Natural Language Processing

Task 8: MeasEval - Counts and Measurements

Term Project - Presentation

Presented By:
Somoshree Datta (20CS60R05)
Pratibha Singh (20CS60R12)
Aditya Anand (20CS60R24)
Yogesh Porwal (20CS60R52)
Tushika Agrawal (20CS60R55)

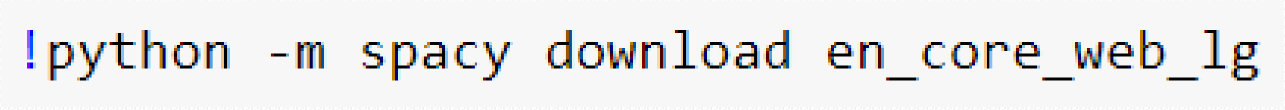
Contents:

- Preprocessing Stage 1:
  - Finding the suitable English corpus for Spacy
  - Connecting colab to google drive and importing data from drive to colab
  - Loading all the text data and preprocessing them as required for training and evaluation
- Preprocessing Stage 2:
  - Loading all the tsv data and preprocessing them as required for training and evaluation
  - Merging data from 4 and 5 as suitable for training and evaluation
- Model Training:
  - Adjusting spacy and training the model
  - Saving and loading the trained model
- Evaluation:
  - Confusion matrix creation and Scores calculation
- Visualization:
  - Plots and its analysis.
- Results:
  - Tsv generation and submission
- Future Scope

Preprocessing Stage 1

- Finding the suitable English corpus for spacy
  - En_core_web_sm – The small corpus (13 MB)
  - En_core_web_md – The medium corpus (44 MB)
  - **En_core_web_lg – The large corpus (742 MB)**

  - <u>Download and Install:</u>

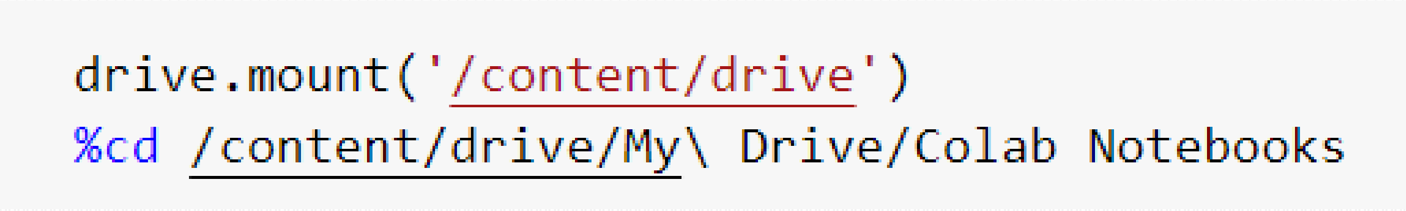


```
!python -m spacy download en_core_web_lg
```

Tushika Agrawal

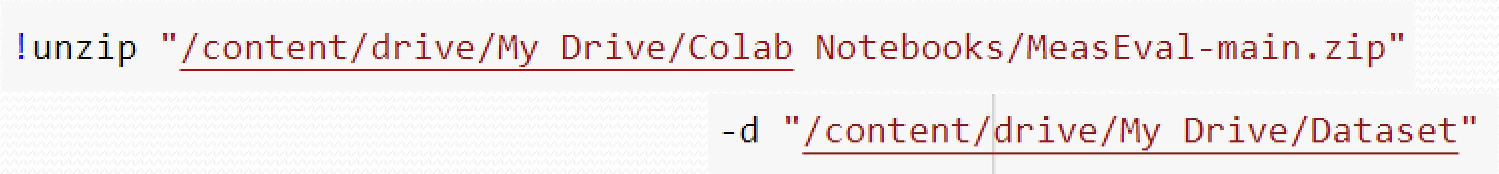
Preprocessing Stage 1

- Connecting Colab to Drive:

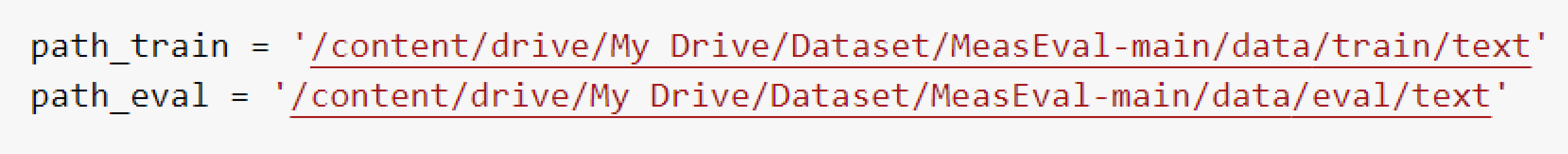


```
drive.mount('/content/drive')
%cd /content/drive/My\ Drive/Colab Notebooks
```

- Importing data from drive to colab:



```
!unzip "/content/drive/My Drive/Colab Notebooks/MeasEval-main.zip"
                                           -d "/content/drive/My Drive/Dataset"
```

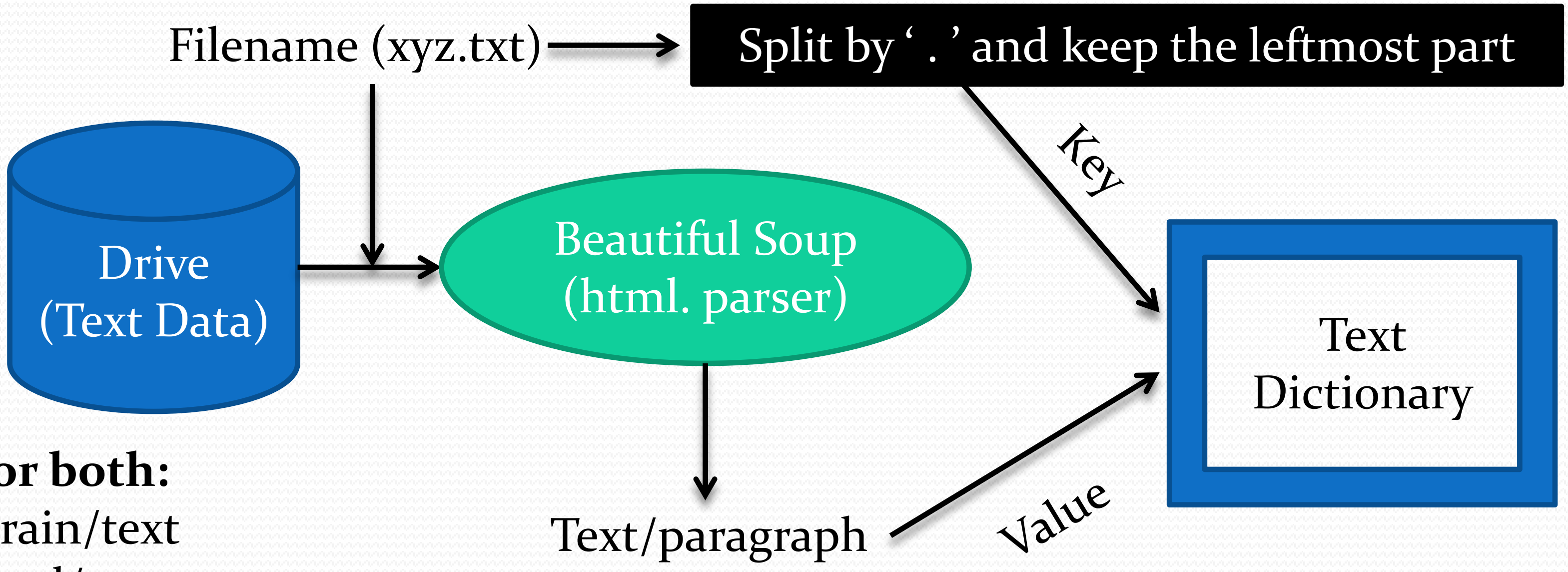


```
path_train = '/content/drive/My Drive/Dataset/MeasEval-main/data/train/text'
path_eval = '/content/drive/My Drive/Dataset/MeasEval-main/data/eval/text'
```

