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

Guided By
Dr. G. Sudhakaran

T. Athiban - 2018103013
N. Prathesh - 2018103576
M.Syed Mohamed Asif - 2018103612

PROBLEM STATEMENT

- To extract information regarding COVID-19 from CORD-19 in a fully autonomous way using NLP language models and techniques.
- To gather named entities such as diseases, proteins and chemicals from the CORD-19 dataset using Named Entity Recognition models such as BiLSTM-CRF.
- To extract relationships between entities (i.e Chemical Induced Disease, Chemical and Protein Interactions) from the CORD-19 dataset using Transformers based Language Models (i.e BERT ,SciBERT ,BioBERT).
- To organize the found entities and their relationships in the form of a biomedical knowledge graph using which further research can be done.
- As an example downstream task, we embed 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 10 entities related to COVID-19 are found and their validity are checked.

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.

For this project, we define 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. We categorize these automated approaches 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.

In this project, we consider the Automated way of extracting knowledge graphs from text using Deep Learning and Natural Language Processing techniques. The advantages of this approach includes 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. We introduce a fully automated generic pipeline consisting of an Information Extraction (IE) system followed by Knowledge Graph construction.

RELATED WORK

1. NAMED ENTITY RECOGNITION (NER)

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. While scanning texts for exactly matching terms included in the dictionaries is a straightforward and precise way of named entity recognition, recall of these systems tends to be lower. One prominent example of a dictionary-based BioNER model is in the association mining tool Polysearch (Cheng et al., 2008). Another example is Whatizit (Rebholz-Schuhmann, 2013), 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), Hidden Markov models (Shen et al., 2003), Decision trees, and Naive Bayesian methods (Nobata et al., 1999). However, the milestone publication by Lafferty et al. (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 (LeCun et al., 2015; Emmert-Streib et al., 2020). For instance, feedforward neural networks (FFNN) (Furrer et al., 2019), recurrent neural networks (RNN), or convolution neural

networks (CNN) (Zhu et al., 2017) 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).

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; Wang et al., 2018a; Giorgi and Bader, 2019; Ling et al., 2019; Weber et al., 2019; Yoon et al., 2019). 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 x1, x2, ..., xn as word representation (Habibi et al., 2017).

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

2. 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).

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), 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 was (Özgür et al., 2008). In contrast to this, the study by Yang et al. (2011) used a similar SVM model, however, for identifying the polarity of food-disease associations. In Jensen et al. (2014), 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 in Bundschus et al. (2008) 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; Zhang et al., 2019b). The feature inputs to DL models may include sentence level, word-level, and lexical-level features represented as vectors (Zeng et al., 2014), 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.

