Department of Computer Science and Engineering,
ANNA UNIVERSITY CHENNAI - 600025

CS6301 – Machine Learning Laboratory

Mini Project

Medical Information extraction and Disease Detection from Transcript Data using NLP

NAME	REG.NO	AADHAR NO	E-MAIL	MOBILE NO
Balaji S	2018103014	7948 7953 0072	balajithestar069@gmail.com	9384359450
Dhananjeyan AK	2018103523	4866 6460 5383	dhananjeyanak@gmail.com	9791652659

INSTRUCTOR & MENTOR: **Dr AROCKIA XAVIER ANNIE R**

Contents

1. ABSTRACT

Text analytics is used to evaluate the effectiveness of medical treatments by comparing various diseases, their outcomes and mode of treatment. It delivers a report of which courses of action prove effective and associate the various side-effects of treatment. It helps in identifying common symptoms to aid diagnosis and determining the most effective drug compounds for treating large number of population. For example, United Health Care analysed its treatment record data to explore the outcomes of patient's group treated with different drug regimens for the same disease and determined ways to cut costs and deliver better medicine. It also has developed clinical profiles to give physicians information about their practice patterns and to compare these with those of other physicians and peer-reviewed industry standards. This idea of implementing a project in Medical data analytics was suggested to us and we implemented it in Jupyter Notebook.

As to what our project does, it will try to structure the unstructured data which is a typical medical transcription and make our NLP model to recognise the diseases from the given medical script. It is proceeded using latest NLP tool called Spacy with other required machine learning concepts.

2. INTRODUCTION

Text Analytics help to automate the discovery of data elements essential to the natural language processing models which can be used to discover new treatments. Standardization of clinical vocabulary and the sharing of data across organizations to enhance the benefits of healthcare data mining applications. NLP is fast emerging in the field of healthcare in various forms. In this project, we use NLP to find out the type of disease from the unstructured data obtained. Spacy is an industrial strength natural processing library in Python and has several neural models for tagging, entity recognising etc. This library was introduced under MIT license by founders of software company called Explosion. Unlike traditional NLP tools, Spacy focussed on providing software for production usage. It features a **Convolutional neural network** for text categorization, POS tagging and other text analysis tasks.

The reason for choosing this as our library for process is it has a full flow support for Natural language processing and has pipelines required for our project like entity recognition pipe…and has several visualisers to visualise our models.

There are 3 parts-(modules) in our project:-

1) Data pre-processing of tab separated text values of our transcript dataset
2) CNN Training of processed data using Spacy NLP.
3) Evaluation , Testing data given to trained model and classification and disease identification.

What is CNN and how it is applied to natural language?

In deep learning, a **convolutional neural network (**CNN**, or** ConvNet**)** is a class of deep neural networks, most commonly applied to analyzing visual imagery. For instance, CNN is used for applications such as image classification, facial recognition, object detection etc.It has been proved to be effective in NLP tasks like Sentence classification, Text classification, Machine translation etc.

CNN basically contains several layers like Convolutional layer followed by pooling layers followed by MLP . This can be used in sentence classification using Spacy with other libraries like pandas , GloVe and keras. Dataset is extracted, preprocessed ,trained with Spacy and result is analysed.

2.1 SPACY CNN

As suggested by many, Spacy excels at large scale information extraction tasks.

Spacy contains about 55 pipelines for about 17 languages for NLP. Named entity recognition from huge chunks of text can be processed utilising these pipelines. For texts more than 3000 words, traditional NLTK takes more time to process. But Spacy's API is simple, easy and time efficient. Its trained CNN models like 'en-core-web-sm','en-core-web-md' etc. are highly useful in our project domain. We tend to use several pipelines of Spacy in our project. Also, spaCy supports custom models for TensorFlow, PyTorch and other frameworks. DisplaCy, a tree visualiser for dependency parse tree comes prebuilt with this library which we make use of.

3. LITERATURE SURVEY

S.NO	PAPER	ADVANTAGES & TAKEAWAYS	DISADVANTAGES
1.	Biomedical named entity recognition at Scale – by Veysel Cocaman @ John snow Labs	Processing of Electronic health records and finding named entities which are diseases	Traditional LSTM model in Spark which is not efficient in some cases.
2.	SCI BERT: A Pretrained Language Model for Scientific Text -by Iz Beltagy Kyle Lo Arman Cohan Allen Institute for Artificial Intelligence, Seattle, WA, USA	Supervised training on corpus using NLP methods and using BERT pretraining.	Performance was poor compared to SoTA models.
3.	Disease Detection in Medical Prescriptions Using Data Mining Tools -by University of Tehran	Using neural networks, SVM's to classify preprocessed transcriptions using Case based reasoning	Preprocessing is not present and have already been done in form of csv. So, no NLP used .
4.	Text Analytics & NLP in Healthcare: Applications & Use Cases – By Andrea Kulkarni in Lexalytics	Brief analysis of applications of NLP on healthcare systems	No detailed specifications about recognition of diseases from scripts.
5.	NLP &Healthcare:Understanding the Language of Medicine – Curai Tech Blog	Brief analysis about medical entity recognition applications	LSTM , softmax approach which tends to be less effective for time being

4.PROBLEM STATEMENT

To arrive with a entity identification tool which should be capable of recognising Natural language and find diseases from a medical transcription data.

Given a medical transcription data from any physician / organisation like ,

```
SUBJECTIVE:, This 23-year-old white female presents with complaint of allergies. She used to
have ...etc.. etc….
```

The model can predict the disease of the patient from the given transcription. This can prove useful in **Digitalisation of data** and better processing of medical data and biomedical text analytics.

5. PROBLEM SOLUTION

5.1 DATA SET:-

The Data Set has been generated by two ways:

- BC5CDR diseases Dataset (in form of TSV) (Chemical Disease Relation Task)
- NCBI disease dataset (Annotated disease mentions in form of TSV)

5.2 DESCRIPTION OF DATASETS:-

The dataset is used for the **Bio Creative V Chemical Disease Relation (CDR)** Task. It contains **1,500 titles** and abstracts from PubMed, where chemical and disease mentions are annotated by human. Following previous studies , we only use the subset with chemical entities and denote is as BC5CDR-chemical.NCBI-disease NCBI-disease contains **793 PubMed abstracts** which are annotated with disease mentions and their corresponding concepts. There are **6,892 disease** mentions from **790 unique disease** concepts in this dataset and 91% of the mentions were mapped to a single disease concept. It has been a widely used research resource for the disease NER.

5.3 APPROACH:-

Firstly, the datasets which are in text form are preprocessed into a way that Spacy can recognise them. Which are of String,{entity} pairs. Then the processed data is feed onto Spacy model pipelined with NER which means Named entity recognition model and trained on the data. The trained model is then evaluated and tested with testing data to recognise diseases.

6. NOVELTY

Data Analytics on any unstructured data is always a task comprising implementation challenges. **Our deployment helps in achieving faster results on large chunks of data utilising spaCy Library.** Our novelty lies on finding a recogniser with higher accuracy and faster model compared to Traditional LSTM. Existing NER models like LSTM, NLTK (*Disease Detection in Medical Prescriptions Using Data Mining Tools -by University of Tehran*) can fail on some cases on testing. Our SpaCy model which is still in blooming phase among Big Data analysts can prove better and can be replacement and revolution on Text analytics. Our model can technically be used as a multi domain recognizer which can help on Biomedical identification and Digitalisation of documents. We also tried to achieve above 93 percent accuracy which is a good start for a developing tool like spaCy. Having conditions of optimal dropout and other hyperparameters, we try to achieve higher accuracy than what we referred for our literary survey (*Biomedical named entity recognition at Scale – by Veysel Cocaman @ John snow Labs* &
Disease Detection in Medical Prescriptions Using Data Mining Tools -by University of Tehran) etc.

