Partial Freezing of MLMs for PoS Tagging

Abstract

This report investigates the impact of partial layer freezing on fine-tuning the `distilbert-base-multilingual-cased` model for multilingual Part-of-Speech (PoS) tagging. We prepared a subset of the `Universal Dependencies English EWT and Naija NSC corpus` [3], established a baseline by fully fine-tuning all parameters, then conducted four additional experiments: freezing all encoder layers, freezing the first 2 layers, freezing the first 4 layers, and alternating-layer freezing. Performance was evaluated via development-set accuracy, convergence behavior, trainable-parameter efficiency, and training time. We saw that the Naija NSC corpus with baseline evaluation accuracy was 98.2% outperformed the English EWT baseline with 96.4% and this could be credited to the chosen multilingual model's capability capture the diverse linguistic patterns present in Low Resource Languages, like in our case of Nigerian Pidgin. Our findings also show that selectively freezing early transformer layers yields minimal accuracy loss (<0.5%) while reducing computation (by up to 42.5% for Nigerian Pidgin and 24.8% for English) and trainable parameters by up to 20.9% (for both English and Nigerian Pidgin), except for the case of freezing all the encoder layers, suggesting an optimal trade-off for resource-constrained settings.

Introduction

Background: Transformer-based language models, such as BERT [1]. and its variants like DistilBERT [4], which is used in this study, have established new performance benchmarks across a wide range of NLP tasks, including text classification, named-entity recognition, and dependency parsing. These models leverage self-attention mechanisms to capture contextual relationships at multiple levels of abstraction. Pre-training on massive multilingual corpora equips them with rich syntactic and semantic representations. However, the process of fully fine-tuning all model parameters for a downstream task can be prohibitively expensive in terms of computation, memory footprint, and inference latency, especially when scaling to multiple languages and resource-constrained environments.

Motivation: Partial layer freezing offers a compromise between full fine-tuning and feature extraction. By selectively freezing the weights of certain encoder layers, one can preserve the general linguistic knowledge captured during pre-training while dedicating compute resources to adapting the remaining layers for a specific task [2, 5]. This approach has the potential to reduce training time, decrease GPU memory consumption, and mitigate catastrophic forgetting of pre-trained representations.

Methodology

Overview

This study systematically evaluates five training configurations of the distilbert-base-multilingual-cased model for multilingual PoS tagging under a unified experimental setup. Here is the approach followed:

1. Establish a baseline by fully fine-tuning all model parameters on the Universal Dependencies English EWT and the Naija-NSC subsets.
2. Implement and compare four partial freezing strategies which includes freezing all encoder layers (classification head-only training), feezing the first two encoder layers, freezing the first four encoder layers and applying an alternating freeze pattern (freeze every other layer).
3. Measure the development-set accuracy, training time, and trainable-parameter efficiency for each strategy.

Data Preprocessing and Tokenization

We used the entire UD English EWT data, containing 12,544 training, 2,001 development, and 2,077 test sentences, and the entire UD Naija NSC data, containing 7,279 training, 990 development, and 972 test sentences. For each experiment, we used a fixed random seed (42) to ensure consistency across runs. We parse Conllu files to extract lowercased tokens and UPOS labels and then tokenize with the distilbert-base-multilingual-cased model. A `tag2id` mapping is created over 45 unique UPOS tags, and word-piece tokens are aligned to labels, marking subword continuation tokens as -100.

Model Configuration and Training

We fine-tune `DistilBertForTokenClassification` with output dimension equal to the number of UPOS tags ($\tilde{1}$34M parameters). The optimizer is AdamW with weight decay 0.01. Hyperparameters are kept constant to ensure uniformity across the strategies. Training and evaluation use HuggingFace's `Trainer` API, facilitating dynamic padding. Development-set accuracy is evaluated after each epoch.

Freezing Strategies

We reload the base checkpoint and apply a layer freezing function before each run. Each freezing strategy changes which parameters get updated during fine-tuning. The model we used for PoS tagging is the DistilBERT, which has 6 encoder layers and when layers are frozen, the remaining unfrozen layers are trained, cutting both memory and compute

roughly in half. We used the best practice of freezing the first layers, which hold lower-level syntathic information, as opposed to freezing the last layers ,which hold higher level semantic information, and could likely degrade performance when frozen, especially for token classification tasks like PoS tagging. The strategies we considered are as follows:

1. **Baseline (No Freeze):** Here, every parameter, from the word embeddings and all six transformer layers to the final classification head, is unfrozen and updated on the PoS-tagging data. This serves as our performance ceiling, demonstrating the best accuracy the model can achieve when it has full capacity to adapt, at the cost of maximum training time and memory usage.

