Maximum Entropy Markov Model

FEATURE ENGINEERING

We used feature engineering to increase the efficiency of the MeMM algorithm for chunking.The features used are:-

Features used for Maximum Entropy Markov Model :

- Prefix of the word : Created list of 50 prefixes
- Suffix of the word : Created list of 99 suffixes
- Capitalization of word
- Is word a start of sentence
- POS Tag of current word
- POS Tag of 2 previous words and 2 next words
- Current word
- 2 previous words and 2 next words
- Chunk Labels of previous two words

Implemented the MeMM model using nltk.classify.MaxentClassifier

Metrics of MeMM model with using POS Tags:

Overall Precision, Recall, and F-score using average = “macro”:
0.9321089026742283, 0.9011599105544924, 0.91516702366346584

Precision, Recall and F-Score using average=”macro” for B tags:
(0.9040158259149357, 0.9597194220430108, 0.9310351852983721

Precision, Recall and F-Score using average=”macro” for I tags:
0.9183772328186497, 0.8762493500491074, 0.8968188268684957

Metrics of MeMM model without using POS Tags and using other tags mentioned above :

Overall Precision, Recall, and F-score using average = “macro”:
0.892921249152352, 0.774456248923309, 0.8120423184103159

Precision, Recall and F-Score using average=”macro” for B tags:
0.8230711233978386, 0.962911626344086, 0.8875166953794932

Precision, Recall and F-Score using average=”macro” for I tags:
0.8639336016096579, 0.7938066901611878, 0.827386866588384

CONDITIONAL RANDOM FIELD

FEATURE ENGINEERING

We used feature engineering to increase the efficiency of the CRF algorithm for chunking.The features used are:-

Features used for Maximum Entropy Markov Model :

- Prefix of the word : Created list of 50 prefixes
- Suffix of the word : Created list of 99 suffixes
- Capitalization of word
- Is word a start of sentence
- POS Tag of current word
- POS Tag of 2 previous words and 2 next words
- Current word
- 2 previous words and 2 next words
- Chunk Labels of previous two words

Metrics of CRF model with using POS Tags:

Overall Precision, Recall, and F-score using average = “macro”:
0.9599530934038502, 0.9602647341673145, 0.9601043609488603

Precision, Recall and F-Score using average=”macro” for B tags:
0.9695013920526449, 0.9653477822580645, 0.967420128804142

Precision, Recall and F-Score using average=”macro” for I tags:
0.9493569131832797, 0.9552256051764978, 0.9522822174226062

Metrics of CRF model without using POS Tags and using other tags mentioned above :

Overall Precision, Recall, and F-score using average = “macro”:
0.9473510100569134, 0.9492015186257924, 0.9482446759036803

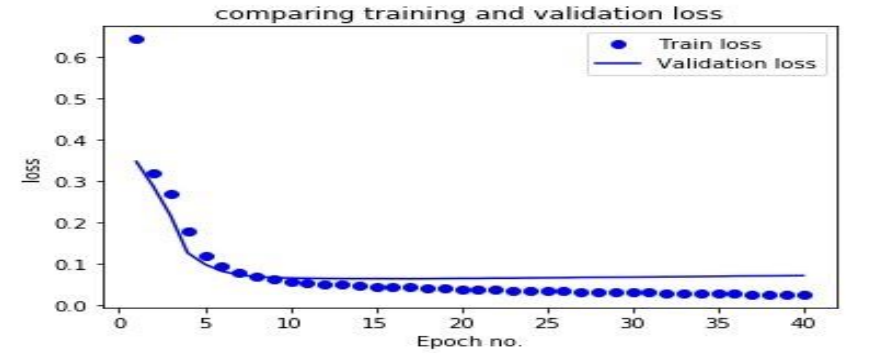
Precision, Recall and F-Score using average=”macro” for B tags:
0.9597754911131899, 0.9480846774193549, 0.9538942653087099

