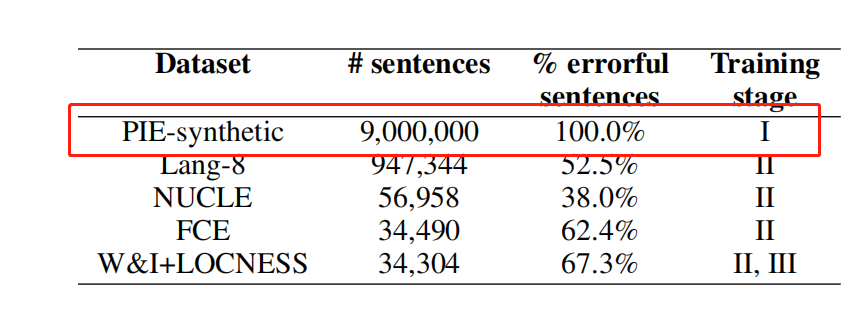
Method 1: Optimise the network with bert feature extraction + bi-directional GRU + classification HEAD. Compare the performance of GRU without and with GRU added

Training environment: NVIDIA RTX 2080TI

Stage1: PIE data is used for training, bea-2019 data is used as test, the amount of PIE data is as follows



The training parameters for both the Bert network and the Bert\_GRU network are:

tune\_bert: 1

skip\_correct: 1

skip\_complex: 0

max\_len: 50

batch\_size: 64

tag\_strategy: keep\_one

cold\_steps\_count: 0

cold\_lr: 1e-3

lr: 1e-5

predictor\_dropout: 0.0

lowercase\_tokens: 0

pieces\_per\_token: 5

vocab\_path: data/output\_vocabulary

label\_smoothing: 0.0

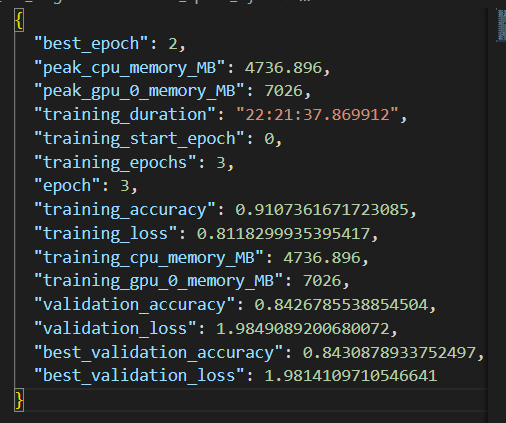
patience: 3

transformer\_model: bert

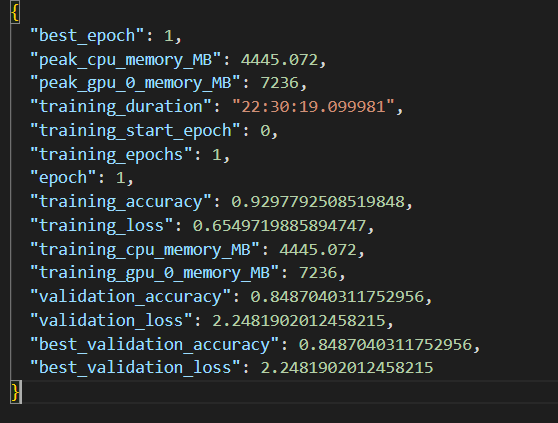
special\_tokens\_fix: 0

Due to the excessive amount of PIE data, the model is almost in a converged state after 2 epochs of training.

The acc results for the PIE training set and BEA2019 test set for the Bert model are as follows.



The results of the Bert+GRU model are as follows:



The training time for both is almost the same, but Bert+GRU is significantly higher than the bert model on both the training and test sets.

For inference testing, both bert and bert\_GRU use the following parameters:

iteration\_count = 2

additional\_keep\_confidence = -0.68

additional\_del\_confidence = -0.84

min\_error\_probability = 0.04

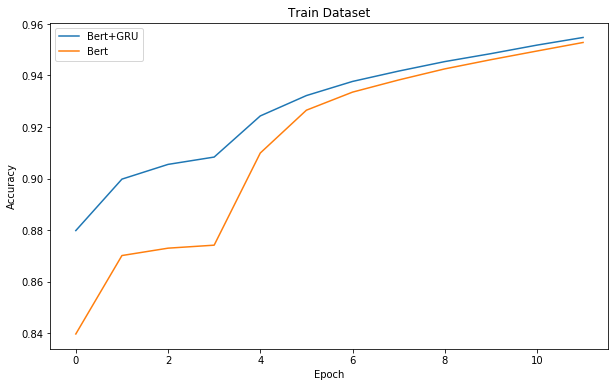
The inference results of the two models for the two datasets are shown below:

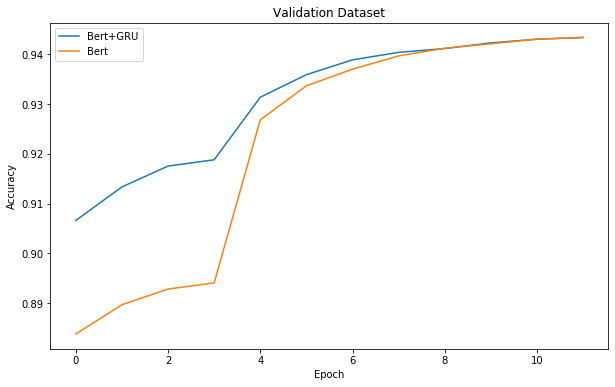
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | CoNLL-2014(test) | | | BEA-2019(dev) | | |
|  | P | R | F0.5 | P | R | F0.5 |
| Bert | 0.3065 | 0.1557 | 0.2567 | 0.2296 | 0.1131 | 0.1904 |
| Bert+GRU | 0.3424 | 0.1953 | 0.2975 | 0.2778 | 0.1458 | 0.2352 |

