Partial Freezing Of MLMs For PoS Tagging

Percy Brown *Prof. En-Shiun Annie Lee*

1 Introduction

In extending the capability of a pre-trained model to a new but related task, it is important to ensure that we prevent catastrophic forgetting. By partially freezing some layers of a pre-trained model, we can preserve existing knowledge from the pre-trained models while adapting specific layers to the new task. It is also advantageous in situations where there is no large amount of data for the new task as well as limited computational resources. The report briefly discusses the approach and results of a baseline distilbert model as compared to a partially frozen one. DistilBert is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40% fewer parameters than bert-base-uncased and runs 60% faster while preserving over 95% of BERT's performances as measured on the GLUE language understanding benchmark.

2 Approach

2.1 Data Pre-Processing

The dataset of the English Language was derived from Universal Dependencies, consisting of the training, dev and test datasets. The datasets were transformed into a list of lists, where each sublist represents a sentence or phrase. Each sublist is a list of tuples containing where each tuple contains a token and corresponding pos tag. This makes the dataset suitable for PoS tagging, where the distilled model can learn the tokens and corresponding pos tags. The unique tags are then retrieved, with a pad tag joined in the 0th slot by convention. Next, mappings between part-of-speech tags and their corresponding indices and between indices and their corresponding part-of-speech tags are created for converting between part-of-speech tags and their numerical representations. Additionally, the device is set to Cuda or CPU depending on whether Cuda-enabled GPU is available for faster execution.

2.2 Tokenizer Initialization and Preliminaries

The DistilBert tokenizer is initialized, along with a class responsible for organizing the tokens into sequences and ensuring uniform padding. For instance, a class preprocesses the input sentences for part-of-speech tagging tasks, tokenizing words, converting them into token IDs, and organizing them into sequences along with their corresponding part-of-speech tag IDs and first piece indicators. It encapsulates the data processing logic required for training and evaluation. Then, a helper function ensures that all sequences within a batch have the same length by padding them with zeros up to the length of the longest sequence in the batch. This padding is necessary for efficient batch processing, particularly in deep learning models where inputs are typically processed in batches and require uniform dimensions.

2.3 Baseline Model Definition, Training and Evaluation

Here, a class is created to define the neural network, which utilizes the DistilBert model for token embeddings and a linear layer for classification, together with a softmax activation to produce class probabilities. It computes the logits, true labels, and predicted labels during the forward pass. Depending on the training or evaluation phase, it switches the DistilBert model's mode accordingly.

Training involves a single step using a batch of data from a training dataset iterator. It computes the loss, performs backpropagation to compute gradients, and updates the model parameters using the optimizer. This process is repeated for each batch in the training data. Additionally, it prints the loss for monitoring purposes at regular intervals. The dataset iterations are divided into batch sizes of 8 and the Adam optimizer was used to scale the learning rates adaptively.

Evaluation is performed on a validation set iterator and further established on unseen test data. It computes predictions, saves the results to a file and calculates the accuracy metric based on the true and predicted labels.

2.4 Partially Frozen Model Definition, Training and Evaluation

The model definition, training and evaluation functions were similar to the baseline model implementation. However, during this phase, selected layers of the DistilBert model are frozen. The downstream task under consideration is new but similar to other tasks performed by the DistilBert model. Therefore, it was deemed appropriate to fine-tune the embedding layer to adapt its word representations to the specific characteristics of PoS tagging. With the DistilBert model having 6 transformer layers, the first 3 layers of the transformer layers are most likely going to capture low-level features, word relationships and language patterns, which are likely applicable to many downstream tasks, of which PoS tagging could be a beneficiary. Therefore, the first 3 layers were frozen, while the remaining layers were assumed to capture higher-level semantic features, hence fine-tuning these layers along with the classification head may help the model learn task-specific representations more effectively.