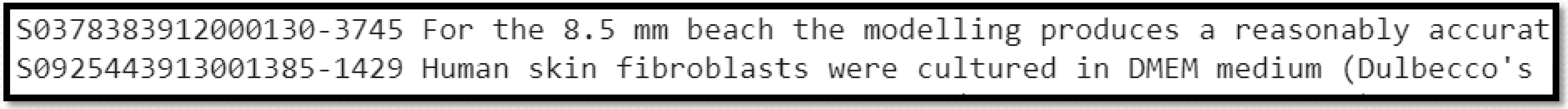
Tushika Agrawal

Preprocessing Stage 1

- Preprocessing text data for training and evaluation:



For both:
/train/text
/eval/text



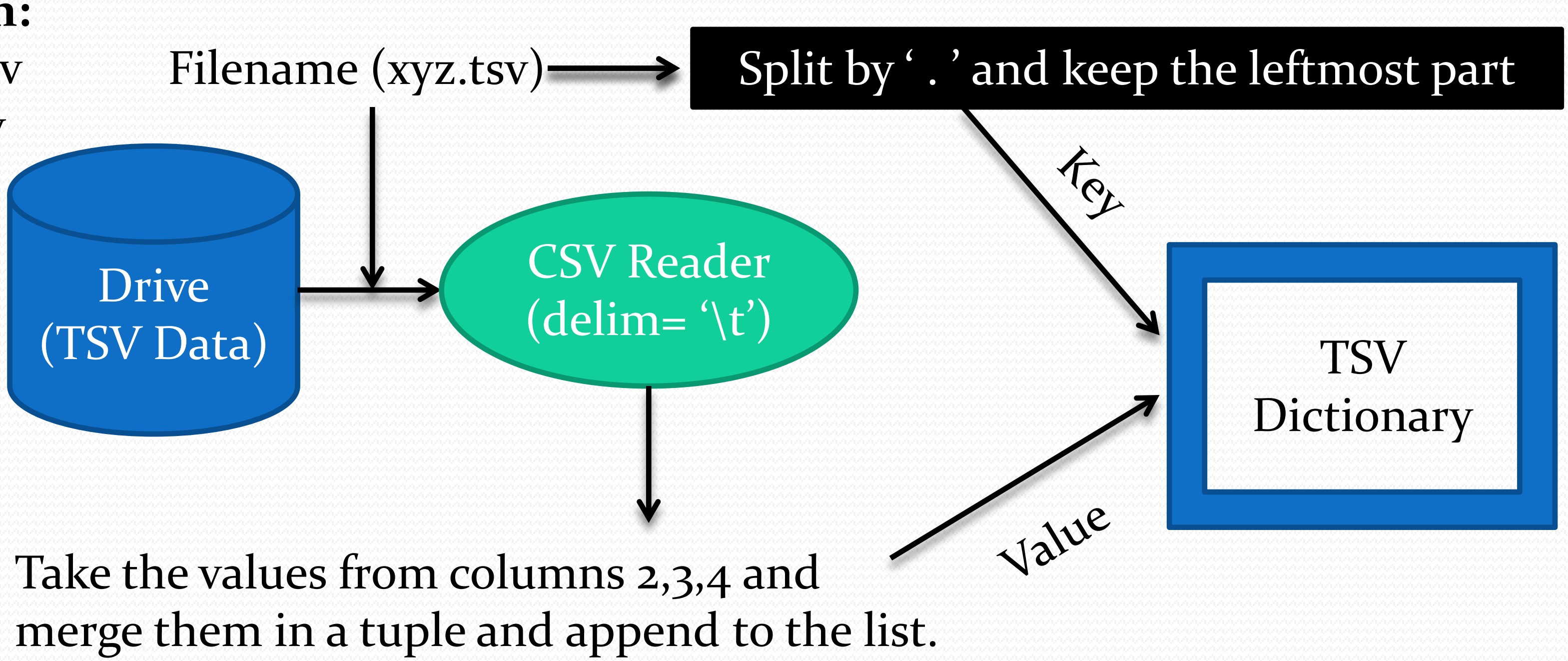
```
S0378383912000130-3745 For the 8.5 mm beach the modelling produces a reasonably accurat
S0925443913001385-1429 Human skin fibroblasts were cultured in DMEM medium (Dulbecco's
```

Tushika Agrawal

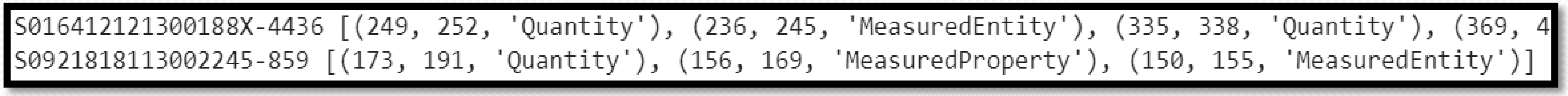
Preprocessing Stage 2

- Preprocessing tsv data for training and evaluation:

For both:
/train/tsv
/eval/tsv



Take the values from columns 2,3,4 and merge them in a tuple and append to the list.

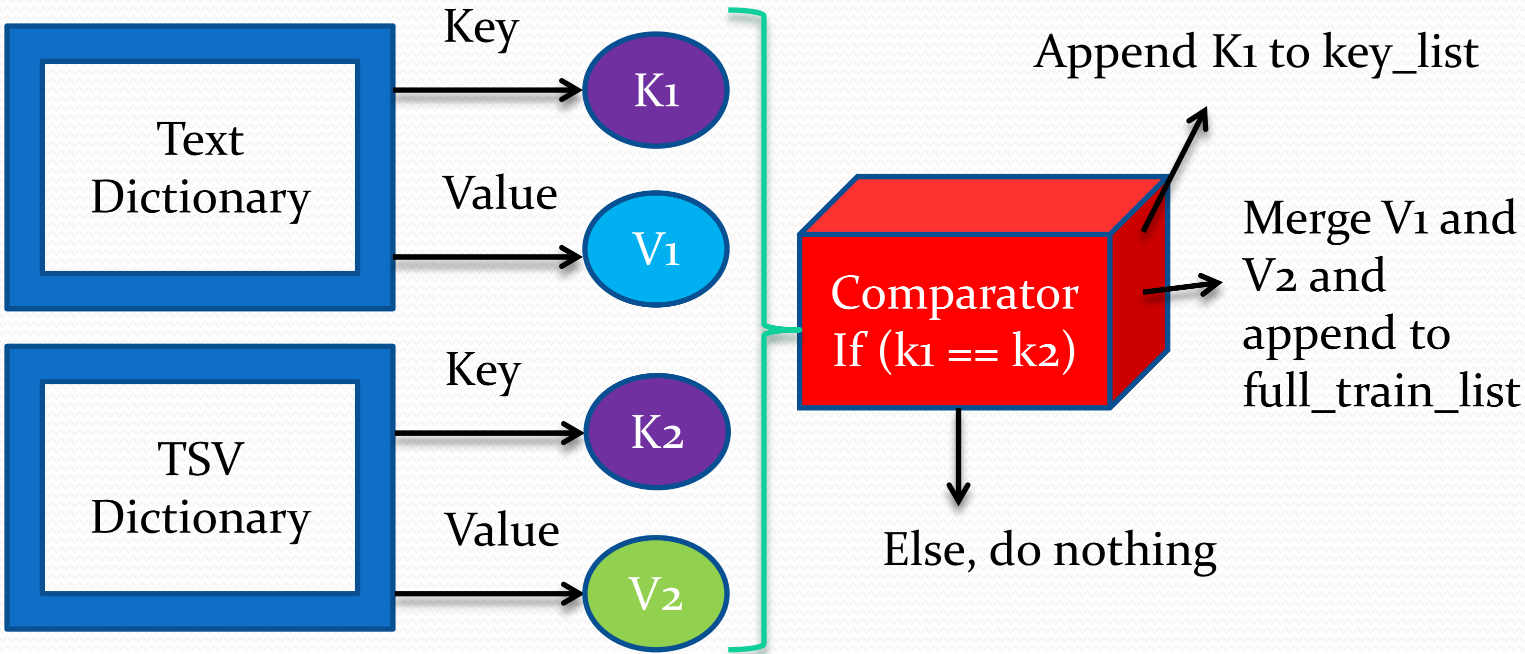


```
S016412121300188X-4436 [(249, 252, 'Quantity'), (236, 245, 'MeasuredEntity'), (335, 338, 'Quantity'), (369, 4
S0921818113002245-859 [(173, 191, 'Quantity'), (156, 169, 'MeasuredProperty'), (150, 155, 'MeasuredEntity')]
```