3. KNOWLEDGE GRAPH

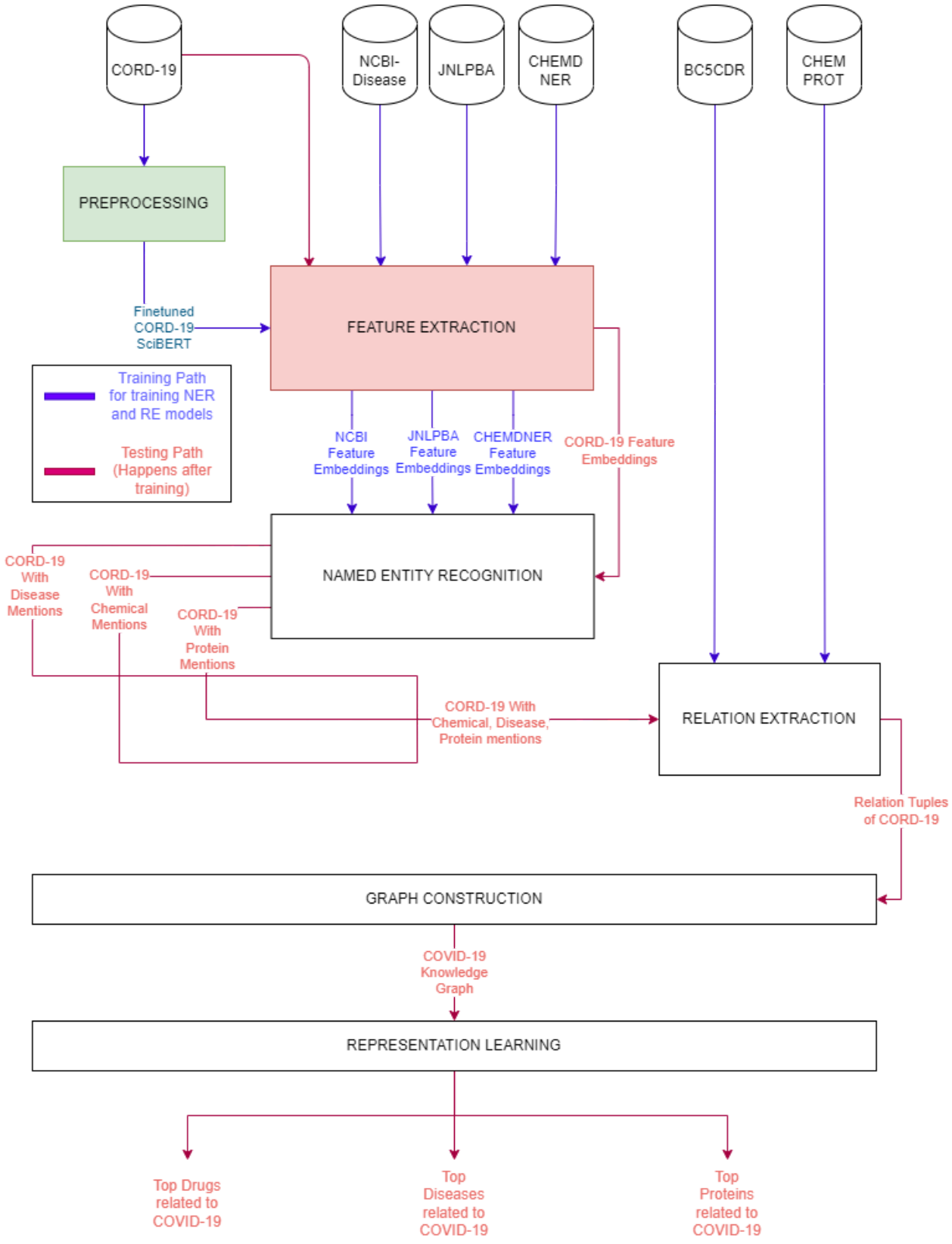
In the paper PharmKG: a dedicated knowledge graph benchmark for biomedical data mining briefings in bioinformatics, they extract knowledge graphs from drugs, diseases and genes databases and also their relations. Its advantage is that a large knowledge graph is obtained. But the data is highly generic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

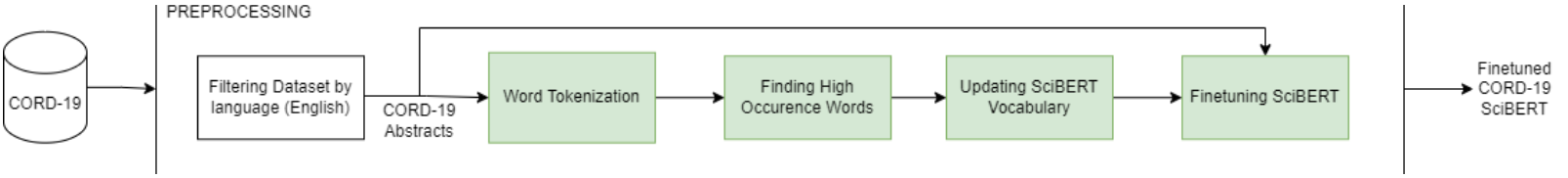
SYSTEM ARCHITECTURE



DETAILS OF MODULE DESIGN INDICATING THE INTERMEDIATE DELIVERABLES

1. PREPROCESSING MODULE

Input - CORD-19 Dataset
Output - Fine tuned CORD-19 SciBERT



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 are found. Then the words with occurrence count greater than 450 are taken. There are about 6000 words in this stage.

Then the **allenai/scibert_scivocab_uncased** 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.

