Ex 04 - Sequence Classification with BERT

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

```
Some weights of the model checkpoint at bert-base-german-cased were not
     used when initializing BertForTokenClassification: [...]

- This IS expected if you are initializing BertForTokenClassification from
     the checkpoint of a model trained on another task or with another
     architecture (e.g. initializing a BertForSequenceClassification model
     from a BertForPreTraining model).

[...]

You should probably TRAIN this model on a down-stream task to be able to
     use it for predictions and inference.
```

I copied parts of the message to make sure that it is clear what I am talking about. What I understand is the following: the BERT model `'bert-base-german-cased'` that I am using has been pre-trained on any other architecture, but *not* on `BertForTokenClassification`. This means that some weigths of some layers as mentioned in the first sentence of the warning message, cannot be initialized because they are part of the pre-trained model but not of the one that I am using. There might also be some weigths missing that my model needs. Those will be initialized randomly. The last sentence is probably the most important as it means that I have to fine-tune the model such that it works well (which can be explained with the missing weights that were initialized randomly, those have to be trained first before they can produce good results).

2. Which model performed best on the evaluation set?

Model	f1 micro	f1 macro
Fine-tuned with 1'000 sentences	0.916	0.348
Fine-tuned with 5'000 sentences	**0.932**	**0.490**
Fine-tuned with 1'000 sentences and frozen embeddings	0.511	0.138
Fine-tuned with 5'000 sentences and frozen embeddings	0.815	0.180

The model that performed best on the eval set is the model that was fine-tuned with 5'000 sentences and non-frozen embeddings. I think this makes sense as 1'000 sentences are probably not enough to fine-tune the model. Also, when training with frozen embeddings, the model

with 5'000 sentences performed better than the smaller one. That the models that use the frozen embeddings perform worse over all is not surprising to me either, as I will explain below.

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

The micro score is really high for all models. I think this is because the micro score does not care about imbalance of classes. It basically calculates how much of the entire dataset was labelled correctly no matter what class the individual sample belongs to. I know that the f-score is a combination of precision and recall and that my explanation below does not reflect in full detail the exact working of this score. However, I only intend to explain how I understand the difference between micro and macro score on a high level, not including all the calculation details.

Let's say of all the samples in the test set, 70% are labelled with the same label. I will call this **label 1**. The remaining 30% are distributed among the other four labels (**labels 2-5**). If 90% of the samples for label 1 are labelled correctly (i.e. 63% of the entire dataset (90% of 70%)), a large amount of the dataset will be labelled correctly. Let's further assume that for the four labels with less samples, the predictions are not as good. Let's say only one third of these samples are labelled correctly. However, as these labels contribute anyway only 30% to the dataset, this means that only 10% of the entire dataset will be labelled correctly for all the remaining labels (labels 2-5). Still, the evaluation will say that 73% of the entire dataset was labelled correctly. But this is somehow not reflecting the real performance of the model, because the predictions for all classes *except one* are really bad.

The described scenario is exactly what is the case in this NER task. Most of the labels are 'O' (most probably those make up even more than just 70% of the entire dataset). However, we are actually interested in how the model labels the NER tags and if it recognized those. We want to punish the model if the above scenario happens and most of the smaller classes are predicted wrongly ('PER', 'LOC' and 'ORG'). This can be done using the macro score. And this is why the micro and macro score differ a lot. The macro score calculates how many samples for each class were labelled correctly and averages the performance of each individual score for each label to produce a final score. I think that for this task, the macro score gives a good intuition about how well the model performs on predicting the smaller classes. If this score is really low, this can mean that the model just predicts the majority class.

4. How large is the performance gap between 1'000 and 5'000 sentences for finetuning?

The performance for the larger set is clearly better. Both model configurations performed a lot better when trained on 5'000 sentences rather than 1'000. Generally, I think it makes sense that

the performance for the larger set is better as the model can be fine-tuned better with more data. I think it would be interesting to see if the model will improve a lot more when trained on e.g. 10'000 sentences or if the performance will flatten out.

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

The scores for the models with the frozen embeddings are a lot worse than for the other models. But due to the explanations given in the first questions, this is not surprising. If the embeddings are frozen, the model is not fine-tuned on the dataset which means that it cannot 'adapt' to the dataset. So, it will use the random weights for those layers that were not in the architecture that the model was originally trained for. It is therefore better not to freeze the embeddings (at least for this task and this dataset). I would assume that for other tasks and other datasets this might also be different.

Resources used other than the links on the Assignment:

- https://github.com/huggingface/transformers/issues/5421