2. **Head-Only Training (Freeze All Encoder Layers):** In this extreme, we lock down all six transformer blocks and train only the lightweight classification head. This tests the representational power of the frozen encoder as a purely fixed feature extractor. While it offers the greatest savings in compute and parameter updates, this gives the worst accuracy among all, since the head must shoulder the entire learning burden.

3. **Freeze First 2 Layers:** Here, we freeze the bottom two layers (layers 0 and 1), which typically capture low-level token and syntactic cues, and fine-tune the remaining four layers plus the head. By preserving the foundational linguistic representations and allowing deeper semantic features to adapt, this strategy often recovers nearly baseline accuracy with a clear reduction in trainable parameters and faster epochs

4. **Freeze First 4 Layers:** Taking freezing further, we lock layers 0–3 and only train the top two transformer layers along with the head. This approach preserves both the token-level and mid-level abstractions learned during premodel-training, while focusing adaptation on the highest-level semantics. It yields even greater compute savings, though at the cost of a slightly larger dip in accuracy compared to freezing only two layers.

5. **Alternating Freeze:** To strike a balance, we freeze every even-indexed layer (0, 2, 4) and fine-tune the odd layers (1, 3, 5) plus the head. This interleaved pattern maintains access to both low- and high-level representations throughout the network, cutting trainable parameters by roughly half. Empirically, it delivers a mix of efficiency and performance, often outperforming the contiguous "freeze four" configuration in accuracy while offering comparable savings.

Baseline Model Training Setup

For the Baseline setup, we fully fine-tune every parameter of the pretrained DistilBERT model on the UD Naija NSC and UD English EWT data. Exery experiment was performed on a single T4 GPU. Here is a break down of the baseline experiment.

1. **Model & Head:** We instantiate a `DistilBertForTokenClassification` with an output head sized to the 45 UPOS tags. This model comprises the embedding layers, six Transformer encoder layers, and a token-classification head.

2. **Hyperparameters:** For this experiment, we keep all the hyperparameters constant, not just for the baseline but also for every freezing strategy. Here are the hyperparameters used.

   Learning Rate: We used 5×10^{-5} AdamW optimizer, which allows fast adaptation during fine-tuning while avoiding instability.

   Batch Size: 16 tokens per GPU. This fairly fits within GPU memory constraints while providing stable gradients.

   Epochs: 5 full passes over the training subset. This number of epochs ensures convergence without overfitting.

   Weight Decay: Regularization to prevent overfitting by penalizing large weights.

3. **Training Pipeline** We use HuggingFace's 'Trainer' API along with 'DataCollatorForTokenClassification' for efficient dynamic padding. After each epoch, we evaluate on the development split to track progress.

This fully-fine-tuned baseline establishes our upper-bound performance and compute cost, against which all partial-freezing strategies are compared. It is worth noting that this same setup was used in every experiment.

Experimentation and Results for Naija Pidgin Data

For this section, we will focus our analysis primarily on the UD Naija NSC data. We will include the results obtained from the UD English data at the end.

Accuracy and Compute Comparison

Strategy	Dev Acc (%)	Params (M)	Param Save (%)	Time (s)	Time Save (%)
Baseline	98.2	134	0.0	388	0.0
Freeze All	93.2	92	31.3	196	49.8
Freeze First 2	98.1	120	10.4	249	35.8
Freeze First 4	97.7	106	20.9	223	42.5
Alternating Freeze	98.2	113	15.7	236	39.2

Table 1: Development Accuracy and Compute Metrics by Strategy (Naija)

To start, Table 1 provides a detailed summary of the experiment carried out, including the strategies explored and their accuracies and compute time reduction. We can see that between Baseline and Alternating Freeze, the accuracy stays at 92.8% even though the compute time is saved by 39.2%. Also, Freeze-First-2 and Freeze-First-4 slightly drop the accuracy by 0.1% and 0.5% but saved compute time by 35.8% and 42.5% respectively, indicating a fair enough trade off in terms of limited resources. Figure 5 and 2 give visual descriptions of the freezing accuracies and compute time for each strategy.