7. ARCHITECTURE

7.1 ABSTRACT ARCHITECTURE

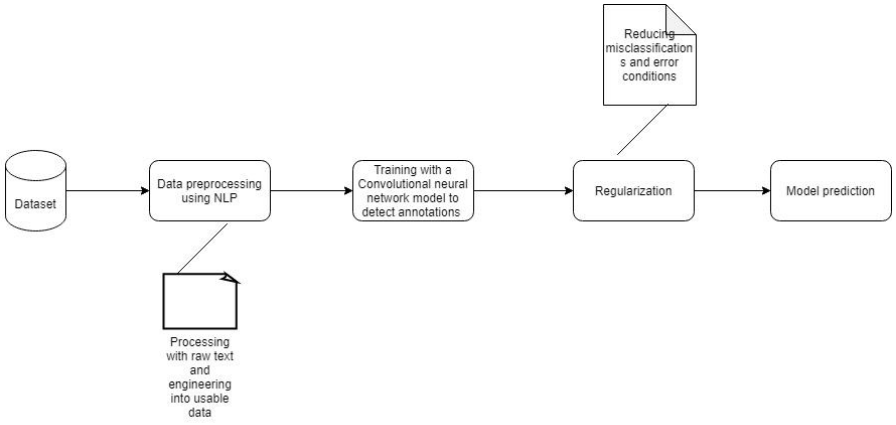
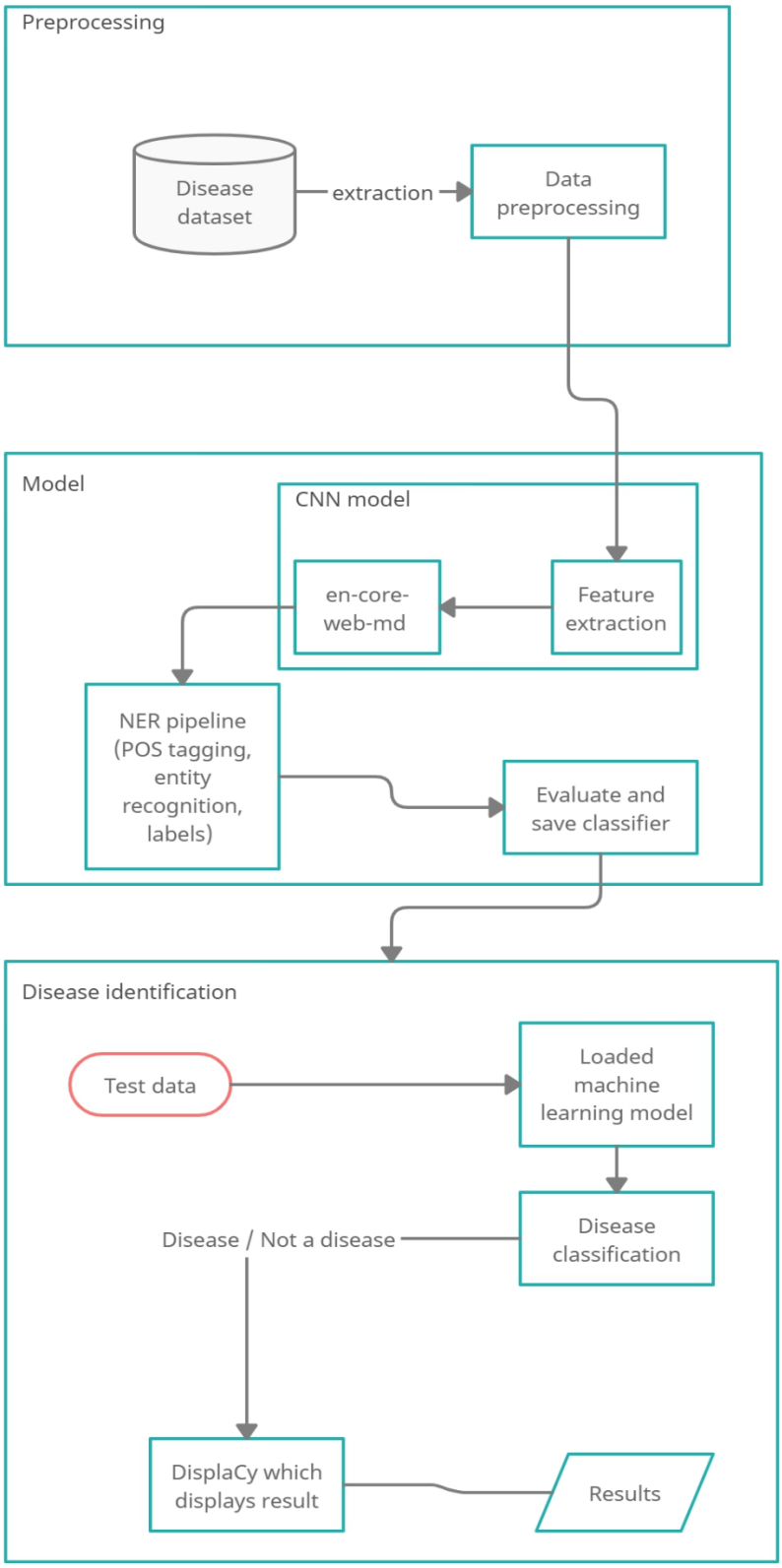


