Exercise 5 - Sequence and Anger Regression using Transformers

Task 1: Named Entity Recognition using Bert

Goal:
The goal is to fine-tune Bert-base model for Named Entity recognition task for Hindi language. Three fine-tuned versions are implemented in this exercise:

1. Fine-tuned with 1,000 sentences
2. Fine-tuned with 3,000 sentences
3. Fine-tuned with 3,000 sentences and frozen embeddings

Dataset:
We used a polyglot-ner dataset for Hindi language. We created 2 training datasets with 1000 sentences and 3000 sentences respectively. We created one evaluation set with 2000 sentences. Evaluation set remain constant for all three fine-tuned versions in order to fairly compare results. There are 4 NER labels corresponding to words in sentences.

1. LOC: Locations
2. ORG: Organisations
3. PER: Persons
4. O: Not a Named Entity

Pretrained Bert-base model:
We used "Davlan/bert-base-multilingual-cased-ner-hrl" model as a pretrained model for our NER task.

Data Preprocessing:
We performed truncation and padding of maximum length 128.

Training:
We trained each fine-tuned model with batch-size=8. We used AdamW optimizer with a learning rate of 1e-5. We fine-tuned 3 models:

1. Trained with 1,000 sentences
2. Trained with 3,000 sentences
3. Trained with 3,000 sentences and frozen embeddings

Evaluation:
All three fine-tuned models were evaluated on an evaluation set. We again used 3 metrics to evaluate the performance of the model: accuracy, f1-macro, f1-micro.

Fine-tuned Version	**Test accuracy**	**Test F1-macro**	**Test F1-micro**
1000 sentences	0.99045	0.44095	0.99045
3000 sentences	**0.99192**	**0.45527**	**0.99192**
3000 sentences with frozen embeddings	0.99152	0.44421	0.99152

Answering highlighted questions:

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

When initialising the BertForTokenClassification-class with BERT-base, we received following warning:

"Some weights of BertForTokenClassification were not initialised from the model checkpoint at bert-base-cased and are newly initialised: ['classifier.bias', 'classifier.weight'] You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference."

The warning we received is likely triggered because of a mismatch in the task type. Bert-base, which is pretrained on a masked language model (MLM) objective for predicting missing words in sentences, has a token classification head (classifier) that was not part of its original training objective. When we initialise a BertForTokenClassification model using bert-base-cased, the token classification head's weights (classifier.bias and classifier.weight) are not present in the original pretrained checkpoint. These weights are specific to token classification tasks, where the goal is to predict the labels of individual tokens in a sequence (e.g., named entity recognition).

Please note that for solving this exercise, we decided to use the Davlan/bert-base-multilingual-cased-ner-hrl model as a pretrained model since it was specifically trained for Named Entity Recognition, here we did not get any warning.

Which model performed best on the evaluation set?

Second model performed the best on the evaluation set which was fine tuned with 3000 sentences.

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

Yes, there are huge differences between f1-micro and f1-macro scores. One of the possible reasons is class imbalance. In imbalanced datasets, F1-macro may be lower due to its equal consideration of all classes, including minorities. If the model faces challenges with minority classes, their lower F1 scores impact the macro-average negatively.

How large is the performance gap between 1’000 and 3’000 sentences for fine tuning?

On the evaluation set, the performance gap is not huge! For F1-macro, the gap is 0.01432 and for F1-micro, the gap is 0.00147.

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

Based on the model's performance on the evaluation set, it is better not to freeze the embeddings.

Task 2: Emotion Regression: How angry are you?

Goal:
The goal is to fine-tune the Transformer model for the Affect of Anger in the English Regression task.

Why and what hypothesis led you to choose this architecture?
For anger regression on Twitter data, we hypothesise that larger transformer models like BERT-base or RoBERTa-base or XLNet-base are good choices for Anger Regression Task. Larger models excel at capturing nuanced language and contextual information in diverse and informal tweets. Pretraining on extensive text data, a characteristic of larger transformers, is expected to yield superior representations for regression tasks, enhancing the model's ability to capture anger intensity and nuances in tweets

Defend why you chose/chose not to do any preprocessing?
In the Twitter dataset, various preprocessing steps were necessary and therefore we chose to perform preprocessing. Instances included usernames mentioned in the data (e.g., @Mothercarehelp), several hashtags such as #simple, and the presence of special characters. Consequently, we opted to remove these elements, resulting in a cleaned dataset.

Model Exploration Experimentally:
We explored 3 models namely, roberta-base, bert-base, and xlnet-base. We fine-tuned each model. We trained all the models for 6 epochs. We evaluated the model on the testing dataset provided. Following are the Pearson r scores on the training dataset:

Model Name	Pearson r scores
roberta-base	0.9542
bert-base	**0.9771**
xlnet-base	0.9467

Based on the performance on the training set, we chose the bert-base model for evaluation on the test set.

Take the best performing model and evaluate it on the test set and report your test set results:

The best performing model was bert-base model. We evaluated this model on given test dataset. Pearson r scores for bert-base model = 0.7341.

Briefly explain what Pearson-R evaluates and what it tells about the performance of our model:

The Pearson correlation coefficient (Pearson-R) evaluates the linear relationship between true values and predictions, providing a measure of how well their relationship can be represented by a straight line. The coefficient ranges from -1 to +1, where 0 indicates no linear correlation, +1 implies a perfect positive linear relationship, and -1 implies a perfect negative linear relationship. In the context of our model's performance, a Pearson-R value of 0.7341 suggests a strong positive linear correlation between the predicted and actual values. This indicates that as the predicted values increase, the actual values also tend to increase, demonstrating a substantial agreement between the model's predictions and the true values. In summary, a Pearson-R of 0.7341 suggests a strong positive linear relationship, signifying good predictive performance of the model.

Did your results support the hypothesis? Why/Why not?

The result i.e. Pearson-R score of 0.7341 supports our hypothesis as it shows a positive linear relationship signifying good predictive performance of the model. A positive Pearson correlation in this range suggests that the model's predictions align well with the observed data, supporting the effectiveness of the model in capturing the underlying patterns in the dataset. Therefore, our results support our hypothesis.