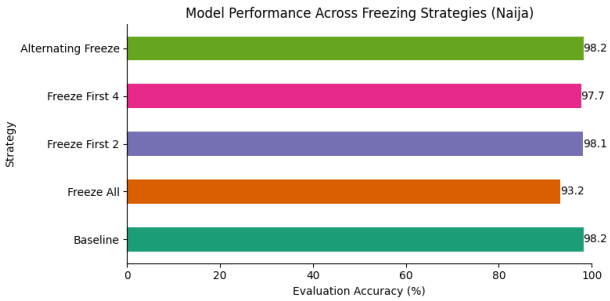

Figure 1: Freezing Accuracies

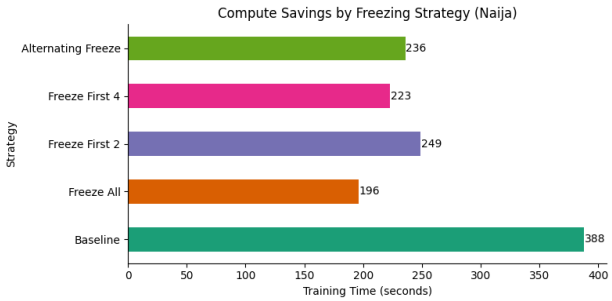

Figure 2: Training Time

Convergence Curves

The convergence curve in 3 shows that all strategies except "Freeze All" quickly approach high accuracy by epoch 5, with the baseline and Alternating Freeze performing nearly identically, from around 97.5% at epoch 1 and 98.2% at epoch 5. Freeze First 2 is slightly lower across all epochs. Freeze All lags behind significantly, highlighting the importance of adapting at least part of the encoder. Overall, freezing the early or alternating layers maintains strong performance while ensuring computational efficiency.

Accuracy vs Parameter Tradeoff

The scatter plot in 4 shows a clear trend of how trainable parameters decrease and how the evaluation accuracy remains high for all the freezing strategies except Freeze All. Freeze

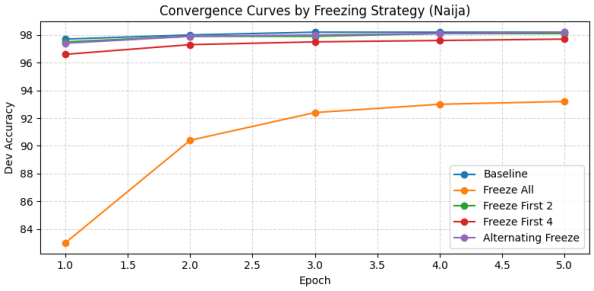

Figure 3: Graduations of Epoch Accuracies

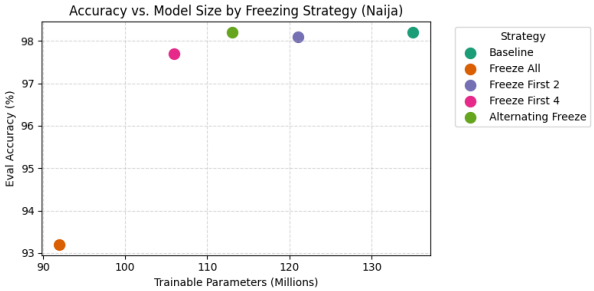

Figure 4: Accuracy vs Parameter Tradeoff

First 2 and Alternating Freeze achieve near-baseline accuracy at 98.1% and 98.2% with fewer parameters of 10.4% and 15.7% respectively, confirming their efficiency. Freeze First 4 remained competitive at 97.7% reducing training parameters by 20.9% while Freeze All, with the smallest trainable size, shows a sharp drop in performance to 93.2%, reinforcing that some encoder layers must adapt for effective PoS tagging.

Results for English Data

The results for the UD English EWT data is given below. The graphs and analysis are very similar tpo that of the UD Naija NSC given in the previpus section.

Strategy	Dev Acc (%)	Params (M)	Param Save (%)	Time (s)	Time Save (%)
Baseline	96.4	134	0.0	532	0.0
Freeze All	91.6	92	31.3	351	34.0
Freeze First 2	96.3	120	10.4	447	16.0
Freeze First 4	96.1	106	20.9	400	24.8
Alternating Freeze	96.2	113	15.7	423	20.5

Table 2: Development Accuracy and Compute Metrics by Strategy (English)

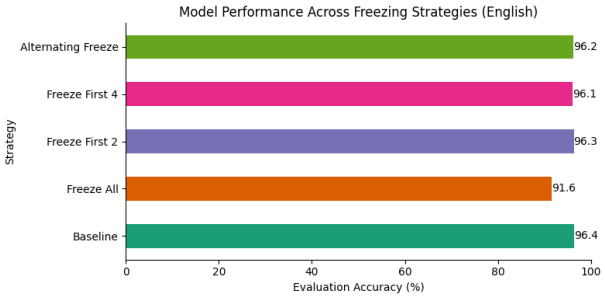

Figure 5: Freezing Accuracies (English)

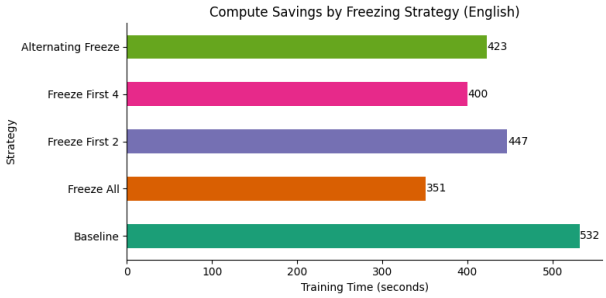

Figure 6: Training Time (English)