Fig 1.Abstract Architecture diagram

7.2 DETAILED ARCHITECTURE:

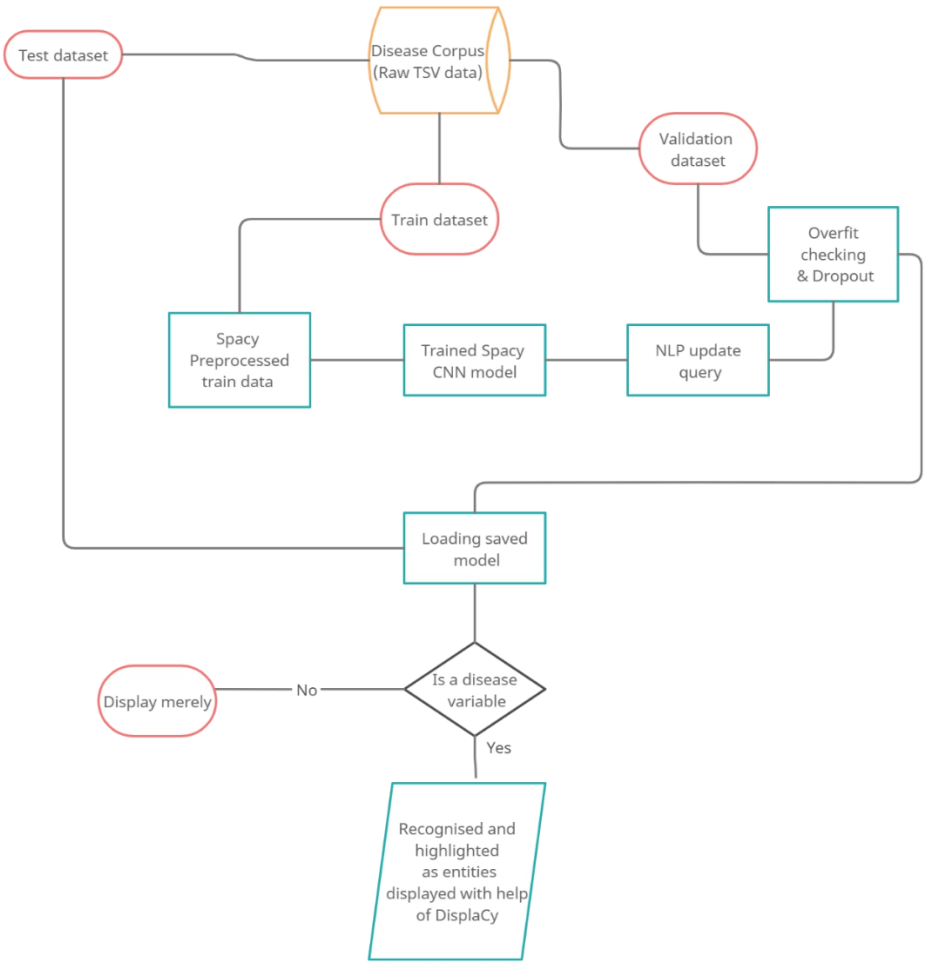


7.2 EXPLANATION OF THE ARCHITECTURE:

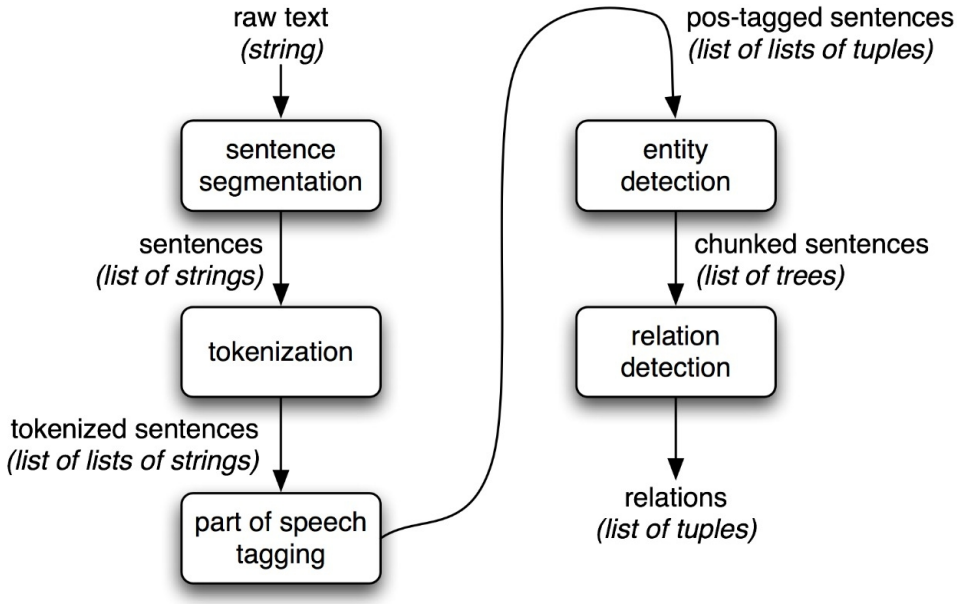
Text Analytics help to automate the discovery of data elements essential to the **natural language processing models** which can be used to discover new treatments. **Standardization of clinical vocabulary** and the sharing of data across organizations to enhance the **benefits of healthcare data mining applications**.We intend to use an advanced NLP library known as **SpaCy** which is used for production use which we use for information extraction and pre-processing. Then a **CNN model is used to train** and finally diseases are recognized from given transcription text.

7.3 DATAFLOW DIAGRAM:

7.3.1 OVERALL FLOW:



7.3.2 CNN PART FLOW:



7.4 EXPLANATION OF DATAFLOW DIAGRAM:

Initially, our dataset is preprocessed to SpaCy accepted format , then trained on en-core-web-md English trained CNN model under optimal hyperparameters. The trained model is evaluated and stopped at optimal conditions and saved to disc and later tested with test data for identification of diseases.

CNN part of SpaCy contains processes of sentence segmentation, tokenization, POS tagging ,entity detection etc powered by several pipelines. The neural networks weights are updated according to entity presence in the sentence.

8. DETAILED MODULE DESIGN:

The project has a total of 3 modules with data preprocessing,training and loading the model:

- Data pre-processing of tab separated text values.
- CNN Training of processed data using Spacy NLP.
- Testing data given to trained model and classification and disease identification.

Detailed description of each module along with pseudo code is presented below.

8.1 Data pre-processing of tab separated text values:

PSEUDOCODE:

Import required modules

Path=provided path

Read both datasets

Opens the file in path and cleansing data by removing junks(\n , tabs)

Make unique labels

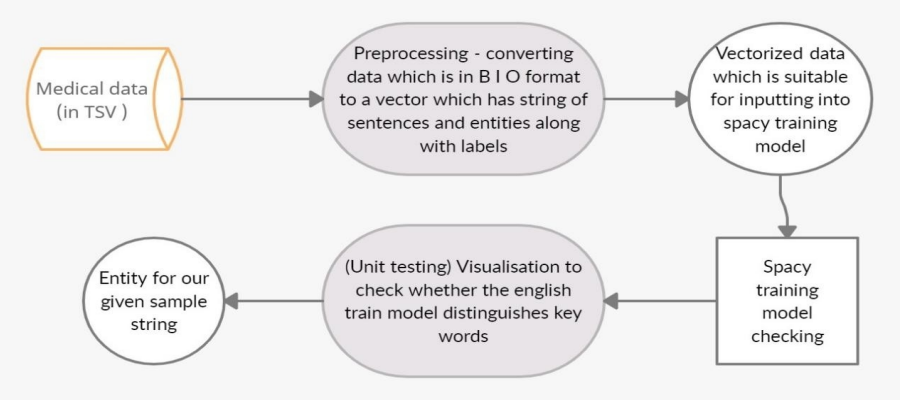
Sentences separated from entities [key,values] . append to train data

The BC5CDR disease dataset consists of sentences which are separated by tabs. The strings are annotated using B , I ,O which represents beginning , intermediate and final words in string. This will be of form word \t label \n.We need to convert this to a format that spacy model can be trained Which is an unstructured text which need some NLP fixings. This function load_data basically performs that preprocessing step of converting the raw data to a vector which contains string, entities pairs.

INPUT: Unstructured raw data(tab separated)

OUTPUT: Vectorized data which contains string,entity pairs

MODULE ARCHITECTURE:



8.2 CNN Training of processed data using Spacy NLP:

PSEUDOCODE:

Def train_model Arguments: (training data , labels, iterations, dropout, freq){

Loading 'en-core-web-md' model

If loaded spacy = blank {

Load blank model }

Search for NER pipe

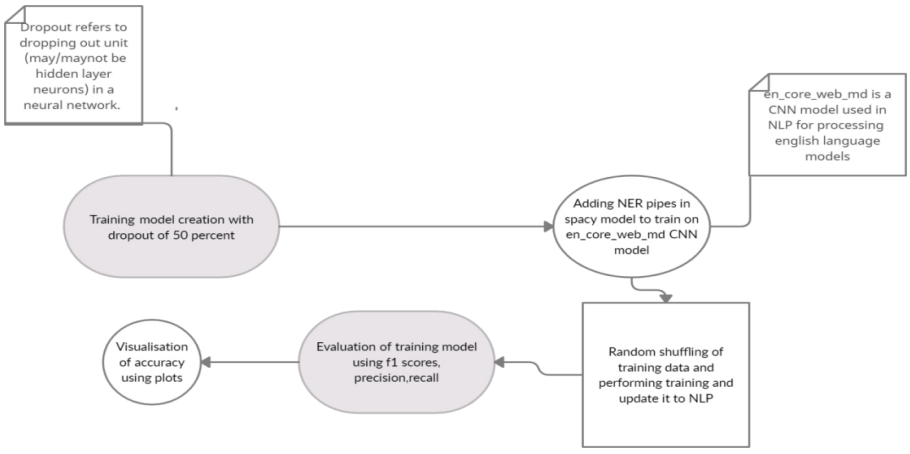
Random shuffle of train data and updation is done

Train model evaluated and stored in disc

}

Precision,recall and f1 calculations and drop out is set to 0.5.Drop out is a regularization method that approximates training a large number of neural networks. The default interpretation of the dropout hyper parameter is the probability of training a given nodein a layer, where 1.0 means no dropout, and 0.0 means no outputsfrom the layer. A good value for dropout in a hidden layer is between 0.5 and 0.8. Input layers use a larger dropout rate, such asof 0.8.Train data,iterations,dropout and display frequencies are passed inthe train_spacy function.en_core_web_md is loaded into nlp.It is used in types of vocabulary,syntax,entities,vectors which is mostlyused in formats of written text.md refers to the size of the datasetwhich is medium.