Pratibha Singh

Preprocessing Stage 2

- Merging text and tsv data as required by the Model:



Text Dictionary

Key

K1

Value

V1

TSV Dictionary

Key

K2

Value

V2

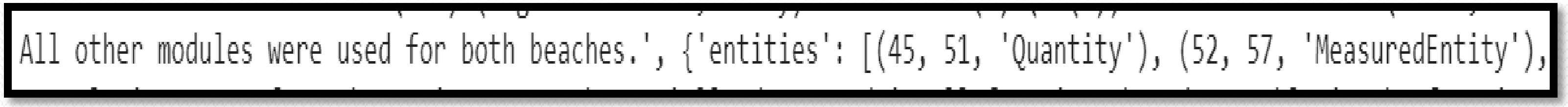
Comparator

If (k1 == k2)

Append K1 to key_list

Merge V1 and V2 and append to full_train_list

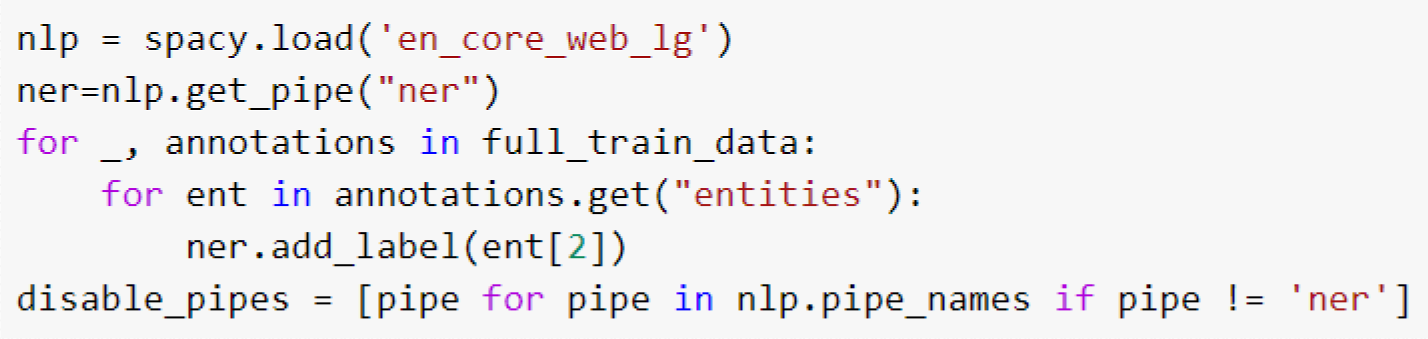
Else, do nothing



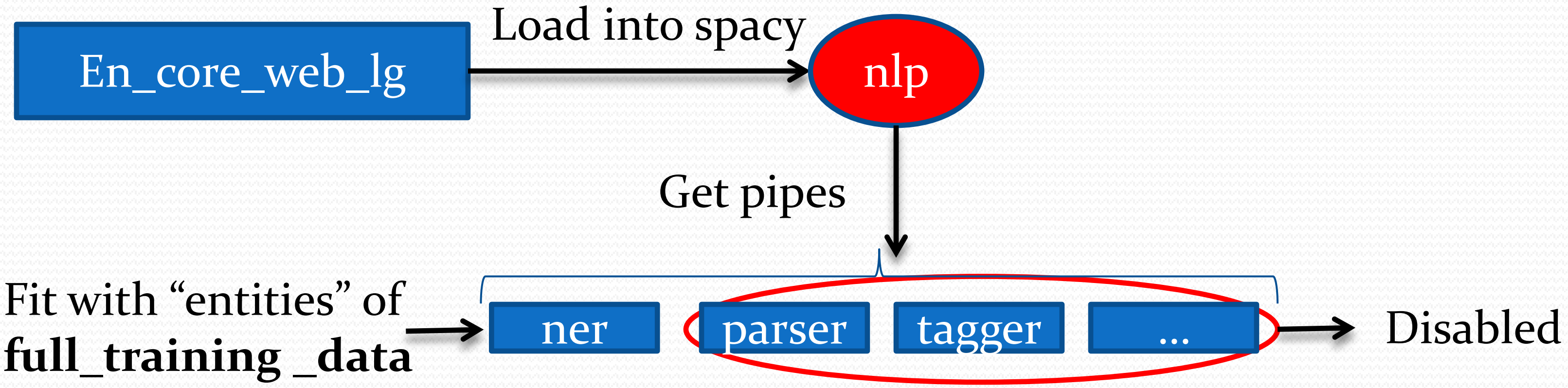
```
All other modules were used for both beaches.', {'entities': [(45, 51, 'Quantity'), (52, 57, 'MeasuredEntity'),
```

Model and Training

- Model Preparation and Adjustments:



```
nlp = spacy.load('en_core_web_lg')
ner=nlp.get_pipe("ner")
for _, annotations in full_train_data:
    for ent in annotations.get("entities"):
        ner.add_label(ent[2])
disable_pipes = [pipe for pipe in nlp.pipe_names if pipe != 'ner']
```



