

Name: Haiyu Lyu
StudentID: z5142340

Task1:

1. The Implementation of 'evaluate ()':

In order to implement this function, I calculated FN, FT and TP. In the calculations section, I first traverse the golden_list to identify the start and end addresses of a block. Then compare this block with the same position in the predict_list. If it is not the same or the next of the block in the predict_list is 'I-TAR' or 'I-HYP', then FN is incremented by 1. Otherwise, TP is increased by 1. Similarly, the same operation can be used to traverse the predict_list to get the value of FP. Then the formula calculates Precision and Recall. Finally we can find the F1 Score.

2. The performance of modification:

Before using 'evaluate ()', the value of F1 measured by training is 0.7956.

But after using 'evaluate ()', the data I tested is still 0.7956.

The modified version does not affect performance

Task2:

1. The Implementation of 'new_LSTMCell ()':

I changed the formula

$$cy = (forgetgate * cx) + (ingate * cellgate)$$

to

$$cy = (forgetgate * cx) + ((1 - forgetgate) * cellgate),$$

and then imported the corresponding package to complete the task.

2. The performance of modification:

After using 'new_LSTMCell ()', the training speed has dropped a lot, and the F1 Score has also decreased by about 0.01.

Task3:

1. The Implementation of 'get_char_sequence ()':

First reduce the dimension. Since the parameters passed in are three-dimensional, what we need is two-dimensional data based on the word structure. Embedding and sorting the new two-dimensional tensor. In this way, the processed data is packaged and put into the char_lstm model. The last two units of the result are merged and sorted. Finally, the results are restored to a three-dimensional sentence structure.

2. The performance of modification:

After using 'get_char_sequence ()', the F1 Score has increased to about 0.8101 and the training time becomes longer.