MODULE ARCHITECTURE:



8.3 Testing data given to trained model and classification and disease identification:

Then, the trained model is loaded and tested with test data of our datasets. For each entities in the content of Bc5CDR dataset.

INPUT: Trained model

OUTPUT: Classification of diseases

9.IMPLEMENTATION:

9.1 INITIAL SETUP:

- Jupyter Notebook is installed.
- Spacy module is installed.
- Tensor Flow is installed.
- Python environment is created.

9.2 CODE SNIPPETS:

9.2.1 DATA PREPROCESSING:

```
In [1]: import spacy
        import csv
        import random
        import time
        import numpy as np
        import en_core_web_md
        from spacy.util import minibatch, compounding
        import sys
        from spacy import displacy
        from itertools import chain
        import matplotlib.pyplot as plt
        from matplotlib.ticker import MaxNLocator
```

```
mortality       O
in      O
patients        O
with    O
Parkinson       B
'       I
s       I
disease I
(       O
PD      B
)       O
randomized      O
to      O
receive O
10      O
mg      O
selegiline      O
per     O
day     O
and     O
L       O
-       O
dopa    O
compared        O
with    O
those   O
taking  O
L       O
```

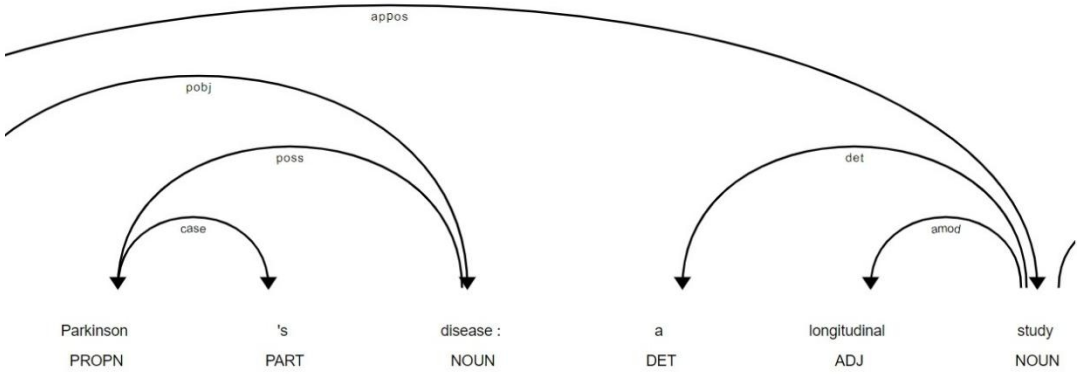
```
In [2]: tsv_file = open("BC5CDR-disease\\train.tsv")
        read_tsv = csv.reader(tsv_file, delimiter="\t")

        for row in read_tsv:
          print(row)
```

```
['Selegiline', 'O']
['-', 'O']
['induced', 'O']
['postural', 'B']
['hypotension', 'I']
['in', 'O']
['Parkinson', 'B']
["'", 'I']
['s', 'I']
['disease', 'I']
[':', 'O']
['a', 'O']
['longitudinal', 'O']
['study', 'O']
['on', 'O']
['the', 'O']
['effects', 'O']
['of', 'O']
['drug', 'O']
['withdrawal', 'O']
```

```
In [3]: def load_data(file_path):
            file = open(file_path, 'r')
            print(file)
            training_data, entity, sen, uniq_labels = [], [], [], []
            current_annotation = None
            start =0
            end = 0
            for line in file:
                line = line.strip("\n").split("\t")
                if len(line) > 1:
                    label = line[1]
                    if(label != 'O'):
                        label = line[1]+"_Disease"
                    word = line[0]
                    sen.append(word)
                    start = end
                    end += (len(word) + 1)

                    if label == 'I_Disease' :
                        entity.append(( start,end-1, label))

                    if label == 'B_Disease':
                        entity.append(( start,end-1, label))



                    if label != 'O' and label not in unique_labels:
                        uniq_labels.append(label)


                if len(line) == 1:
                    if(len(entity) > 0):
                        sen = " ".join(sen)
                        training_data.append([sentence, {'entity' : entity}])

                    end = 0
```

ound an increased mortality in patients with Parkinson ' s disease (PD) randomized to receive 10 mg selegiline per day and L - dopa compared with those taking L - dopa alone .", {'entities': [(32, 41, 'B_Disease'), (42, 43, 'I_Disease'), (44, 45, 'I_Disease'), (46, 53, 'I_Disease'), (132, 141, 'B_Disease'), (142, 143, 'I_Disease'), (144, 145, 'I_Disease'), (146, 153, 'I_Disease'), (156, 158, 'B_Disease')]}], ['Recently , we found that therapy with selegiline and L - dopa was associated with selective systolic orthostatic hypotension which was abolished by withdrawal of selegiline .', {'entities': [(92, 100, 'B_Disease'), (101, 112, 'I_Disease'), (113, 124, 'I_Disease')]}], ['The aims of this study were to confirm our previous findings in a separate cohort of patients and to determine the time course of the cardiovascular consequences of stopping selegiline in the expectation that this might shed light on the mechanisms by which the drug causes orthostatic hypotension .', {'entities': [(274, 285, 'B_Disease'), (286, 297, 'I_Disease')]}], ['METHODS : The cardiovascular responses to standing and head - up tilt were studied repeatedly in PD patients receiving selegiline and as the drug was withdrawn .', {'entities': [(97, 99, 'B_Disease')]}], ['RESULTS : Head - up tilt caused systolic orthostatic hypotension which was marked in six of 20 PD patients on selegiline , one of whom lost consciousness with unrecordable blood pressures .', {'entities': [(32, 40, 'B_Disease'), (41, 52, 'I_Disease'), (53, 64, 'I_Disease'), (95, 97, 'B_Disease')]}], ['A lesser degree of orthostatic hypotension occurred with standing .', {'entities': [(19, 30, 'B_Disease'), (31, 42, 'I_Disease')]}], ['Orthostatic hypotension was ameliorated 4 days after withdrawal of selegiline and totally abolished 7 days after discontinuation of the drug .', {'entities': [(0, 11, 'B_Disease'), (12, 23, 'I_Disease')]}], ['Stopping selegiline also significantly reduced the supine systolic and diastolic blood pressures consistent with a previously undescribed supine pressor action .', {'entities': [(39, 46, 'B_Disease'), (47, 50, 'I_Disease'), (51, 57, 'I_Disease'), (58, 66, 'I_Disease'), (67, 70, 'I_Disease'), (71, 80, 'I_Disease'), (81, 86, 'I_Disease'), (8

```
from spacy import displacy
displacy.render(doc, style='dep',jupyter=True)
```




```
In [7]: from spacy import displacy
        displacy.render(doc, style='dep',jupyter=True)
```