En_core_web_lg

Load into spacy

nlp

Get pipes

Fit with “entities” of **full_training _data**

ner

parser

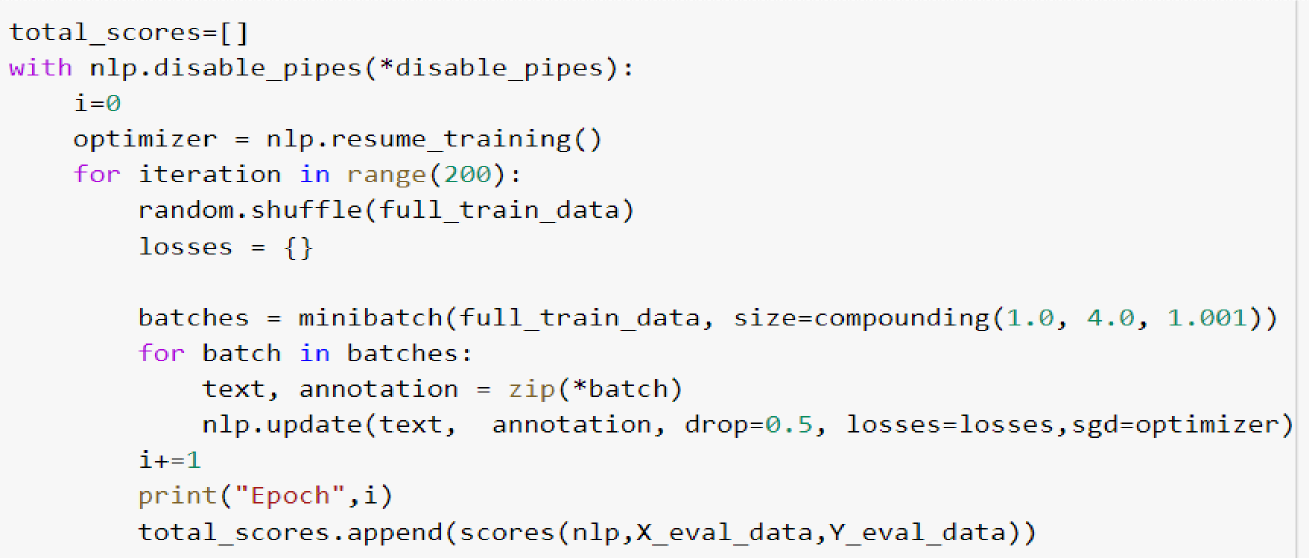
tagger

...

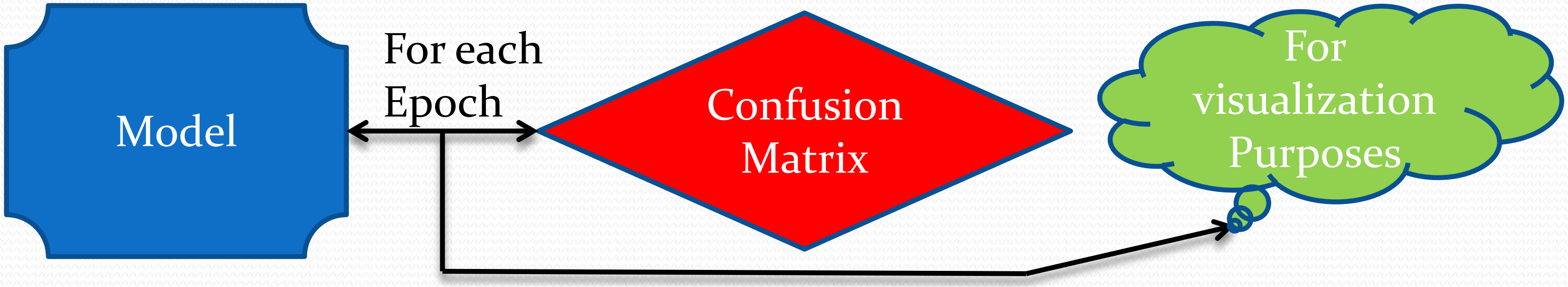
Disabled

Model and Training

- Model Training:



```
total_scores=[]
with nlp.disable_pipes(*disable_pipes):
    i=0
    optimizer = nlp.resume_training()
    for iteration in range(200):
        random.shuffle(full_train_data)
        losses = {}

        batches = minibatch(full_train_data, size=compounding(1.0, 4.0, 1.001))
        for batch in batches:
            text, annotation = zip(*batch)
            nlp.update(text,  annotation, drop=0.5, losses=losses,sgd=optimizer)
        i+=1
        print("Epoch",i)
        total_scores.append(scores(nlp,X_eval_data,Y_eval_data))
```

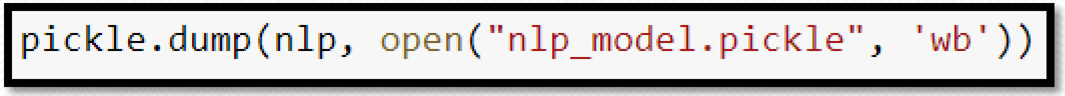


Somoshree Datta

Model Saving and Loading

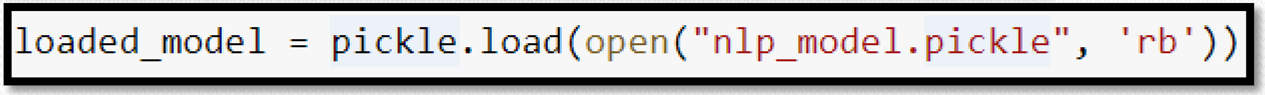
- Training the model is time consuming (takes nearly 2 hours).
- Beneficial for multiple runs.
- It is an optional step.

  - Saving:



```
pickle.dump(nlp, open("nlp_model.pickle", 'wb'))
```

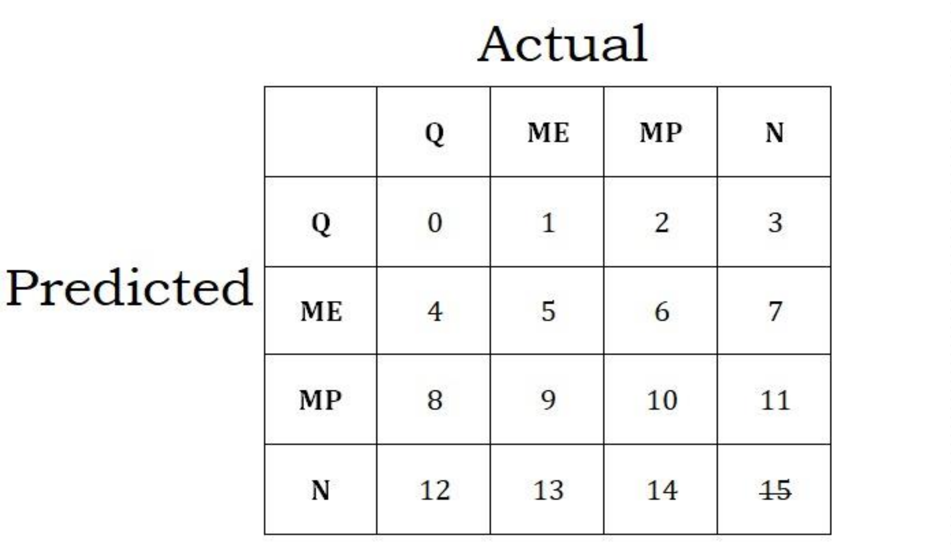
  - Loading:



```
loaded_model = pickle.load(open("nlp_model.pickle", 'rb'))
```

Evaluation Method

- The Confusion Matrix:



		Actual			
		Q	ME	MP	N
Predicted	Q	0	1	2	3
	ME	4	5	6	7
	MP	8	9	10	11
	N	12	13	14	~~15~~

Q: Quantity
ME: Measured Entity
MP: Measured Property
N: None of these

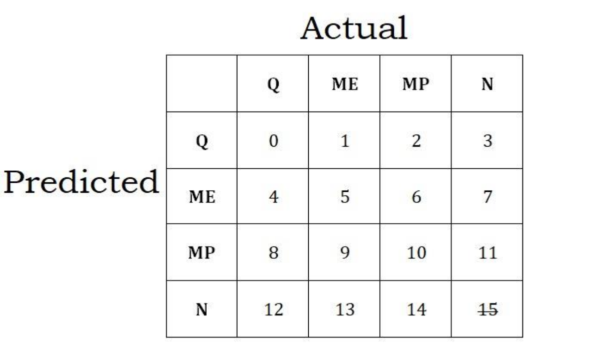
Represented in the form of 1D-array (size = 16).

- Calculated on the **/eval/text** data as **test data** and **/eval/tsv** data as the **corresponding labels**.

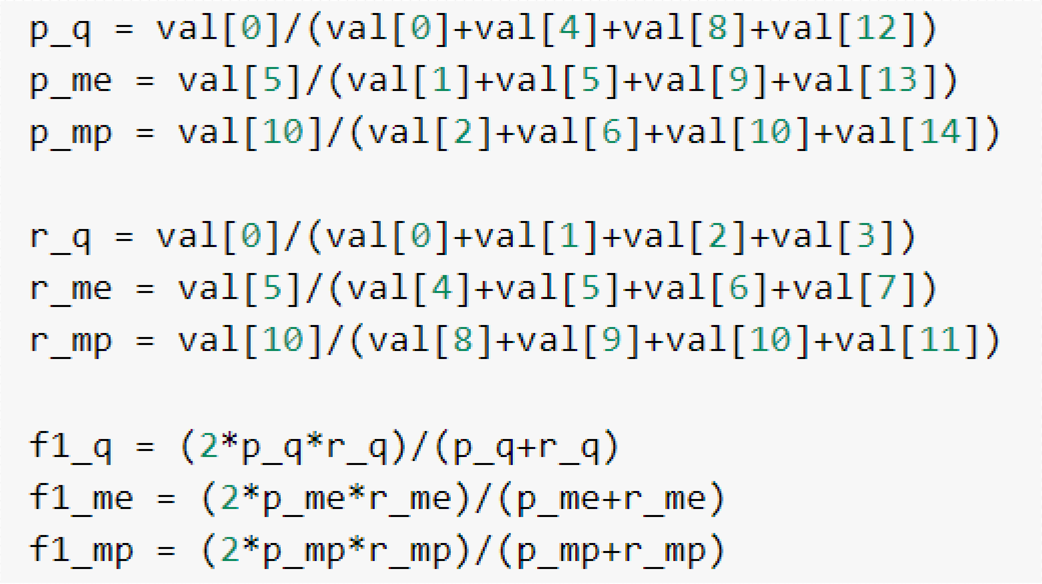
Aditya Anand

Evaluation Method

- Precision, Recall and F1-Score calculation:



		Actual			
		Q	ME	MP	N
Predicted	Q	0	1	2	3
	ME	4	5	6	7
	MP	8	9	10	11
	N	12	13	14	~~15~~

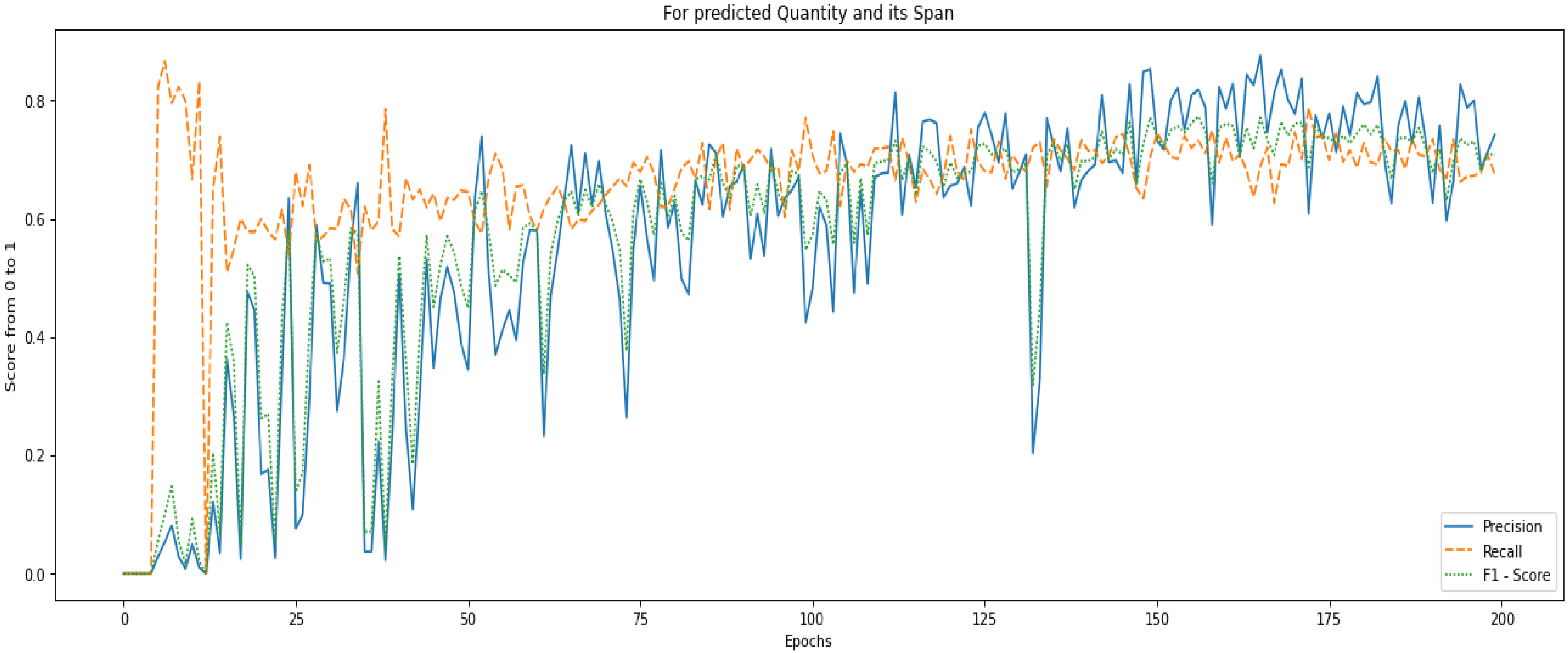


```
p_q = val[0]/(val[0]+val[4]+val[8]+val[12])
p_me = val[5]/(val[1]+val[5]+val[9]+val[13])
p_mp = val[10]/(val[2]+val[6]+val[10]+val[14])

r_q = val[0]/(val[0]+val[1]+val[2]+val[3])
r_me = val[5]/(val[4]+val[5]+val[6]+val[7])
r_mp = val[10]/(val[8]+val[9]+val[10]+val[11])

f1_q = (2*p_q*r_q)/(p_q+r_q)
f1_me = (2*p_me*r_me)/(p_me+r_me)
f1_mp = (2*p_mp*r_mp)/(p_mp+r_mp)
```