Stage2: Since the reproduction results of this paper are inconsistent with the original paper, this paper is not compared with the original paper. In order to better verify the bert+GRU model performance advantage. The second experiment in this paper is only training test on small data on fce dataset. Control variable method to analyse the performance of the two models. In the future, if there is time to consider reproducing the results of the original paper:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | CoNLL-2014(test) | | | BEA-2019(dev) | | | FCE(dev) | | |
|  | P | R | F0.5 | P | R | F0.5 | P | R | F |
| Bert | 0.2163 | 0.1114 | 0.1820 | 0.2016 | 0.1178 | 0.1765 | 0.2211 | 0.1176 | 0.188 |
| Bert+GRU | 0.2213 | 0.1467 | 0.2008 | 0.2454 | 0.1508 | 0.2181 | 0.2365 | 0.1082 | 0.1912 |

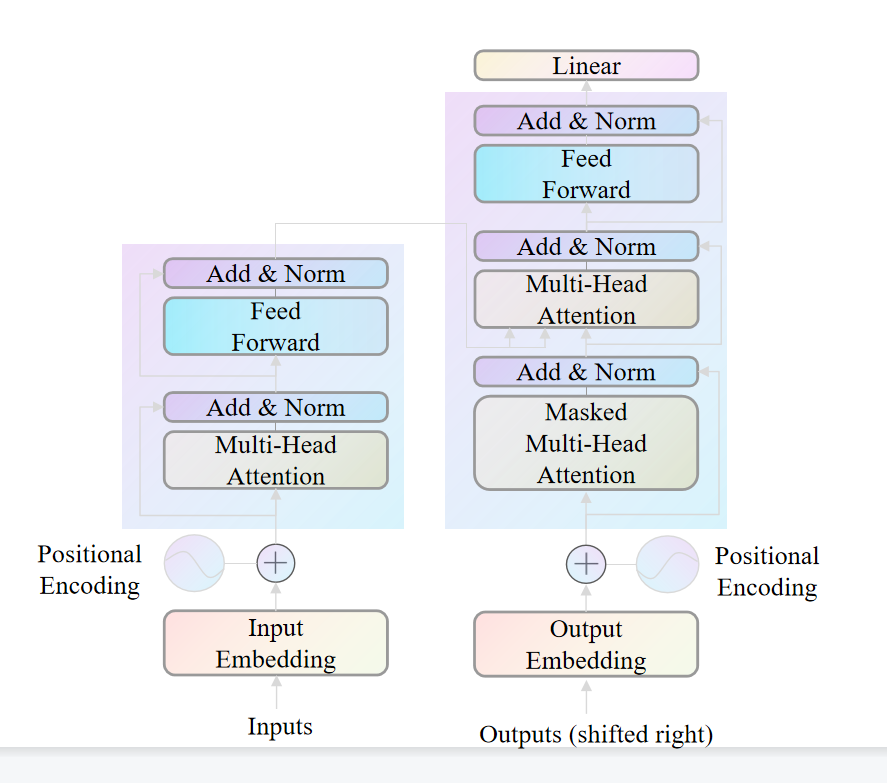
Performance comparison of bert and bert+GRU for each epoch during model training. Regardless of the training set and validation set, it can be found that the convergence speed of Bert+GRU is faster. Meanwhile, when the model converges, although the accuracy of the two models in the validation set is the same, the accuracy of the training set of bert+GRU is a bit higher:



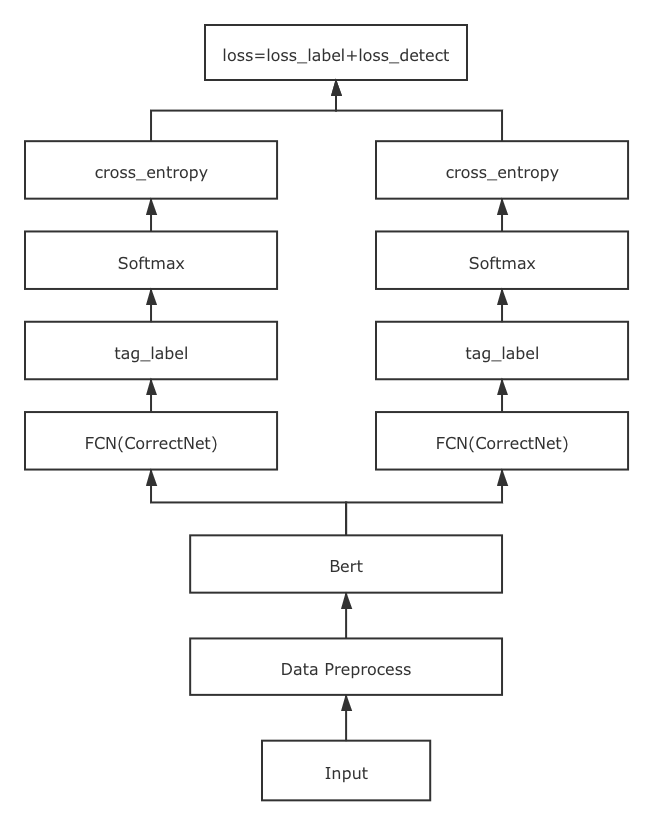


Network Architecture

Bert's network structure:



GECTOR Original Network:



GECTOR Improvement Network:

