the system uses TransD Knowledge Graph Embedding Method. The system's approach is evaluated only on CORD-19 dataset and no additional resources have been employed. However, due to the lack of gold standard data, the system uses the metrics from Relation Extraction as the final metrics of the system. Also, the final Knowledge graph is evaluated by comparing the embeddings of COVID-19 with all other entities to find the top COVID-19 related diseases, chemicals and proteins.

As a future scope, Researchers can plan to implement a normalization and abbreviation expansion module after the detection of entities. The study of these top predicted entities, by the domain experts, can help them understand the different types of associations and relationships they exhibit with respect to COVID-19.

REFERENCES

1. Ayoub Harnoune, Maryem Rhanoui, Mounia Mikram, Siham Yousfi, Zineb Elkaimbillah, Bouchra El Asri (2021) BERT based clinical knowledge extraction for biomedical knowledge graph construction and analysis, Computer Methods and Programs in Biomedicine Update, Volume 1, 2021, 100042,ISSN2666-9900.
2. Cheng D, Knox C, Young N, Stothard P, Damaraju S, Wishart DS. PolySearch: a web-based text mining system for extracting relationships between human diseases, genes, mutations, drugs and metabolites. *Nucleic Acids Res* (2008) Jul 1;36(Web Server issue):W399-405. doi: 10.1093/nar/gkn296. Epub 2008 May 16. PMID: 18487273; PMCID: PMC2447794.
3. Chikashi Nobata, Nigel Collier, and Jun-ichi Tsujii (1999) Automatic term identification and classification in biology texts. In Proc. of the 5th NLPRS. Citeseer, 369–374.
4. Dan Shen, Jie Zhang, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2003. Effective Adaptation of Hidden Markov Model-based Named Entity Recognizer for Biomedical Domain. In Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine, pages 49–56, Sapporo, Japan. Association for Computational Linguistics.
5. Daniel Domingo-Fernandez, Shounak Baksi, Bruce´ Schultz, Yojana Gadiya, Reagon Karki, Tamara Raschka, Christian Ebeling, Martin Hofmann Apitius, and Alpha Tom Kodamullil. (2020) Covid19 knowledge graph: a computable, multimodal, cause-and-effect knowledge model of covid-19 pathophysiology. bioRxiv.

6. Doğan RI, Leaman R, Lu Z. (2014) NCBI disease corpus: a resource for disease name recognition and concept normalization. *J BiomedInform.* 2014 Feb;47:1-10. doi: 10.1016/j.jbi.2013.12.006. Epub2014 Jan 3. PMID: 24393765; PMCID: PMC3951655.
7. Fundel, K., Küffner, R., & Zimmer, R. (2006). RelEx—Relation extraction using dependency parse trees. *Bioinformatics*, 23(3), 365-371.
8. Giorgi, J., Wang, X., Sahar, N., Shin, W. Y., Bader, G. D., & Wang, B. (2019). End-to-end named entity recognition and relation extraction using pre-trained language models. *arXiv preprint arXiv:1912.13415*.
9. Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. (2015) Knowledge Graph Embedding via Dynamic Mapping Matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, Beijing, China. Association for Computational Linguistics.
10. Iz Beltagy, Kyle Lo, and Arman Cohan (2019) Scibert: A pretrained language model for scientific text. In *EMNLP/IJCNLP*.
11. J. Lafferty, A. McCallum, F.C.N Pereira (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
12. Jensen, K., Panagiotou, G., & Kouskoumvekaki, I. (2014). Integrated text mining and chemoinformatics analysis associates diet to health benefit at molecular level. *PLoS computational biology*, 10(1), e1003432.
13. Jettakul, A., Wichadakul, D., & Vateekul, P. (2019). Relation extraction between bacteria and biotopes from biomedical texts with attention mechanisms and domain-specific contextual representations. *BMC bioinformatics*, 20(1), 1-17.

