

Intro to NLP

Assignment-2

Report

2020101075

For validation data, following are the scores (in case of epoch 6)

	precision	recall	f1-score	support
ADP	0.94	1.00	0.97	1415
PRON	0.88	0.99	0.93	414
PROPN	0.99	0.99	0.99	1551
CCONJ	1.00	0.99	1.00	107
AUX	0.95	1.00	0.97	266
DET	0.99	0.87	0.92	568
ADJ	0.97	0.93	0.95	227
INTJ	1.00	1.00	1.00	35
PART	0.00	0.00	0.00	73
PAD_POS	1.00	1.00	1.00	17422
UNK_TAG	0.00	0.00	0.00	2
NOUN	0.97	0.99	0.98	1143
NUM	0.99	0.90	0.94	131
VERB	0.98	0.98	0.98	653
ADV	1.00	0.80	0.89	59
accuracy			0.99	24066
macro avg	0.84	0.83	0.83	24066
weighted avg	0.99	0.99	0.99	24066

Epoch 6 Training Loss: 1.8240303672961335
Validation Loss: 1.8264363209406536

For test data, the following are the scores

Loss: 0.13220558762550355

	precision	recall	f1-score	support
ADP	0.96	1.00	0.98	1434
PRON	0.85	0.98	0.91	392
PROPN	0.99	0.99	0.99	1567
CCONJ	1.00	0.98	0.99	109
AUX	0.93	1.00	0.96	256
DET	0.99	0.87	0.92	512
ADJ	0.92	0.97	0.94	220
INTJ	1.00	1.00	1.00	36
PART	0.00	0.00	0.00	56
PAD_POS	1.00	1.00	1.00	12791
NOUN	0.99	0.99	0.99	1166
NUM	0.96	0.94	0.95	127
VERB	0.99	0.96	0.98	629
ADV	0.91	0.68	0.78	76
accuracy			0.99	19371
macro avg	0.89	0.88	0.89	19371
weighted avg	0.99	0.99	0.99	19371

Hyperparameters:

Number of epochs: 6

learning_rate : 0.01

Batch size: 64

loss_fn : CrossEntropyLoss

optimizer: Adam

embedding_dim : 200

Hidden layer size: 150

Number of layers: 1

Network: Embeddings, LSTM, Fully connected layer

The model underwent several epochs during training and at the epoch where loss is minimum, the model is saved.

Experimenting by tweaking parameters- When the number of layers is increased, the model performs well but then it has taken longer time to train. In case of increasing embedding dimensions, hidden layer size dimension (LSTM), the model performed well on test data and validation data. Also, as the number of epochs increased, the performance of the model has improved.

Instead of learning embeddings, using pre-trained embeddings (like word2vec, Glove) could improve the model's performance and the train time would be lower than the previous case.

Here, though the model is performing well on test data, it is not performing well on complex sentences, data that isn't similar to the train data. This is because there is scarcity of training data (less traindata). In order to get better performance, we need to train the model on different datasets.