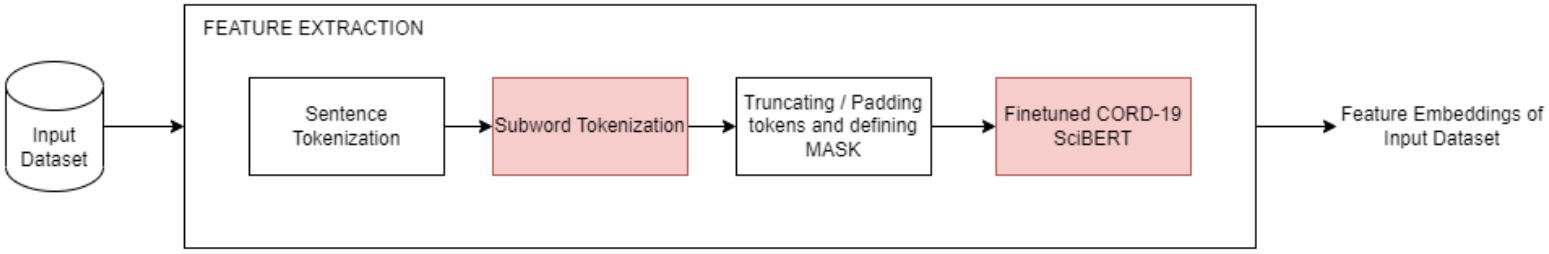
This updated vocabulary leads to updated subword tokenizer which will be used for fine tuning. Then the CORD-19 abstracts are taken and are subword tokenized using the new tokenizer and the subwords are converted into input ids and are padded to the length of 256.

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.

2. FEATURE EXTRACTION

Input - NER training and testing datasets (NCBI Disease, JNLPBA, CHEMDNER, CORD-19)

Output - Feature Embeddings of input dataset



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.

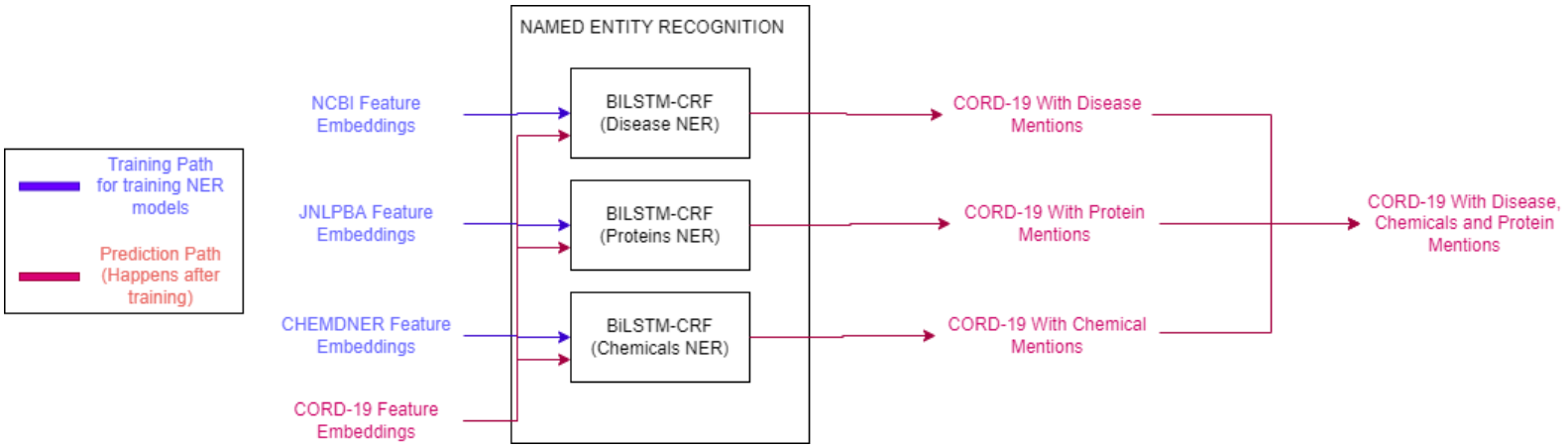
Finally the individual sentence of size (256) is passed to the fine tuned CORD-19 SciBERT model and the last hidden layer output is taken from the BERT model which is of the dimension 768.

The 768 dimension vector will be used as features for the Named Entity Recognition using BiLSTM-CRF.

3. NAMED ENTITY RECOGNITION

Input - Feature Embeddings of Training and Testing dataset(NCBI-Disease, JNLPBA, CHEMDNER, CORD-19)

Output - CORD-19 with disease, chemical and protein mentions.



In this module, the NCBI-Disease dataset is used for recognition of diseases. CHEMDNER dataset is used for recognition of Drugs. JNLPBA dataset is used for recognition of Proteins.

The CORD-19 SciBERT embeddings of each dataset are taken. Then the tokens and their corresponding named entity tags are associated.