14. Jun'ichi Kazama, Takaki Makino, Yoshihiro Ohta, and Jun'ichi Tsujii (2002) Tuning support vector machines for biomedical named entity recognition. In Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain, pages 1–8, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
15. Kim, D., Lee, J., So, C. H., Jeon, H., Jeong, M., Choi, Y., ... & Kang, J. (2019). A neural named entity recognition and multi-type normalization tool for biomedical text mining. *IEEE Access*, 7, 73729-73740.
16. Kim, T.; Yun, Y.; Kim, N. (2021) Deep Learning-Based Knowledge Graph Generation for COVID-19. *Sustainability* 2021, 13, 2276.
17. Krallinger M, Rabal O, Leitner F, Vazquez M, Salgado D, LuZ, Leaman R, Lu Y, Ji D, Lowe DM, Sayle RA, Batista-NavarroRT, Rak R, Huber T, Rocktäschel T, Matos S, Campos D, TangB, XuH, Munkhdalai T, Ryu KH, Ramanan SV, Nathan S, Žitnik S, BajecM, Weber L, Irmer M, Akhondi SA, Kors JA, Xu S, An X, Sikdar UK, Ekbal A, Yoshioka M, Dieb TM, Choi M, Verspoor K, KhabsaM, Giles CL, Liu H, Ravikumar KE, Lamurias A, Couto FM, Dai HJ, Tsai RT, Ata C, Can T, Usié A, Alves R, Segura-Bedmar I, MartínezP, Oyarzabal J, Valencia A. (2015) The CHEMDNER corpus of chemicals and drugs and its annotation principles. *J Cheminform.* 2015Jan19.
18. Ling, Y., Hasan, S. A., Farri, O., Chen, Z., van Ommering, R., Yee, C., & Dimitrova, N. (2019). A domain knowledge-enhanced LSTM-CRF model for disease named entity recognition. *AMIA Summits on Translational Science Proceedings*, 2019, 761.

19. Luo, L., Yang, Z., Yang, P., Zhang, Y., Wang, L., Lin, H., & Wang, J. (2018). An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. *Bioinformatics*, 34(8), 1381-1388.
20. Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018) Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
21. Nigel Collier and Jin-Dong Kim. (2004) Introduction to the bio-entity recognition task at jnlpba. In *NLPBA/BioNLP*.
22. Percha, B., Garten, Y., & Altman, R. B. (2012). Discovery and explanation of drug-drug interactions via text mining. In *Biocomputing 2012* (pp. 410-421).
23. Rebholz-Schuhmann D. (2013) Biomedical Named Entity Recognition, Whatizit. In: Dubitzky W., Wolkenhauer O., Cho KH., Yokota H. (eds) *Encyclopedia of Systems Biology*. Springer, New York, NY. https://doi.org/10.1007/978-1-4419-9863-7_151.
24. Repke T., Krestel R. (2021) Extraction and Representation of Financial Entities from Text. In: Consoli S., Reforgiato Recupero D., Saisana M. (eds) *Data Science for Economics and Finance*. Springer, Cham.
25. Weber, L., Sanger, M., Munchmeyer, J., Habibi, M., Leser, U., & Akbik, A. (2019). HunFlair: an easy-to-use tool for state-of-the-art biomedical named entity recognition. *Bioinformatics*, 37(17), 2792-2794.

26. Yanran Li, Wenjie Li, Fei Sun, and Sujian Li (2015) Component-enhanced chinese character embeddings. arXiv preprint arXiv:1508.06669.
27. Yoshua Bengio, R. Ducharme, Pascal Vincent, and Christian Janvin (2003) A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.
28. Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014, August). Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers* (pp. 2335-2344).
29. Zheng, S., Rao, J., Song, Y., Zhang, J., Xiao, X., Fang, E., Yang, Y. and Niu, Z., (2020) PharmKG: a dedicated knowledge graph benchmark for biomedical data mining. *Briefings in Bioinformatics*.
30. Zhu, Q., Li, X., Conesa, A., & Pereira, C. (2017). GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text. *Bioinformatics*, 34(9), 1547-1554.