3 Results

3.1 Training Losses for Baseline and Partially Frozen Models

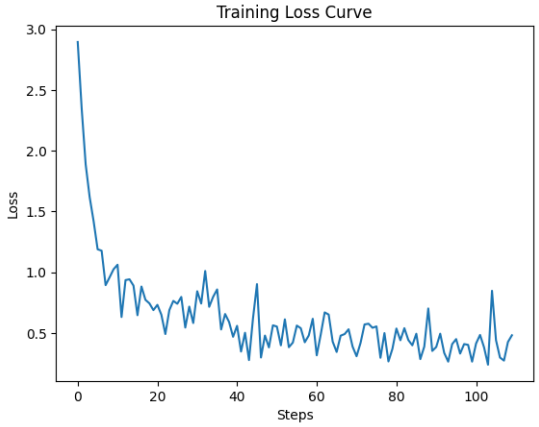

Figure 1: The Training Loss Curve for the Baseline Model

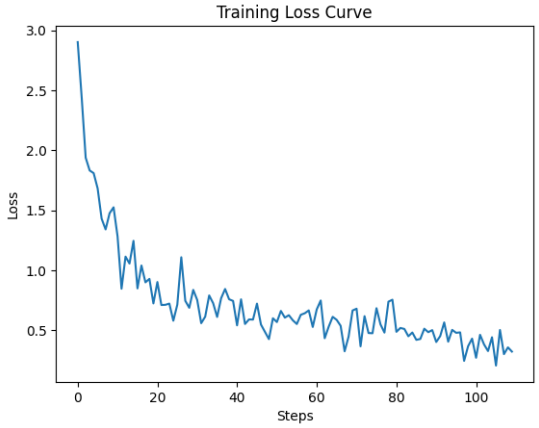

Figure 2: The Training Loss Curve for the Partially Frozen Model

From the figures 1 and 2, both training loss curves indicate instability in the training process which could be a result of the learning rate, batch size or optimization algorithm. Nonetheless, both curves show

promising signs of convergence in subsequent iterations.

3.2 Performance (Accuracy) Comparisons between Baseline and Partially Frozen Model

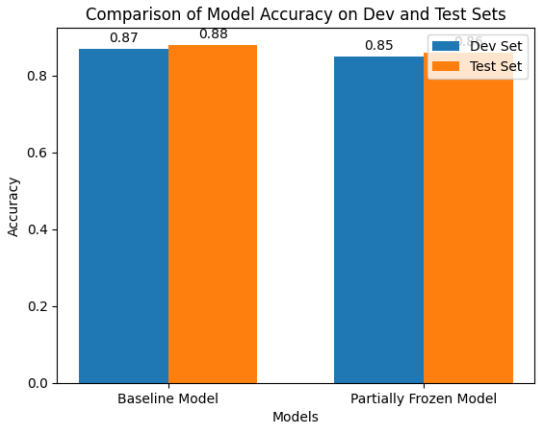

Figure 3: The Training Loss Curve for the Partially Frozen Model

As mentioned earlier, both models were evaluated on validation and test sets of approximately equal sentence lengths, with each result saved in a text file for reference. As shown in the figure 3, the accuracy in results on both sets of data for both models were almost the same, with a difference of 0.01% improvement on the testing set. However, it can be seen that the baseline model performs slightly better as compared to the partially frozen model, with about 0.02% differences in performance results.

4 Conclusion

Although the difference in performance is not significant, it is evident that the baseline DistilBert model consistently exhibited slightly superior performance compared to the partially frozen variant across both the validation and test datasets. Despite the expectation that partial layer freezing would enhance the model's adaptability to the PoS tagging task, the experimental outcomes demonstrated otherwise. An important question therefore arises: Could it be that the PoS tagging task itself may not align well with the objectives of partial layer freezing?

One potential explanation for the underperformance of the partially frozen model could be attributed to the intricate interplay between frozen and trainable layers. It is plausible that freezing certain layers disrupted the model's ability to learn task-specific representations effectively, leading to suboptimal performance. Therefore, future work could explore other types of distilled models and alternative fine-tuning strategies to assess the impact of partial layer freezing on tasks such as PoS tagging.