q = quantity
me = measured entity
mp = measured property

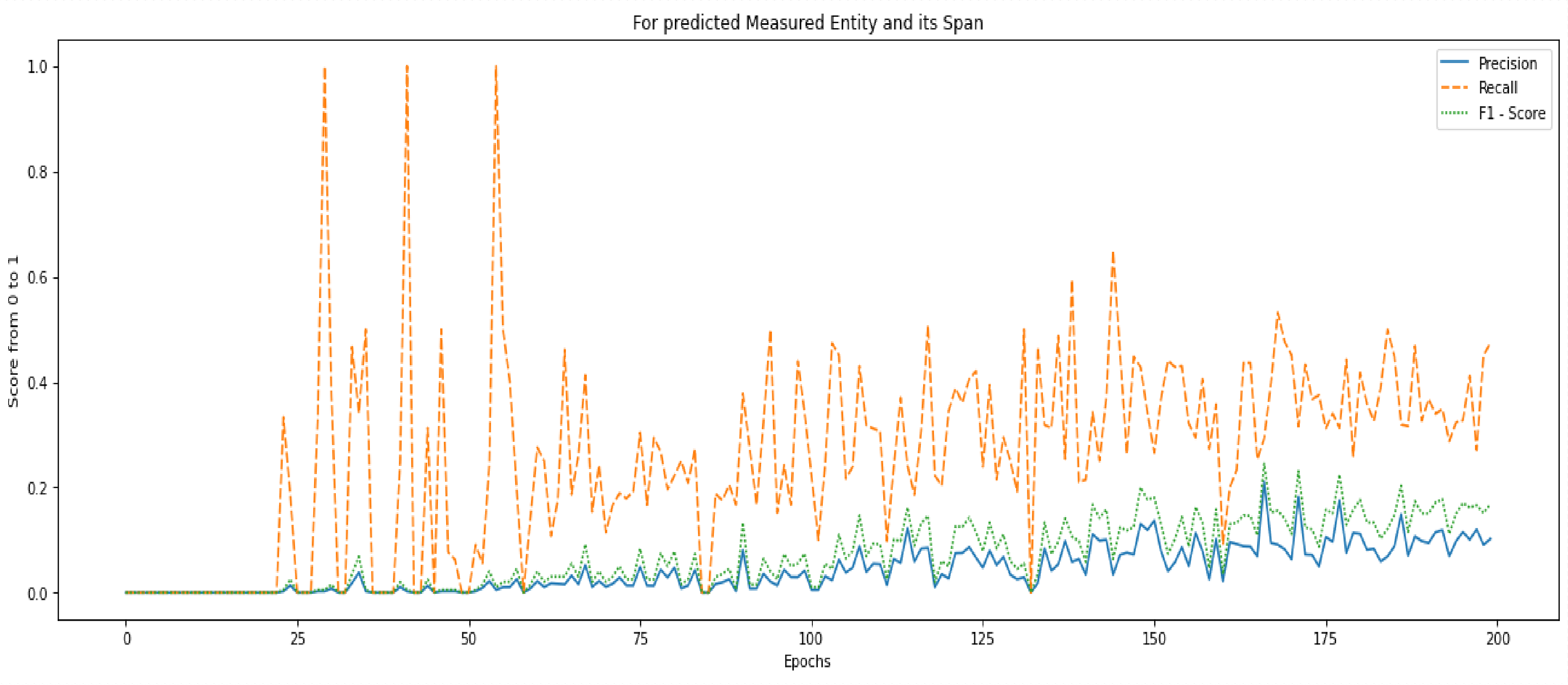
p = precision = TP / (TP + FP) ; r = recall = TP / (TP + FN)
f1 = f1-score = (2*p*r) / (p+r)