Then each individual datasets are fed into a BiLSTM-CRF model, and the results are tested. The BiLSTM takes the BERT embeddings of 768 dimensions and outputs 512 dimension vectors. And Dropout is used to add regularization. Then the output is passed to a Fully connected layer and its output is passed to the CRF layer. The CRF layer finds the best transition from the previous to the next layer. The model uses Negative log-likelihood loss to optimize the model as a minimization problem.

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.
Example encoding,

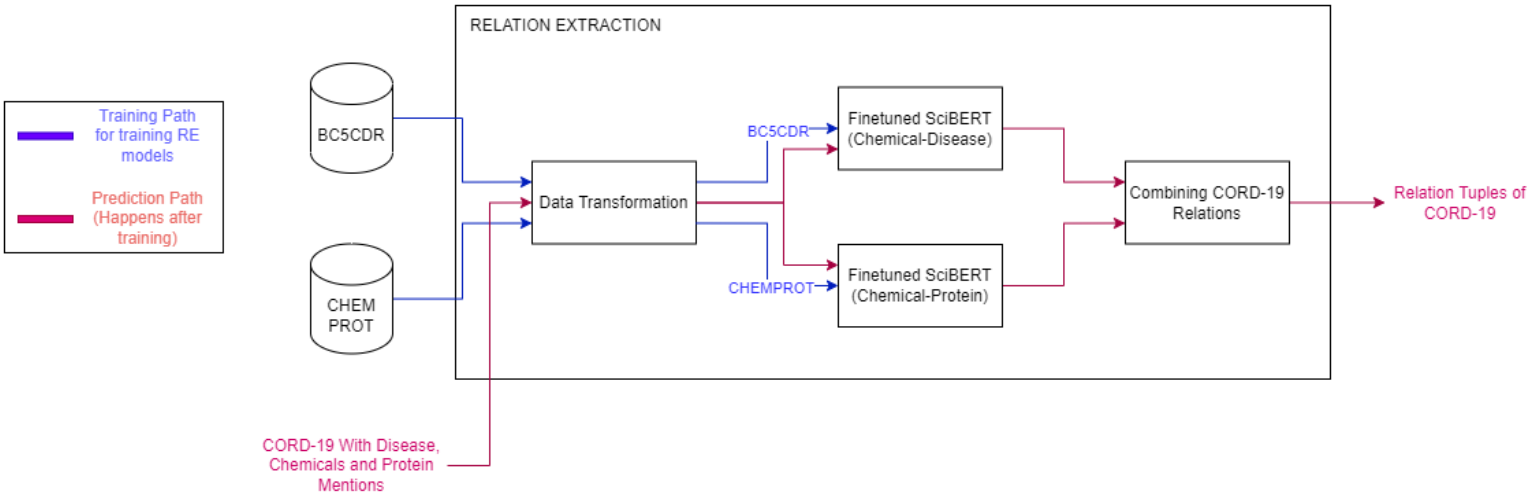
```
Management      0
of              0
critically      0
ill             0
patients        0
with            0
Severe          B-Disease
Acute           I-Disease
Respiratory     I-Disease
Syndrome        I-Disease
.               0
```

These Tags are combined so that the final output contains CORD-19 dataset with all the diseases, chemicals and proteins mentioned.

4. RELATION EXTRACTION MODULE

Input - BC5CDR, CHEMPROT, CORD-19 with entity mentions

Output - Relation Tuples of CORD-19



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 the sentences are processed as the first sentence contains the relation entities and the second sentence contains the text containing the relations.

The Drug-disease relations dataset is fed into the SciBERT model which produces the relations.

The CHEMPROT dataset is fed into an individual SciBERT model which will be finetuned for finding relations in that particular dataset. Finally the performance of the models are measured.

Now the CORD-19 dataset with entity mentions is preprocessed where only the sentences with two or more entities are forwarded into the model. 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) or (Entity1, Entity2, Relation).