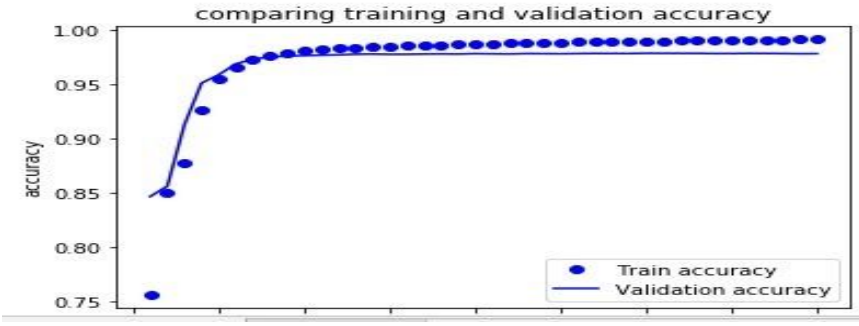
Precision, Recall and F-Score using average=”macro” for I tags:
0.9564443005181347, 0.9590842669264491, 0.95776246453182

Bi-LSTM

We did’nt used any features for Bi-LSTM. We just passed the words ank chunk labels to Bi-LSTM created in last assignment.

```
accuracy with Paddings :   0.9782905133302747

accuracy without paddings :   0.9283030021534434
```



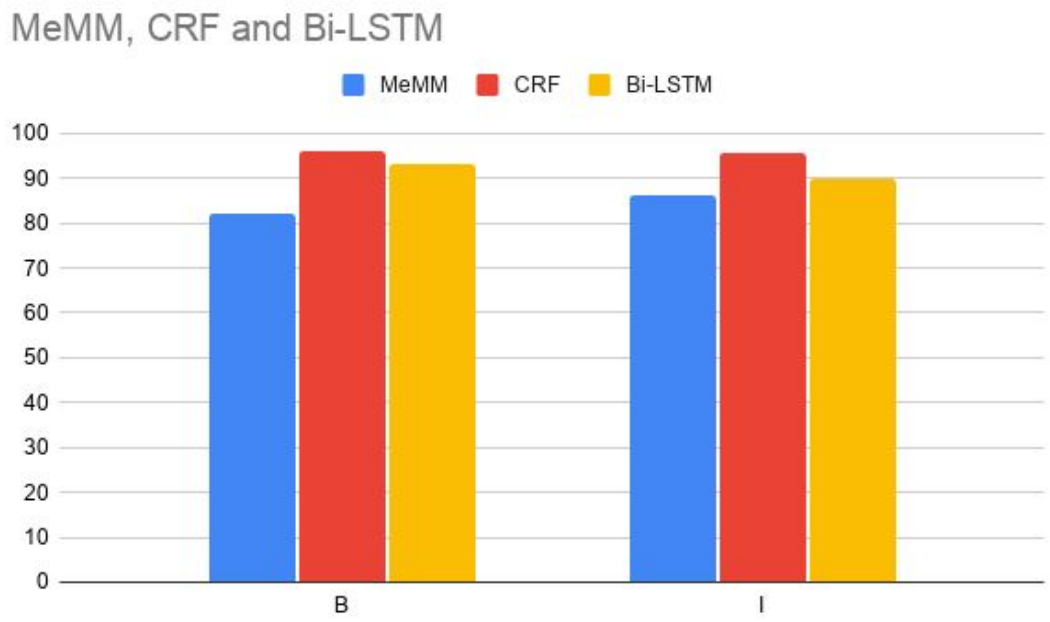


```
7148/7148 [==============================] - 23s
3ms/step - loss: 0.0158 - accuracy: 0.9952 -
masked_accuracy: 0.9842 - val_loss: 0.0923 -
val_accuracy: 0.9754 - val_masked_accuracy: 0.9196

history of accuracy during training :  [0.7998472,
0.873827, 0.9433605, 0.96994495, 0.9810024,
0.9847725, 0.98696065, 0.9882574, 0.98933893,
0.99034876, 0.9911487, 0.99177283, 0.9923414,
0.99284184, 0.9934265, 0.99364537, 0.9940704,
0.99460673, 0.9949027, 0.9952004]
Predicted  -PADDING-      B      I     O   Total
Actual
-PADDING-     109559      0      0     0  109559
B                  1  22192   1593    66   23852
I                  4   1475  15610   256   17345
O                  0    104    226  5850    6180
Total         109564  23771  17429  6172  156936

accuracy with Paddings :  0.9762642096141102

accuracy without paddings :  0.9214725998480114
```

Confusion Matrix

	B	I	O
B	22192	1593	66
I	1475	15610	256
O	104	226	5850

Accuracy Per Chunk Label without using POS TAG



Accuracy Per Label using POS TAG