Plots and Analysis



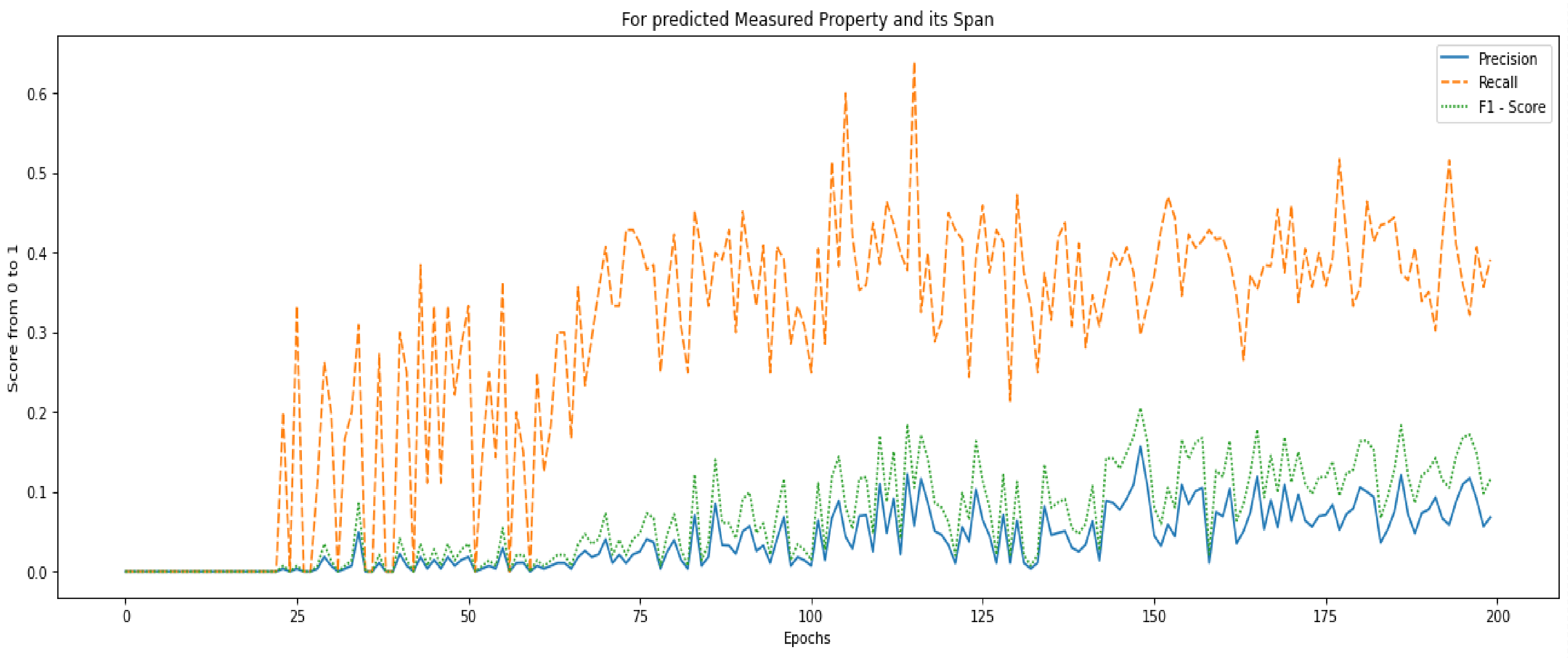
Yogesh Porwal

Plots and Analysis



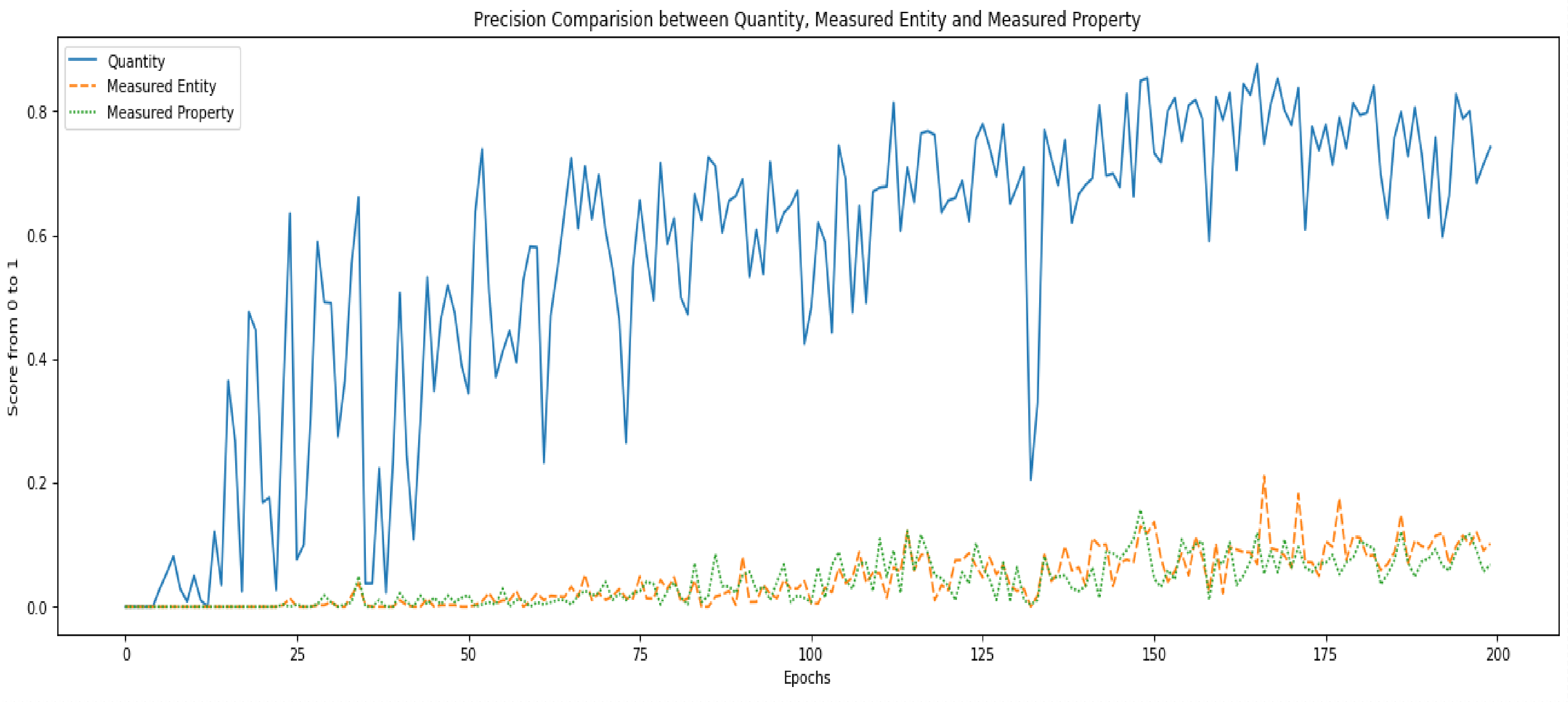
Yogesh Porwal

Plots and Analysis



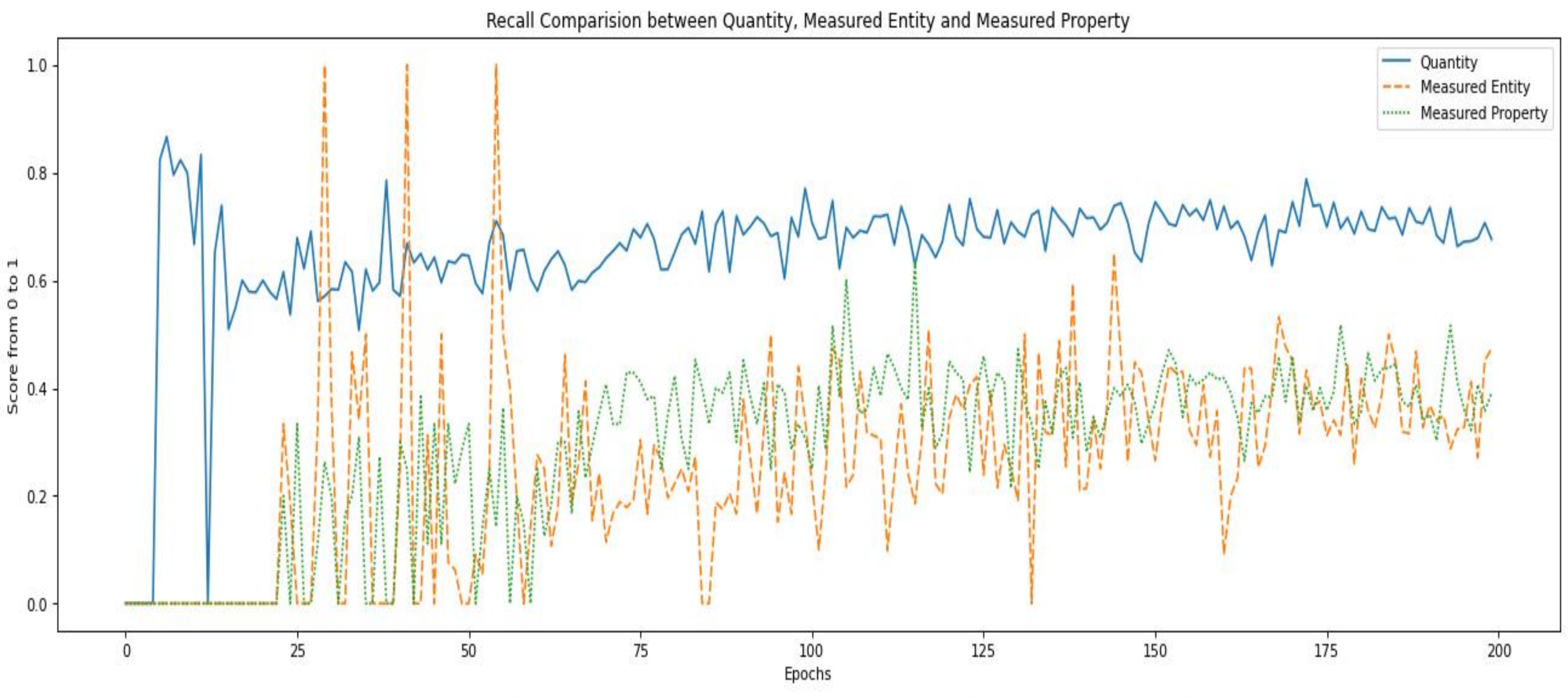
Yogesh Porwal

Plots and Analysis



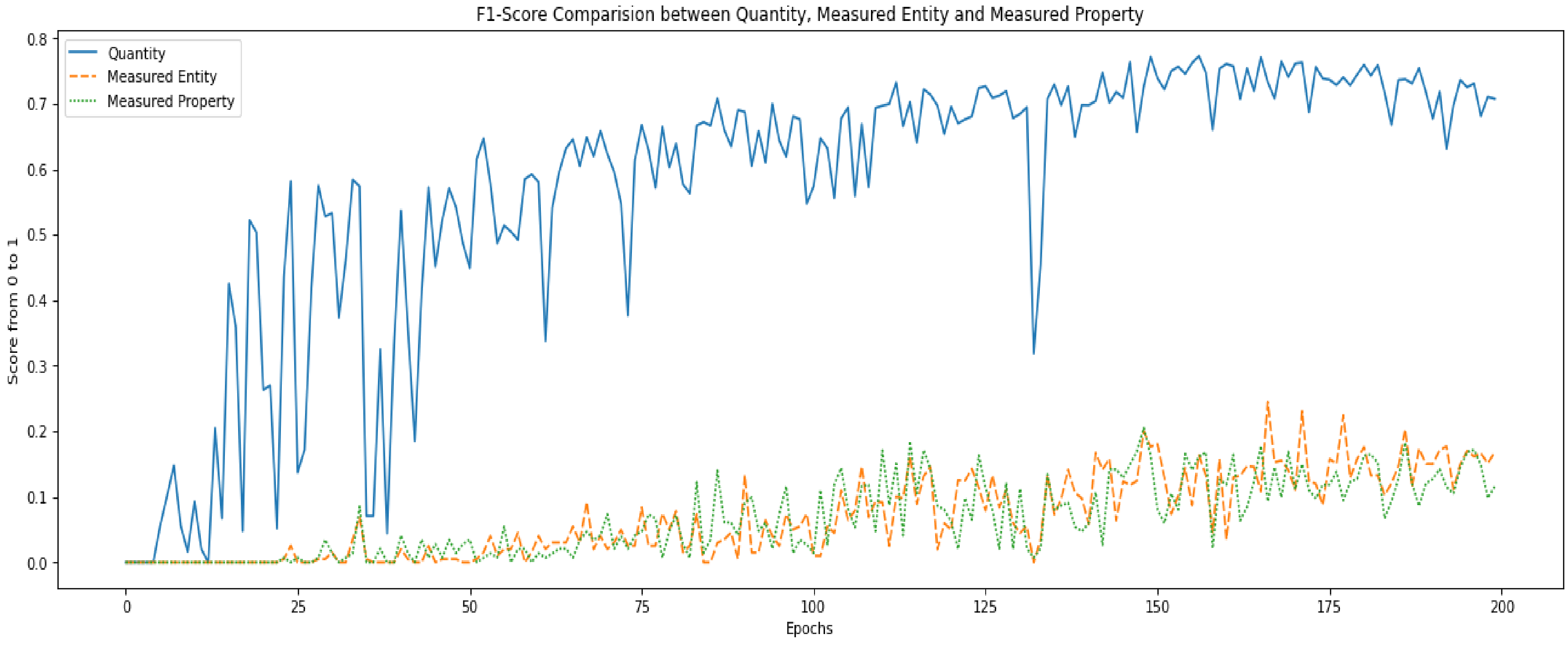
Pratibha Singh

Plots and Analysis



Pratibha Singh

Plots and Analysis



Pratibha Singh

TSV generation

- For each /eval/text data file, do:
  - Create a xyz.tsv file, where **xyz is the filename removing ‘.txt’.**
  - Now, to that xyz.tsv write the following header:

```
docId \t annotSet \t annotType \t startOffset \t endOffset \t annotId \t text \t other \n
```

  - For each predicted entity, append to file the following info:

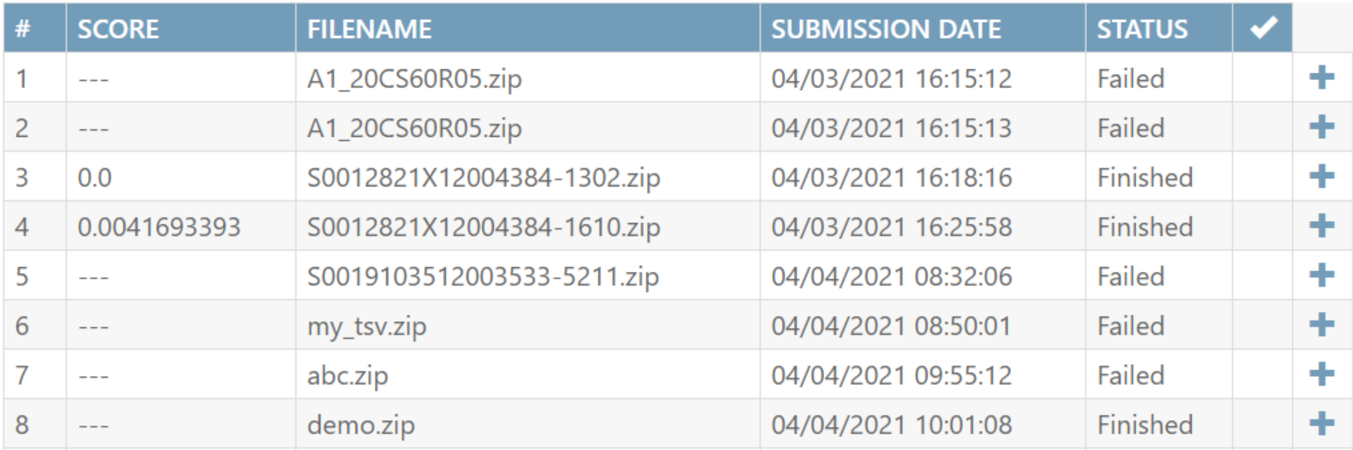
```
xyz \t {dummy} \t V \t S \t E \t {dummy} \t xyz_data.substr(S,E) \t {dummy} \n
```

    - Predicted entity → (S, E, V)
      - S denotes Start of the span – [0-9]+
      - E denotes End of the span – [0-9]+
      - V denotes the type, i.e., Quantity, Measured Entity or Measured Property – [a-zA-Z]+
    - xyz_data stores the text data obtained from xyz.txt file.

Yogesh Porwal

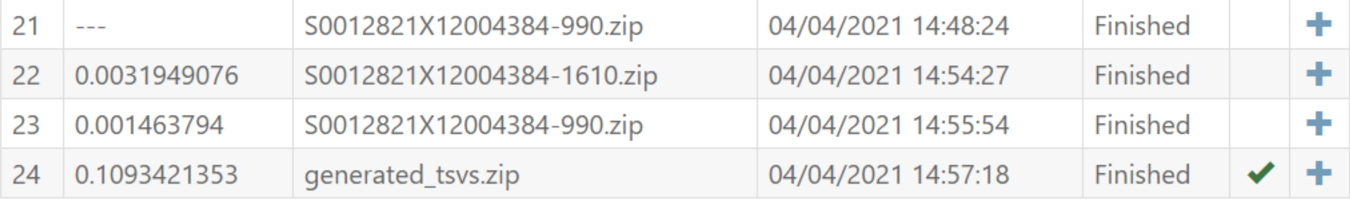
Submission on CodaLab

- We were facing a lot of issues while submission, so our entries increased by a lot.