5. GRAPH CONSTRUCTION MODULE

Input - CORD-19 Relations along with their entities
Output - COVID-19 Knowledge Graph

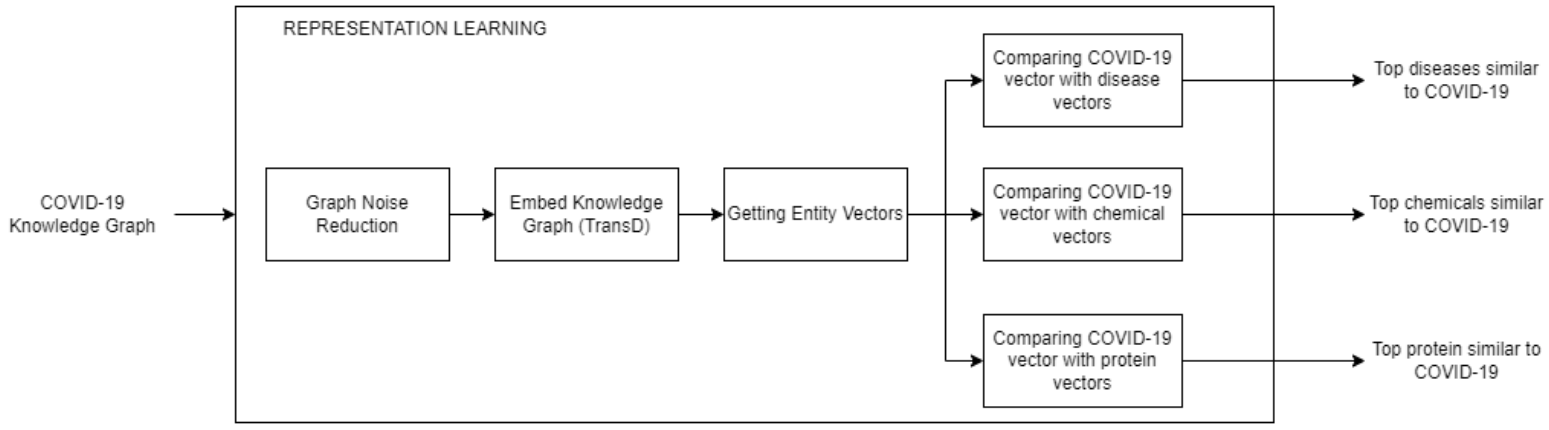
We construct a KG which is defined as KG = (E, R, G), where,

- E: a set of nodes representing disease/ protein/ drug entities.
- R: a set of labels representing chemical-protein relation or chemical-disease.
- G ⊆ E × R × E: a set of edges that represent facts connecting entity pairs.

Here entities with no relations or in-degree less than 5 (for example) can be removed which helps with the density of the resultant knowledge graph.

6. REPRESENTATION LEARNING MODULE

Input - COVID-19 Knowledge Graph
Output - Top Diseases, Chemicals, Proteins related to COVID-19



The noise of the input knowledge graph is reduced by removing entities with in-degree less than N. Then the knowledge graph is embedded using the geometric method TransD.

After embedding the Knowledge Graph, the vectors of all entities in the Knowledge graph are obtained. From that, the COVID-19 vector is taken.

This vector is compared with all the remaining vectors using cosine similarity and top COVID-19 related diseases, chemicals and proteins are found and are returned.

IMPLEMENTATION DETAILS (30%) AND THE CORRESPONDING RESULTS / SNAPSHOTS

1. PREPROCESSING MODULE

i. Filtering Dataset By Language (English)

Initially the CORD-19 dataset metadata is loaded as follows,

	paper_id	doi	title	abstract	publish_time	authors	journal	pdf_json_files
0	f9tg6xsg	10.1186/s40560-019-0415-z	Dexmedetomidine improved renal function in pat...	BACKGROUND: Dexmedetomidine has been reported ...	2020-01-02	Nakashima, Tsuyoshi; Miyamoto, Kyohei; Shima, ...	J Intensive Care	document_parses/pdf_json/44449ad1cca160ce491d7...
1	f73c639r	10.1186/s40635-019-0284-8	Aortic volume determines global end-diastolic ...	BACKGROUND: Global end-diastolic volume (GEDV)...	2020-01-02	Akohov, Aleksej; Barner, Christoph; Grimmer, S...	Intensive Care Med Exp	document_parses/pdf_json/def41c08c3cb1b3752bcf...
2	1qgpa45q	10.1186/s12864-019-6400-z	Whole genome sequencing and phylogenetic analy...	BACKGROUND: Human metapneumovirus (HMPV) is an...	2020-01-02	Kamau, Everlyn; Oketch, John W.; de Laurent, Z...	BMC Genomics	document_parses/pdf_json/f5ae3f66face323615df3...

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.

	paper_id	metadata	title	abstract	body_text	lang
0	44449ad1cca160ce491d7624f8ae1028f3570c45	{'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
1	def41c08c3cb1b3752bcff34d3aed7f8486e1c86	{'title': 'Aortic volume determines global end...	Aortic volume determines global end- diastolic...	Background: Global end-diastolic volume (GEDV)...	Transpulmonary thermodilution is commonly used...	en
2	f5ae3f66face323615df39d838e056ab5fcc98df	{'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

Then all the rows with languages other than english are removed, the dataset rows before and after filtering are as follows,

```
print('No of rows before language filter : ',len(meta_df))
meta_df = meta_df[meta_df['lang'] == 'en']
print('No of rows after language filter : ',len(meta_df))
```

```
No of rows before language filter :  229777
No of rows after language filter :  221520
```

ii. Word Tokenization

Then the abstracts from the CORD-19 dataset is taken and is cleaned of all punctuations and is word tokenized. This is an sample abstract.

```
cord_uid ug7v899j
sha d1aafb70c066a2068b02786f8929fd9c900897fb
source_x PMC
title Clinical features of culture-proven Mycoplasma pneumoniae infections at King Abdulaziz University Hospital, Jeddah, Saudi Arabia
doi 10.1186/1471-2334-1-6
abstract OBJECTIVE: This retrospective chart review describes the epidemiology and clinical features of 40 patients with culture-proven Mycop
lasma pneumoniae infections at King Abdulaziz University Hospital, Jeddah, Saudi Arabia. METHODS: Patients with positive M. pneumoniae cultur
es from respiratory specimens from January 1997 through December 1998 were identified through the Microbiology records. Charts of patients we
re reviewed. RESULTS: 40 patients were identified, 33 (82.5%) of whom required admission. Most infections (92.5%) were community-acquired. Th
e 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 mo
st 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 m
alaise (58.8%) were the most common symptoms, and crepitations (60%), and wheezes (40%) were the most common signs. Most patients with pneumo
nia had crepitations (79.2%) but only 25% had bronchial breathing. Immunocompromised patients were more likely than non-immunocompromised pat
ients 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 comorbidit
ies. 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.
publish_time 2001-07-04
authors Madani, Tariq A; Al-Ghamdi, Aisha A
journal BMC Infect Dis
url https://www.ncbi.nlm.nih.gov/pmc/articles/PMC35282/
```

After word tokenization, the output is as follows,

```
['objective', 'this', 'retrospective', 'chart', 'review', 'describes', 'the', 'epidemiology', 'and', 'clinical', 'features', 'of', '40', 'pat
ients', 'with', 'culture-proven', 'mycoplasma', 'pneumoniae', 'infections', 'at', 'king', 'abdulaziz', 'university', 'hospital', 'jeddah', 's
audi', 'arabia', 'methods', 'patients', 'with', 'positive', 'm', 'pneumoniae', 'cultures', 'from', 'respiratory', 'specimens', 'from', 'janua
ry', '1997', 'through', 'december', '1998', 'were', 'identified', 'through', 'the', 'microbiology', 'records', 'charts', 'of', 'patients', 'w
ere', 'reviewed', 'results', '40', 'patients', 'were', 'identified', '33', '82', '5', 'of', 'whom', 'required', 'admission', 'most', 'infecti
ons', '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', 'th
e', 'fall', '35', 'and', 'spring', '30', 'more', 'than', 'three-quarters', 'of', 'patients', '77', '5', 'had', 'comorbidities', 'twenty-fou
r', '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', 'symptom
s', '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', 'lik
ely', 'than', 'non-immunocompromised', 'patients', 'to', 'present', 'with', 'pneumonia', '8', '9', 'versus', '16', '31', 'p', '0', '05', 'o
f', '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', 'wer
e', 'more', 'common', 'in', 'infants', 'and', 'preschool', 'children', 'and', 'that', 'the', 'mortality', 'rate', 'of', 'pneumonia', 'in', 'p
atients', 'with', 'comorbidities', 'was', 'high']
```

iii. Finding High Occurrence Words

Then the occurrence count of all words in the abstracts are found and the words with occurrence count greater than 450 will be taken. Some of the top high occurrence words are as follows,

```
The length of the vocabulary is  6968
['chart', 'describes', 'epidemiology', 'mycoplasma', 'pneumoniae', 'university', 'saudi', 'arabia', 'cultures', 'specimens', '1998', 'microbi
ology', 'records', 'charts', 'reviewed', '33', '82', 'whom', '92', 'community-acquired', 'infants', '32', 'fall', '35', 'spring', '77', 'como
rbidities', 'isolates', 'upper', 'tract']
```

iv. Updating SciBERT vocabulary

The **allenai/scibert_scivocab_uncased** bert model is downloaded. Then the vocabulary of the scibert is updated with the new words. The older vocabulary and newer vocabulary count are as follows,

```python
print("Old vocabulary length : ",len(tokenizer))
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
```

Some of the newly added vocabulary words are as follows,

covid19, coronavirus-2, betacoronavirus,antivirals etc.,

v. Fine tuning SciBERT

The abstracts are taken and are subword tokenized using SciBERT tokenizer and the abstract is truncated or padded to 256 block size and the input to bert is created. Sample abstract SciBERT input is as follows,

```
{'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,
          563,   241,   755,  1495,   121,  6548,  3291,   305,   137, 31348,
         2694,  1808,  1931,   305,   256,  4118,   996,   579,  5194,   563,
          241,   755,  1495,   121,   111,  3913,  2638,   137,  5249,  1339,
          475,   506,   874,   579, 11124, 30113,   131,   568,  7466,   305,
          883, 17852,  8434,   579,  1379,  6197,  2242,   267,  1111,   190,
         9520,  1128,  2638,   190,   504, 31829,   182, 31415,   316,  6449,
         5352,   137,   170,   305,   190, 31092, 17431,  8707,   305, 10551,
         4710,   137,  1774, 12937, 30107,  4878,   493,   267,   111,   755,
         1495,  3049,   137,  1443, 16380,   288,  2242,   137,   330, 11436,
          123,  1921,   267,   111,   755,  1495,  6482,   755,   568,   190,
         9520,   883,  1443, 16380,   288,  8455,   170,   563,   617,  1552,
          883, 19440, 14641, 31093,   568,   267,   475,  1987,   506,   699,
          579, 31093,   568,   147,   709,   190,  9520,   493,   514,  3304,
         1107,  3877,   118,   244, 10764,   131,   111,  1540,   568,   190,
         9520,  1128,  4878,   239,   883, 24492, 13802,   161,  4266,   286,
         1107,   450,  8498,   982,   693,  4929,   239,   760,   305,  9761,
          923,   131,   127, 17119,  2486,   103], dtype=torch.int32)}
```

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.

```
[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})
```

Then the CORD-SciBERT model is obtained which will be used for the feature extraction module.

2. FEATURE EXTRACTION MODULE

i. Loading the dataset

The dataset is loaded from the files. The sentences and their tags are loaded from the dataset.

Chronic	O
administration	O
of	O
haloperidol	B-Chemical
increased	O
Dpp6	O
expression	O
in	O
mouse	O
brains	O
.	O

ii. Sentence Tokenization

The datasets are sentence tokenized and their tags are loaded.

```
['OBJECTIVE: This retrospective chart review describes the epidemiology and clinical features of 40 patients with culture-proven Mycoplasma p
neumoniae 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 t
hrough 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 bronchioliti
s.',
 'Cough (82.5%), fever (75%), and malaise (58.8%) were the most common symptoms, and crepitations (60%), and wheezes (40%) were the most comm
on 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 bec
ause 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 chi
ldren and that the mortality rate of pneumonia in patients with comorbidities was high.']
```

iii. Subword Tokenization

Then the sentence is subword tokenized using the 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', '.']
```

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.

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

Then the label tag is encoded to integer and the feature embedding for the input is generated.

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

iv. Fine tuned SciBERT

Then the dataset row is passed to the CORD-19 SciBERT model and the embedding of the input is generated.

```
['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])
```

3. NAMED ENTITY RECOGNITION MODULE

In this module, we have defined the model. But we haven't fine tuned the model hyperparameters. The model is as follows,

```
class BiLSTM_CRF:
  def forward(self, ids, mask,
token_type_ids, target_tag):
    x, _ = self.bert(ids,
           attention_mask=mask,
           token_type_ids=token_type_ids)

    h, _ = self.bilstm(x)

    o_tag = self.dropout_tag(h)

    tag = self.hidden2tag_tag(o_tag)

    mask = torch.where(mask==1, True,
           False)

    loss = - self.crf_tag(tag,
           target_tag, mask=mask,
           reduction='token_mean')

