# Lab Report of Exercise 4

## Approach

1. **Get Datasets**

   Load the training and validation sets from DataSets **'clue'** / **'tnews'**, which is in Chinese;

   Get 1000 instances, 5000 instances and 500 instances from the train section as 3 different training sets;

   Get 2000 instances from the validation section as the evaluation set.

2. **Tokenize and Vectorize Data**

   Use the BertTokenizer of **'bert-base-chinese'** to tokenize the sentences, transform them into tensors;

   One-hot encode the labels.

3. **Fine Tune Bert Model**

   Feed training data into the pre-trained **'bert-base-chinese'** model of BertForTokenClassification class;

   Train all layers of the model or freeze embeddings and only train the top classification layer.

4. **Predict Labels on Evaluation Set**

   Use the tuned models on evaluation set;

   Get the output / logits;

   Deal with the output to get the predicted labels.

5. **Compute Macro and Micro F1 Score**

   Use the predict labels and true labels to calculate the macro-f1 and micro-f1.

## Technical Description

Table 1. Descriptions of Defined Classes and Functions

| Class / Function | Description |
|---|---|
| Class DataSet | ● Load datasets from train or validation sets<br>● Get data with different number of instances |
| Class TokenizeVectorizeData | ● Use BertTokenizer to tokenize sentences<br>● One-hot encode labels<br>● Get data loader |
| Function tune_bert(freeze=True/False) | ● Define hyperparameters<br>● Tune model<br>● Or freeze model and only tune the top classifier layer |
| Function predict | ● Feed evaluation data into model to get the output<br>● Deal with the output to get the predicted label<br>● Get the true label |

## Inspect Data

### Example of Dataset 'clue'/ 'tnews'

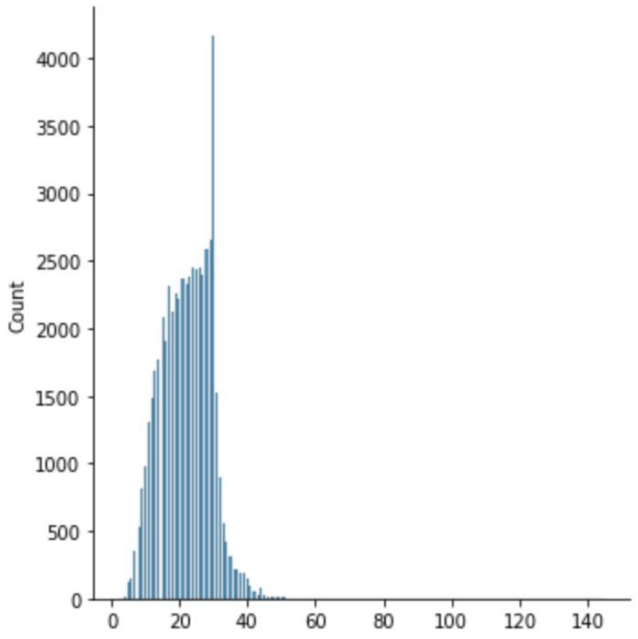
```
{'idx': 0,
 'label': 7,
 'sentence': '上课时学生手机响个不停，老师一怒之下把手机摔了，家长拿发票让老师赔，大家怎么看待这种事？'}
```

The dataset has 15 unique labels, from 0 to 14.

## Word Count Distribution of Sentences

I check the number of words in sentences in the train dataset (see Figure 1 below), Most of the sentence lengths are below 40, so I choose **max_length=40** when tokenizing the sentences.

Figure 1. Distribution of Sentence Lengths



# Results

I set the **batch_size=8, optimizer=AdamW(params=model.parameters(), lr=1e-5), epochs=30**;

Trained the bert model on different training sets with different number of instances;

Predicted labels on evaluation data, the evaluation macro f1 and micro f1 of each model are as follow:

Table 2. Model Comparison

| **Model** | **Macro f1** | **Micro f1** |
|---|---:|---:|
| Tuned on 1000 sentences | 0.43 | 0.47 |
| Tuned on 5000 sentences | **0.44** | **0.47** |

|  |  |  |
|---|---:|---:|
| Tuned on 500 sentences | 0.09 | 0.15 |
| Freeze embeddings, tuned top layer on 1000 sentences | 0.01 | 0.05 |
| Freeze embeddings, tuned top layer on 5000 sentences | 0.01 | 0.01 |

(1) Fine tuned model on 5000 sentences has the best performance on evaluation set, the macro f1 is 0.44 and micro-f1 is 0.47, which are  a bit better than fine tuned model on 1000 sentences;
(2) If we freeze the embeddings and only tune the top layer, the performance is much worse than that of the non-freeze models. That's because the frozen base model can't learn from the training dataset.
(3) The frozen model tuned on 1000 sentences is a little better than the frozen model tuned on 5000 sentences, which may be because the model tuned on 5000 sentences has overfitted when training for only one layer.

## Answers to Questions

1. When initializing the BertForTokenClassification-class with BERT-base you should get a warning message. Explain why you get this message.

   When using the bert base model for downstream tasks, such as binary and multi-class classification, next sentence prediction and etc, the pre-trained base model can be loaded with different classes such as BertForTokenClassification and BertModel. The layers, number of encoders, last layer and etc are different for each class. That's why we get a warning message, noticing us this information.

2. Which model performed best on the evaluation set?

   Fine tune model on 5000 sentences performs the best on the evaluation set, the macro f1 score is 0.44  and micro f1 score / accuracy is 0.47.

3. Are there differences between f1-micro and f1-macro score? If so, why?

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

For Macro f1 and Micro f1, the precision and recall are different.

**Macro f1:** get TP, FP of each class, calculate the precision and recall of each class, average them to get average precision and recall, then use the average precision and recall to calculate macro f1 score.

**Micro f1:** get TP, FP of each class, average them to get average TP, FP;  then calculate the precision and recall, then use the precision and recall to calculate micro f1 score.

So for binary classification ,the micro f1 and macro f1 are the same;  however, for multi-class classification, macro f1 and micro f1 are different, and the micro f1  is equal to accuracy.

Note: TP=number of true positive, FP=number of false positive

4. How large is the performance gap between 500 and 5’000 sentences for fine tuning?

   From Table 2. We can see, the model tuned on 5000 sentences has better performance than the model tuned on 500 sentences. The micro-f1 gap between the two is 0.32, the macro-f1 gap between the two is 0.35. However, that’s the result when training epochs is 30. When increasing training epochs, the gap would become narrower.

5. Is it better to freeze or not to freeze the embeddings?

   It’s better NOT to freeze the embeddings. As the base bert model is trained on other corpus, if we freeze the embeddings and use them directly for the classification, it’s not customized for the new task with new training data, so it cannot learn much from the training data. That’s why it’s better not to freeze the embeddings but train the model on the training data for downstream tasks.