#	SCORE	FILENAME	SUBMISSION DATE	STATUS	✓	
1	---	A1_20CS60R05.zip	04/03/2021 16:15:12	Failed		+
2	---	A1_20CS60R05.zip	04/03/2021 16:15:13	Failed		+
3	0.0	S0012821X12004384-1302.zip	04/03/2021 16:18:16	Finished		+
4	0.0041693393	S0012821X12004384-1610.zip	04/03/2021 16:25:58	Finished		+
5	---	S0019103512003533-5211.zip	04/04/2021 08:32:06	Failed		+
6	---	my_tsv.zip	04/04/2021 08:50:01	Failed		+
7	---	abc.zip	04/04/2021 09:55:12	Failed		+
8	---	demo.zip	04/04/2021 10:01:08	Finished		+

• • •



21	---	S0012821X12004384-990.zip	04/04/2021 14:48:24	Finished		+
22	0.0031949076	S0012821X12004384-1610.zip	04/04/2021 14:54:27	Finished		+
23	0.001463794	S0012821X12004384-990.zip	04/04/2021 14:55:54	Finished		+
24	0.1093421353	generated_tsvs.zip	04/04/2021 14:57:18	Finished	✓	+

Yogesh Porwal

Future Scope

- We have just created an initial version of NER.
- Future improvements can be:
  - Using a BERT:
    - General Purpose Language Model pretrained on large datasets.
    - Can be used to achieve state-of-the-art perfomance.
  - Using Dependency Parsing:
    - Using dependency parse tree, to examine nature of relationship between various components of a sentence.
    - Can be used to determine Measured Entities and Properties corresponding to Particular Quantities[subtask-5, in our case].

Yogesh Porwal

Thank you