    return loss
```

METRICS FOR EVALUATION

Since Named Entity Recognition is a classification problem, where a token is classified as a particular named entity, classification performance metrics can be used here. Relation Extraction is also a classification problem so the same measures can be used here as well. They are,

1. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

2. Recall is the ratio of correctly predicted positive observations to the all observations in actual class.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

3. F1-Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

Since, the CORD-19 dataset doesn't possess any ground truth information. The relations extracted can only be verified via fact-checking known websites like [The Comparative Toxicogenomics Database | CTD (ctdbase.org)](ctdbase.org).

Also the system can be evaluated by fact-checking the final found COVID-19 related entities by manually checking the relation between them.

TEST CASES

1. PREPROCESSING MODULE

EXPECTED OUTPUT	ACTUAL OUTPUT
Expect only non-english rows to be removed	==Error TestCases== Mycobacterium tuberculosis (M.tb) is responsible for more deaths globally than any other pathogen. T ca - Catalan Background: Salmonella enterica serovar Typhi (S. Typhi) is a highly invasive bacterium that infects it - Italian

2. FEATURE EXTRACTION MODULE

EXPECTED OUTPUT	ACTUAL OUTPUT
Expect Similarity between “coronavirus” and “computer” to be 0.19833344	0.6503298

REFERENCES

1. Ayoub Harnoune, Maryem Rhanoui, Mounia Mikram, Siham Yousfi, Zineb Elkaimbillah, Bouchra El Asri, 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, https://doi.org/10.1016/j.cmpbup.2021.100042.
2. Minsoo Cho, Jihwan Ha, Chihyun Park, Sanghyun Park, Combinatorial feature embedding based on CNN and LSTM for biomedical named entity recognition, Journal of Biomedical Informatics, Volume 103, 2020, 103381, ISSN 1532-0464, https://doi.org/10.1016/j.jbi.2020.103381.

3. Perera, N., Dehmer, M., & Emmert-Streib, F. (2020). Named Entity Recognition and Relation Detection for Biomedical Information Extraction. Frontiers in cell and developmental biology, 8, 673. https://doi.org/10.3389/fcell.2020.00673
4. 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.
5. Kim, T.; Yun, Y.; Kim, N. Deep Learning-Based Knowledge Graph Generation for COVID-19. Sustainability 2021, 13, 2276. https://doi.org/10.3390/su13042276.
6. 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,.
7. 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.
8. Kim, T.; Yun, Y.; Kim, N. Deep Learning-Based Knowledge Graph Generation for COVID-19. Sustainability 2021, 13, 2276. https://doi.org/10.3390/su13042276
9. W. E. Zhang and Q. Nguyen, "Constructing COVID-19 Knowledge Graph from A Large Corpus of Scientific Articles," 2021 IEEE International Conference on Big Knowledge (ICBK), 2021, pp. 237-244, doi: 10.1109/ICKG52313.2021.00040.
10. Minsoo Cho, Jihwan Ha, Chihyun Park, Sanghyun Park, Combinatorial feature embedding based on CNN and LSTM for biomedical named entity recognition, Journal of Biomedical Informatics, Volume 103, 2020, 103381, ISSN 1532-0464, https://doi.org/10.1016/j.jbi.2020.103381.
11. Peng, Y., Wei, CH. & Lu, Z. Improving chemical disease relation extraction with rich features and weakly labeled data. J Cheminform 8, 53 (2016). https://doi.org/10.1186/s13321-016-0165-z
12. Nada Boudjellal, Huaping Zhang, Asif Khan, Arshad Ahmad, "Biomedical Relation Extraction Using Distant Supervision", Scientific Programming, vol. 2020, Article ID 8893749, 9 pages, 2020. https://doi.org/10.1155/2020/8893749.

13. Doğan RI, Leaman R, Lu Z. 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.
14. 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. The CHEMDNER corpus of chemicals and drugs and its annotation principles. J Cheminform. 2015Jan19;7(Suppl 1 Text mining for chemistry and the CHEMDNER track):S2. doi: 10.1186/1758-2946-7-S1-S2. PMID: 25810773; PMCID: PMC4331692.
15. Nigel Collier and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at jnlpba. In NLPBA/BioNLP.
16. Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF Models for Sequence Tagging. ArXiv, abs/1508.01991.
17. Ma, X., & Hovy, E.H. (2016). End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. ArXiv, abs/1603.01354.
18. Su, P., Peng, Y., & Vijay-Shanker, K. (2021). Improving BERT Model Using Contrastive Learning for Biomedical Relation Extraction. BIONLP.