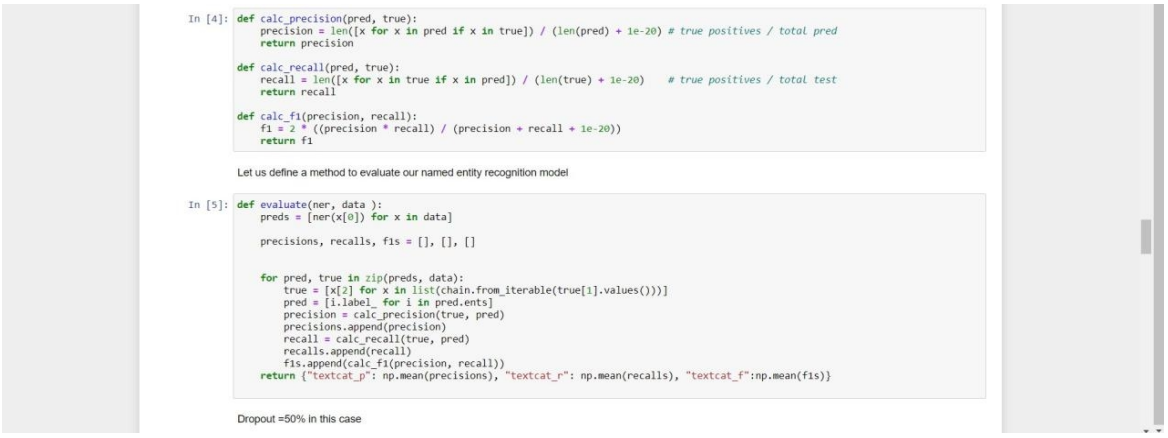
Selegiline - PROPN induced VERB postural ADJ hypotension NOUN in ADP Parkinson ' PROPN s PART disease : NOUN a DET longitudinal ADJ study NOUN on ADP the DET effects NOUN of ADP drug NOUN withdrawal NOUN npadvmod amod amod prep poss appos pobj det amod appos prep det pobj prep compound pobj

```
In [11]: for token in doc:
             if (token.dep_=='poss'):
                 print(token.text)
             elif (token.dep_=='pobj'):
                 print(token.text)
```

```
Parkinson
disease
effects
withdrawal
```

We can see that entities which we're required, Parkinson,withdrawal etc are recognised. We next use actual training data to train model to find disease variables and classify them.

9.2.2 Training processed data:



```
In [4]: def calc_precision(pred, true):
            precision = len([x for x in pred if x in true]) / (len(pred) + 1e-20) # true positives / total pred
            return precision

        def calc_recall(pred, true):
            recall = len([x for x in true if x in pred]) / (len(true) + 1e-20)    # true positives / total test
            return recall

        def calc_f1(precision, recall):
            f1 = 2 * ((precision * recall) / (precision + recall + 1e-20))
            return f1
```

Let us define a method to evaluate our named entity recognition model

```
In [5]: def evaluate(ner, data ):
            preds = [ner(x[0]) for x in data]

            precisions, recalls, f1s = [], [], []


            for pred, true in zip(preds, data):
                true = [x[2] for x in list(chain.from_iterable(true[1].values()))]
                pred = [i.label_ for i in pred.ents]
                precision = calc_precision(true, pred)
                precisions.append(precision)
                recall = calc_recall(true, pred)
                recalls.append(recall)
                f1s.append(calc_f1(precision, recall))
            return {"textcat_p": np.mean(precisions), "textcat_r": np.mean(recalls), "textcat_f":np.mean(f1s)}
```

Dropout =50% in this case

```
In [6]: def train_spacy(train_data, labels, iterations, dropout = 0.5, display_freq = 1):
            valid_fiscores=[]
            test_fiscores=[]
            nlp = spacy.load("en_core_web_md")
            #nlp = spacy.blank('en')
            if 'ner' not in nlp.pipe_names:
                ner = nlp.create_pipe('ner')
                nlp.add_pipe(ner)
            else:
                ner = nlp.get_pipe("ner")

            for i in labels:
                ner.add_label(i)

            other_pipes = [pipe for pipe in nlp.pipe_names if pipe != 'ner']
            with nlp.disable_pipes(*other_pipes):
                #nlp.vocab.vectors.name = 'spacy_model'
                optimizer = nlp.begin_training()
                for itr in range(iterations):
                    random.shuffle(train_data)
                    losses = {}
                    batches = minibatch(train_data, size = compounding(16.0, 64.0, 1.5))
                    for batch in batches:
                        texts, annotations = zip(*batch)
                        nlp.update(
                            texts,
                            annotations,
                            drop = dropout,
                            sgd = optimizer,
                            losses = losses)
                    scores = evaluate(nlp,VALID_DATA)
                    valid_fiscores.append(scores["textcat_f"])
                    print('==========================================')
                    print('Interation = '+str(itr))
                    print('Losses = '+str(losses))
```

9.2.3 Testing loaded models:

Loaded models:

model1

```
In [8]: ner = load_model("spacy_example")


        test_sentences = [x[0] for x in TEST_DATA[0:4000]]
        for x in test_sentences:
            doc = ner(x)
            for ent in doc.ents:
                print(ent.text, ent.start_char, ent.end_char, ent.label_)
            displacy.render(doc,jupyter=True, style = "ent")
```

System Testcase 1:

```
Torsade 0 7 B_Disease
de 8 10 I_Disease
pointes 11 18 I_Disease
ventricular 19 30 I_Disease
tachycardia 31 42 I_Disease
dilated 111 118 B_Disease
cardiomyopathy 119 133 I_Disease
congestive 138 148 B_Disease
heart 149 154 I_Disease
failure 155 162 I_Disease
```

Torsade B_DISEASE de I_DISEASE pointes I_DISEASE ventricular I_DISEASE tachycardia I_DISEASE during low dose intermittent dobutamine treatment in a patient with dilated B_DISEASE cardiomyopathy I_DISEASE and congestive B_DISEASE heart I_DISEASE failure I_DISEASE .

System TC02:

The authors describe the case of a 56 - year - old woman with chronic B_DISEASE , severe heart B_DISEASE failure I_DISEASE secondary to dilated B_DISEASE cardiomyopathy I_DISEASE and absence of significant ventricular B_DISEASE arrhythmias I_DISEASE who developed QT prolongation and torsade B_DISEASE de I_DISEASE pointes I_DISEASE ventricular I_DISEASE tachycardia I_DISEASE during one cycle of intermittent low dose (2 . 5 mcg / kg per min) dobutamine .

System TC03:

```
hyperammonemia 54 68 B_Disease
bacterial 99 108 B_Disease
infections 109 119 I_Disease
```

Higher plasma ammonium levels and more rapid onset of hyperammonemia B_DISEASE were seen in 18 patients with bacterial B_DISEASE infections I_DISEASE (p = 0 . 003 and 0 . 0006 , respectively) and in nine patients receiving high daily doses (2600 or 1800 mg / m2) of 5 - FU (p = 0 . 0001 and < 0 . 0001 , respectively) .

```
hyperammonemic 16 30 B_Disease
encephalopathy 31 45 I_Disease
```

In conclusion , hyperammonemic B_DISEASE encephalopathy I_DISEASE can occur in patients receiving continuous infusion of 5 - FU .

System TC04:

Meloxicam - induced liver toxicity .

```
rheumatoid 44 54 B_Disease
arthritis 55 64 I_Disease
acute 79 84 B_Disease
cytolytic 85 94 I_Disease
hepatitis 95 104 I_Disease
```

We report the case of a female patient with rheumatoid B_DISEASE arthritis I_DISEASE who developed acute B_DISEASE cytolytic I_DISEASE hepatitis I_DISEASE due to meloxicam .

System TC05:

```
NG 12 14 B_Disease
- 27 28 I_Disease
arginine 29 37 I_Disease
L 40 41 B_Disease
- 42 43 I_Disease
NOARG 44 49 I_Disease
catalepsy 112 121 B_Disease
```

RATIONALE : NG B_DISEASE - nitro - L - I_DISEASE arginine I_DISEASE (L B_DISEASE - I_DISEASE NOARG I_DISEASE) , an inhibitor of nitric - oxide synthase (NOS) , induces catalepsy B_DISEASE in mice .

```
catalepsy 84 93 B_Disease
```

Neuroleptic drugs such as haloperidol , which block dopamine receptors , also cause catalepsy B_DISEASE in rodents .

Checking a sample text from test data:

```
In [10]: ner = load_model("spacy_example")
         doc = ner("Selegiline - induced postural hypotension in Parkinson ' s disease : a longitudinal study on the effects of drug withd
         displacy.render(doc,jupyter=True, style = "ent")
```

Selegiline - induced postural B_DISEASE hypotension I_DISEASE in Parkinson B_DISEASE ' I_DISEASE s I_DISEASE disease I_DISEASE : a longitudinal study on the effects of drug withdrawal.The aims of this study were to confirm our previous findings in a separate cohort of patients and to determine the time course of the cardiovascular B_DISEASE consequences of stopping selegiline in the expectation that this might shed light on the mechanisms by which the drug causes orthostatic B_DISEASE hypotension I_DISEASE

DATASET 2:

NCBI DISEASE DATA:

SYSTEM TC06:

```
In [9]: ner = load_model("ncbi")



        test_sentences = [x[0] for x in TEST_DATA[0:4000]]
        for x in test_sentences:
            doc = ner(x)
            for ent in doc.ents:
                print(ent.text, ent.start_char, ent.end_char, ent.label_)
            displacy.render(doc,jupyter=True, style = "ent")
```

```
Torsade 0 7 B_Disease
de 8 10 I_Disease
pointes 11 18 I_Disease
ventricular 19 30 I_Disease
tachycardia 31 42 I_Disease
dilated 111 118 B_Disease
cardiomyopathy 119 133 I_Disease
congestive 138 148 B_Disease
heart 149 154 I_Disease
failure 155 162 I_Disease
```

Torsade B_DISEASE de I_DISEASE pointes I_DISEASE ventricular I_DISEASE tachycardia I_DISEASE during low dose intermittent dobutamine treatment in a patient with dilated B_DISEASE cardiomyopathy I_DISEASE and congestive B_DISEASE heart I_DISEASE failure I_DISEASE .

```
chronic 62 69 B_Disease
```

SYSTEM TC07:

```
coronary 31 39 B_Disease
artery 40 46 I_Disease
disease 47 54 I_Disease
dobutamine 57 67 B_Disease
induced 68 75 I_Disease
ischaemia 76 85 I_Disease
ventricular 123 134 B_Disease
dysfunction 135 146 I_Disease
```

CONCLUSIONS : In patients with coronary B_DISEASE artery I_DISEASE disease I_DISEASE , dobutamine B_DISEASE induced I_DISEASE ischaemia I_DISEASE results in prolonged reversible left ventricular B_DISEASE dysfunction I_DISEASE , presumed to be myocardial stunning , similar to that seen after exercise .

SYSTEM TC08:

```
hypertensive 72 84 B_Disease
psoriatic 85 94 I_Disease
gingival 201 209 B_Disease
hyperplasia 210 221 I_Disease
```

Our findings indicate that sustained - release nifedipine is useful for hypertensive B_DISEASE psoriatic I_DISEASE patients under long - term treatment with cyclosporin A , but that these patients should be monitored for gingival B_DISEASE hyperplasia I_DISEASE .

SYSTEM TC09:

```
psoriatic 9 18 B_Disease
hypertension 33 45 B_Disease
```

Thirteen psoriatic B_DISEASE patients with hypertension B_DISEASE during the course of cyclosporin A therapy were treated for 25 months with a calcium channel blocker , sustained - release nifedipine , to study the clinical antihypertensive effects and adverse events during treatment with both drugs .

SYSTEM TC10:

```
muscle 108 114 B_Disease
fasciculations 115 129 I_Disease
```

Alfentanil 50 micrograms kg - 1 effectively inhibits the incidence and intensity of suxamethonium - induced muscle B_DISEASE fasciculations I_DISEASE ; moreover , intragastric pressure remains at its control value .

We can see that **Intragastric pressure is not recognized.**

Checking with a sample text:

```
In [13]: ner = load_model("ncbi")
         doc = ner("Identification of APC2 , a homologue of the adenomatous polyposis coli tumour suppressor .For females there was a high
         displacy.render(doc,jupyter=True, style = "ent")
```

Identification of APC2 , a homologue of the adenomatous B_DISEASE polyposis I_DISEASE coli I_DISEASE tumour I_DISEASE suppressor .For females there was a high risk of endometrial B_DISEASE cancer I_DISEASE (0 . 5 at age 60 years) and premenopausal B_DISEASE ovarian I_DISEASE cancer I_DISEASE (0 . 2 at 50 years).

10. RESULTS AND COMPARISON:

10.1 Test Cases:

TC 1:

10 ITERATIONS :

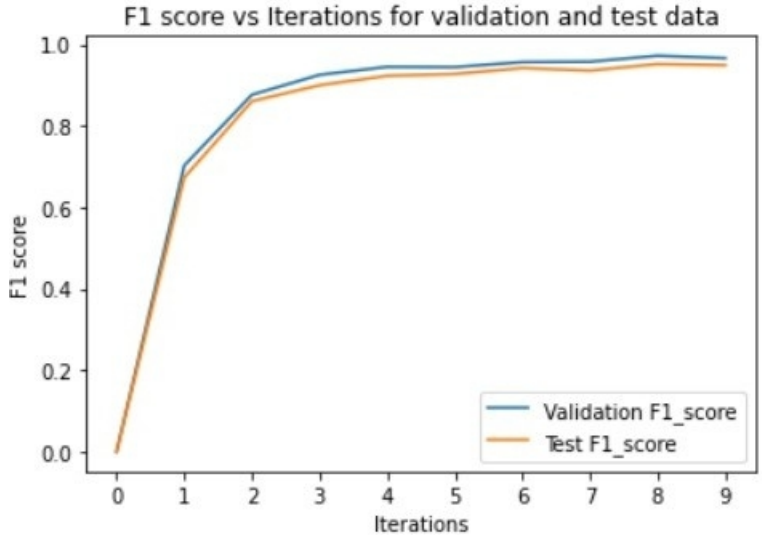
DROPOUT: 40 %

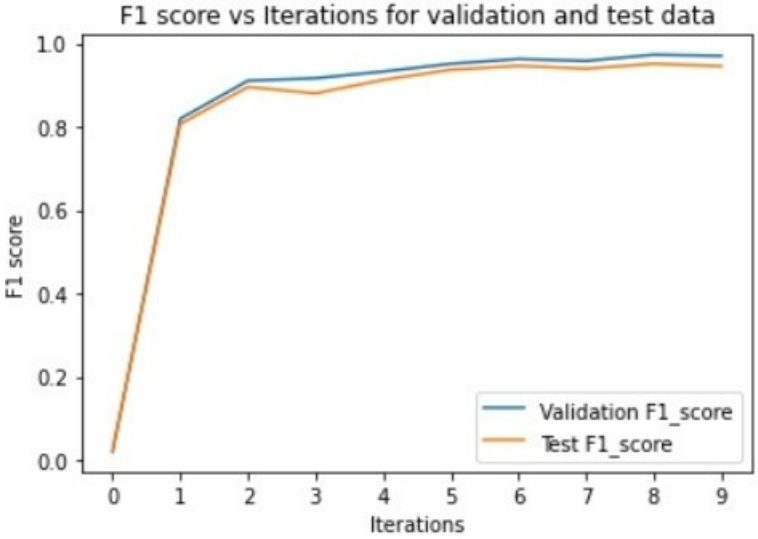
DATASET 1:

```
F1-score = 0.9725230503992354
Precision = 0.9762875890879994
Recall = 0.9761604700496845
==============TEST DATA=======================
F1-score = 0.9521196490584244
Precision = 0.9536376888417704
Recall = 0.9622139764996908
======================================
======================================
Interation = 9
Losses = {'ner': 44492.48299217224}
==============VALID DATA=======================
F1-score = 0.9664957798103787
Precision = 0.9674432209382385
Recall = 0.9715067269580752
==============TEST DATA=======================
F1-score = 0.9492586778301063
Precision = 0.9462165385634773
Recall = 0.9611935683364253
======================================
```

DATASET 2:

```
Precision = 0.9769028312770985
Recall = 0.9743667308658517
==============TEST DATA=======================
F1-score = 0.9506595136465267
Precision = 0.952122537326619
Recall = 0.9608313455252231
======================================
======================================
Interation = 9
Losses = {'ner': 44677.79912185669}
==============VALID DATA=======================
F1-score = 0.9696875958099322
Precision = 0.9706161958726437
Recall = 0.9750683860883158
==============TEST DATA=======================
F1-score = 0.9448884448884448
Precision = 0.9420730629914303
Recall = 0.9577922077922078
======================================
```


F1 score vs Iterations for validation and test data
F1 score
Iterations
Validation F1_score
Test F1_score


F1 score vs Iterations for validation and test data
F1 score
Iterations
Validation F1_score
Test F1_score

TC 2:

ITERATIONS:10

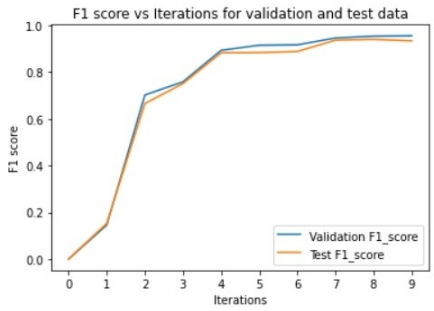
DROPOUT : 50 %

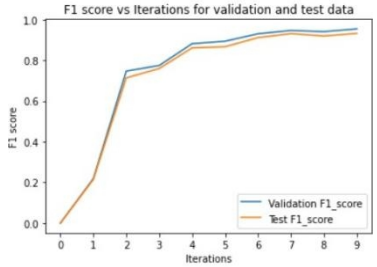
DATASET 1:

```
F1-score = 0.9539585935101762
Precision = 0.9533804592567782
Recall = 0.96609236308826
==============TEST DATA=======================
F1-score = 0.940385511513963
Precision = 0.9390140471773125
Recall = 0.9603741496598639
=====================================
=====================================
Interation = 9
Losses = {'ner': 51486.79979133606}
==============VALID DATA=======================
F1-score = 0.9555204006565121
Precision = 0.9563075699212863
Recall = 0.9650742477530284
==============TEST DATA=======================
F1-score = 0.9337134987195012
Precision = 0.9303560385193038
Recall = 0.9575757575757575
=====================================
```

DATASET 2:

```
Precision = 0.9412119224399413
Recall = 0.9507606207782058
==============TEST DATA=======================
F1-score = 0.91916031813991
Precision = 0.9182348263980917
Recall = 0.9410018552875695
=====================================
=====================================
Interation = 9
Losses = {'ner': 51476.82190322876}
==============VALID DATA=======================
F1-score = 0.9538320383805178
Precision = 0.9560565883250526
Recall = 0.9614532741584324
==============TEST DATA=======================
F1-score = 0.9317041235408583
Precision = 0.9318248078452159
Recall = 0.9487631416202844
=====================================
```





TC 3:

ITERATIONS:20

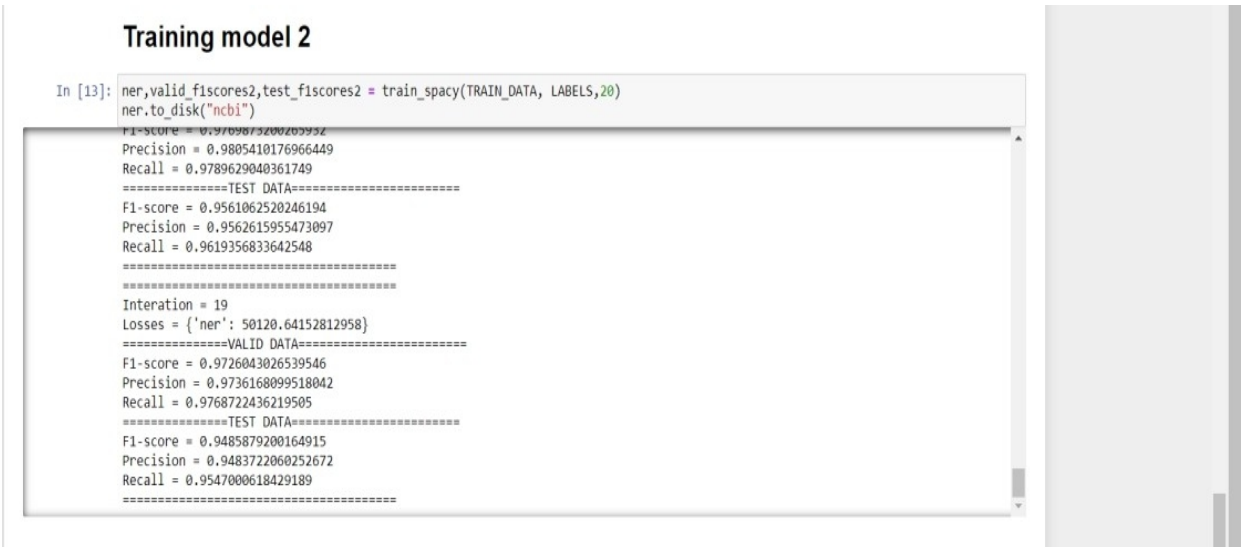
DROPOUT : 50 %

DATASET 1 :

Training model 1

```
In [7]: ner,valid_f1scores,test_f1scores = train_spacy(TRAIN_DATA, LABELS,20)
        ner.to_disk("spacy_example")
        F1-score = 0.9162449803449371
        Precision = 0.9205639573466873
        Recall = 0.934688140189533
        ==============TEST DATA=======================
        F1-score = 0.9069525712382857
        Precision = 0.9136766910077958
        Recall = 0.9227170671224155
        ========================================
        ========================================
        Interation = 19
        Losses = {'ner': 44403.25470161438}
        ==============VALID DATA=======================
        F1-score = 0.9191648697883337
        Precision = 0.9299747183591195
        Recall = 0.9299526234623727
        ==============TEST DATA=======================
        F1-score = 0.9079336802067274
        Precision = 0.9184043866837673
        Recall = 0.9197503714129328
        ========================================
```

DATASET 2 :



