Assignment 3 Summary

Method Description: In this assignment, I picked the Named entity detection (NED) task to recognize the entity types in Twitter data. In natural language processing systems, NED is a fairly common task. It is used to extract text entities such as people, places, and organizations etc. I wanted to try a different coding environment this time, so I started with the nlp toolkit by reading Extracting Information from Text (Chapter 7) from the nltk ebook, but I couldn't get the evaluation part of the code to work. Due to a lack of time, I decided to experiment with tensorflow libraries in a Colab coding environment, as I found a wealth of resources related to NED tasks online. I was able to complete this assignment successfully using Tensor flow while also learning a new coding environment platform.

Implementation: For this entity recognition assignment, we are given a corpus of text derived from Twitter data. A BIO label has been assigned to each word in a sentence in the dataset. For the data preprocessing, I followed these two tutorials [1] [2]. Training, validation, and test sets of data were already given separately. The training data tokens were 2394, the validation tokens were 1000, and the evaluation tokens were 2394. I used two mappings to train a neural network: 1) token id: address the row in the embeddings matrix for the current token; 2) tag id: one-hot ground truth probability distribution vectors for calculating the loss at the network's output. Special tokens used in this assignment include the UNK> token for out of vocabulary tokens and the PAD> token for padding sentences to the same length while creating batches of sentences.

Create a recurrent neural network (RNN) Model: In this assignment, for each token in a sentence, I first built an LSTM network that will generate a probability distribution over tags. I used Bi-Directional LSTM to account for both the token's right and left contexts (Bi-LSTM)[3]. To perform tag classification, a dense layer is applied on top. I used softmax on the last layer to compute the neural network's actual predictions, and argmax to find the most probable tags. I used cross-entropy loss during training, which is easily implemented in TF as cross entropy with logits and to optimize the loss, I used Adam.

Evaluation: For the evaluation phase in this task, I used two functions: 1) *predict_tags:* generates predictions using a model and converts indices to tokens and tags; 2) *eval_conll:* measures precision, recall, and F1 for the results. For hyperparameters, I used *batch_size*: 32; epochs: 4; starting value of *learning_rate*: 0.005, *learning_rate_decay*: a square root of 2; *dropout_keep_probability*: try several values: 0.1, 0.5, 0.9. The following are the test set quality results as compared to the baseline framework given in the assignment:

Sr. No.	***Entity Type***	***Baseline Model (CRF)***	***BiLSTM Model***
1	Precision	40.34%	96.35%
2	Recall	32.22%	96.93%
3	FB1 score	35.83	96.63

The following is a screenshot of the detail results from the designed model's test set:

```
---------- Test set quality: ----------
processed 48863 tokens with 1496 phrases; found: 1505 phrases; correct: 1450.

precision:  96.35%; recall:  96.93%; FB1:  96.63

        company: precision:  83.14%; recall:  83.63%; F1:  83.38 172

        facility: precision:  94.39%; recall:  97.12%; F1:  95.73 107

        geo-loc: precision:  99.28%; recall:  99.64%; F1:  99.46 277

        movie: precision:  100.00%; recall:  100.00%; F1:  100.00 34

        musicartist: precision:  98.21%; recall:  100.00%; F1:  99.10 56

        other: precision:  96.05%; recall:  97.33%; F1:  96.69 228

        person: precision:  99.12%; recall:  100.00%; F1:  99.56 453

        product: precision:  100.00%; recall:  97.94%; F1:  98.96 95

        sportsteam: precision:  94.12%; recall:  94.12%; F1:  94.12 51

        tvshow: precision:  96.88%; recall:  91.18%; F1:  93.94 32
```

References:

[1] Preprocessing data with TensorFlow Transform. Accessed in 2021 [online]. Available: https://www.tensorflow.org/tfx/tutorials/transform/census
[2] Named Entity Recognition Tagging. Accessed in 2021 [online]. Available: https://cs230.stanford.edu/blog/namedentity/
[3] Twitter Entity Recognition. Accessed in spring 2020 [online]. Available: https://www.kaggle.com/amoghjrules/twitter-entity-recognition-using-bilstms