TC 4:

ITERATIONS : 30

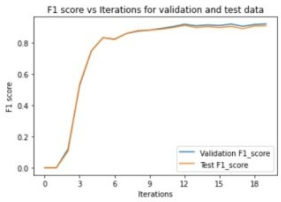
DROPOUT : 50 %

```
Precision = 0.982036894654128
Recall = 0.9842474739016357
==========TEST DATA==========
F1-score = 0.9545277851400301
Precision = 0.9571273964131106
Recall = 0.95717377860235
==========
==========
Interation = 29
Losses = {'ner': 49391.069215774536}
==========VALID DATA==========
F1-score = 0.9815647622152346
Precision = 0.9839011133779598
Recall = 0.983169346284821
==========TEST DATA==========
F1-score = 0.9527372853903467
Precision = 0.9568800247371678
Recall = 0.9539270253555967
==========
```

F1 graph for Dataset 1:

f1 graph for 1st dataset

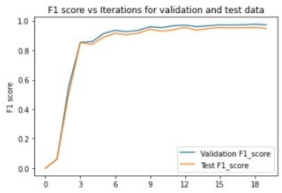
```
In [8]: x=range(0,20)

        ax = plt.figure().gca()
        #...
        ax.xaxis.set_major_locator(MaxNLocator(integer=True))
        ax.plot(valid_fiscores,label="Validation F1_score")
        ax.plot(test_fiscores,label="Test F1_score")
        ax.set_xlabel('Iterations')
        ax.set_ylabel('F1 score')
        ax.legend()
        ax.set_title('F1 score vs Iterations for validation and test data')
        # x axis ('Iteration')
        # y axis('F1 score')

Out[8]: Text(0.5, 1.0, 'F1 score vs Iterations for validation and test data')
```



F1 Graph for Dataset 2:

f1 graph for second dataset

```
In [14]: x=range(0,20)

         ax = plt.figure().gca()
         #...
         ax.xaxis.set_major_locator(MaxNLocator(integer=True))
         ax.plot(valid_fiscores2,label="Validation F1_score")
         ax.plot(test_fiscores2,label="Test F1_score")
         ax.set_xlabel('Iterations')
         ax.set_ylabel('F1 score')
         ax.legend()
         ax.set_title('F1 score vs Iterations for validation and test data')
         # x axis ('Iteration')
         # y axis('F1 score')

Out[14]: Text(0.5, 1.0, 'F1 score vs Iterations for validation and test data')
```



10.2 INFERENCE TABLE:

INFERENCE TABLE FOR DATASET 1 :

	Iterations	Dropout	F1Score (VD)	Precision (VD)	Recall (VD)	F1Score (TD)	Precision (TD)	Recall (TD)
Dataset 1	**10**	**40%**	**0.966**	**0.967**	**0.971**	**0.949**	**0.946**	**0.961**
	10	**50%**	**0.955**	**0.956**	**0.965**	**0.933**	**0.930**	**0.957**
	20	**50%**	**0.919**	**0.929**	**0.929**	**0.907**	**0.918**	**0.919**
	30	**50%**	**0.981**	**0.983**	**0.983**	**0.952**	**0.956**	**0.953**

INFERENCE TABLE FOR DATASET 2:

	Iterations	Dropout	F1Score (VD)	Precision (VD)	Recall (VD)	F1Score (TD)	Precision (TD)	Recall (TD)
Dataset 2	**10**	**40%**	**0.969**	**0.970**	**0.975**	**0.944**	**0.942**	**0.957**
	10	**50%**	**0.953**	**0.956**	**0.961**	**0.931**	**0.931**	**0.948**
	20	**50%**	**0.972**	**0.973**	**0.976**	**0.948**	**0.948**	**0.954**

11. CONCLUSION

The project aims at creating a model that efficiently classifies disease entities (words) from a document with higher accuracies. We also have our visual representations of results with different possible conditions. NLP predictions involves different types of documents and higher degrees of cross validations. We tried out possible interpretations and proved better results in finding diseases. We achieved a f1 score of 95% and 94% percent with our standard medical NLP corpus of B5CDR and NCBI disease datasets. This proposed system can be very useful among hospitals and healthcare workers and particularly incorporated on medical bots (chatbots and mechanical bots) for better automated future.

12. LIST OF REFERENCES

- Beltagy, I., Lo, K., Cohan, A.: SciBERT: A Pretrained Language Model for Scientific Text. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, pp. 3606–3611 (2019)
- Cho, H., Lee, H.: Biomedical Named Entity Recognition Using Deep Neural Networks with Contextual Information. BMC bioinformatics 20(1), 735 (2019)
- Biomedical named entity recognition at Scale – by Veysel Cocaman @ John snow Labs ,DE ,USA (12 Nov 2020)
- https://spacy.io/models
- Disease Detection in Medical Prescriptions Using Data Mining Tools -by University of Tehran
- A Survey on Deep Learning for Named Entity Recognition Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li (18th March 2020) , Cornell University
- A review of the application of natural language processing in clinical medicine (IEEE 2018 13th conference on NLP)
- A novel system for the automatic extraction of a patient problem summary (IEEE 2017, Symposium on Computer science)

13.APPENDIX A:

The undersigned acknowledge they have completed implementing the project "Medical information extraction and disease detection using NLP " and agree with the approach it presents.

Signature: _____ Date: 19/05/2021

Name: Balaji S

Signature: _____ Date: 19/05/2021

Name: Dhananjeyan AK

14. APPENDIX B: REFERENCES

The following table summarizes the research papers referenced in this document.

Document Name and Version	Description	Location
A novel system for the automatic extraction of a patient problem summary (IEEE 2017, Symposium on Computer science)	The implemented system relies on an NLP pipeline, for the extraction of relevant medical entities contained in narrative health records, and on several queries, necessary for the scanning of structured documents.	https://ieeexplore.ieee.org/document/8024526
A review of the application of natural language processing in clinical medicine	This paper first introduces the principle of natural language processing and select the domestic representative extraction method and the application system and classified according to different application technology, the paper introduced all kinds of technology in the application in the field of clinical medicine.	https://ieeexplore.ieee.org/document/8398172
Biomedical named entity recognition at Scale – by Veysel Cocaman @ John snow Labs ,DE ,USA (12 Nov 2020)	Reimplementing a Bi-LSTM-CNN-Char deep learning architecture on top of Apache Spark, we present a single trainable NER model that obtains new state-of-the-art results on seven public biomedical benchmarks without using heavy contextual embeddings like BERT	https://arxiv.org/pdf/2011.06315.pdf

15.APPENDIX C: KEY TERMS

The following table provides definitions for terms relevant to this document.

Term	Definition
Big Data	**Big data** is a term for **data** sets that are **large** or complex that traditional **data** processing applications are inadequate
CNN	CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean **fully connected networks**, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks make them **prone to overfitting** data.
NER	Named-entity recognition (NER) (also known as (named) entity identification, entity chunking, and entity extraction) is a subtask of **information extraction** that seeks to locate and classify **named entities** mentioned in **unstructured text** into pre-defined categories such as person names, organizations, locations, **medical codes**, time expressions, quantities, monetary values, percentages, etc.
F1 score	The F1 Score is the **2*((precision*recall)/(precision+recall))**. It is also called the F Score or the **F Measure**. Put another way, the F1 score conveys the **balance** between the **precision and the recall**.
Dropout	The term “dropout” refers to dropping out units (both hidden and visible) in a neural network. At each training stage, individual nodes are either dropped out of the net with probability *1-p* or kept with probability *p*, so that a reduced network is left; incoming and outgoing edges to a dropped-out node are also removed.