Across all strategies, the baseline model delivered the highest development accuracy at 96.4%, but it also required the most resources to train the full 134M parameters. Freezing the first two layers proved highly effective, with nearly identical accuracy (96.3%) and a modest 16% reduction in training time. The alternating freeze strategy also performed well, matching 96.2% accuracy and cutting compute by about 20.5%. Freezing the first four layers resulted in slightly lower accuracy (96.1%) but offered a better trade-off in terms of compute savings of 24.8%. The most drastic reduction came from freezing all encoder layers, which slashed training time by 34% but dropped accuracy to 91.6%. From the convergence curves, we observed that most strategies reached peak performance by epoch 3 - 4, while the freeze-all model lagged throughout. Overall, freezing the lower or alternating layers helped retain accuracy while improving efficiency, making them strong candidates for resource-constrained training.

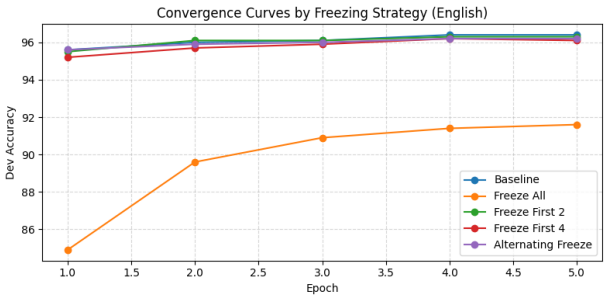

Figure 7: Graduations of Epoch Accuracies (English)

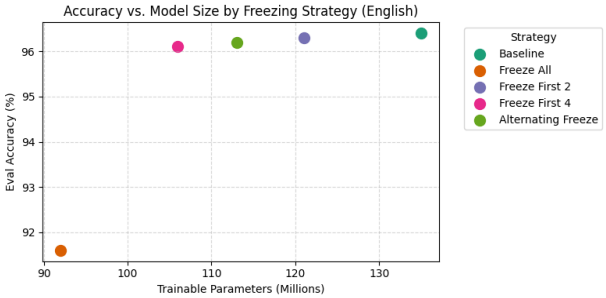

Figure 8: Accuracy vs Parameter Tradeoff (English)

Conclusion and Recommendations

This study has demonstrated that partial layer freezing is a viable and efficient alternative to full fine-tuning for multilingual Part-of-Speech tagging. Using the DistilBERT multilingual model, we explored various freezing strategies on both English EWT and Naija NSC datasets. Results show that while fully fine-tuning all layers yields the highest performance, selectively freezing early encoder layers (particularly the first two) retains nearly all the accuracy while significantly reducing computational cost.

Among the strategies tested, freezing the first two layers consistently delivered the best trade-off, achieving over 96% accuracy on English and 98% on Naija, with more than 10% savings in trainable parameters and up to 16% reduction in training time. Alternating freeze patterns also performed well, offering balanced compute savings with minimal accuracy degradation. In contrast, freezing all encoder layers resulted in notable performance loss, reaffirming the importance of allowing some adaptation in the encoder for token classification tasks.

The convergence plots and efficiency curves further reinforce that partial freezing does not substantially affect learning dynamics when done strategically. This makes it a practical solution for deploying transformer models in low-resource or compute-constrained environments.

We recommend that partial freezing (e.g., freezing the first 2 or 4 layers) be used when compute or memory is limited, as it offers significant savings with minimal accuracy loss and head-only training (freeze all) should be avoided for PoS tagging, as it consistently underperforms even under ideal conditions. Also, consider alternating freezing for a good balance between model adaptability and efficiency.

Future research should investigate combining partial freezing with techniques like layer-wise learning rates, low-rank adaptation (LoRA), or mixed precision training to further enhance performance under constrained settings.

References

[1] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). "BERT: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint https://arxiv.org/abs/1810.04805.

[2] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., & Gelly, S. (2019). "Parameter-efficient transfer learning for NLP." In *Proceedings of the 36th International Conference on Machine Learning (ICML).* arXiv preprint https://arxiv.org/pdf/1902.00751

[3] Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov S., Pyysalo S., Silveira N., Tsarfaty R., Zeman, D. (2025). *Universal Dependencies 2.16*. LINDAT/CLARIAH-CZ digital library. Retrieved from https://universaldependencies.org

[4] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint https://arxiv.org/abs/1910.01108*.

[5] Talukdar, K., Sarma, S. K., & Bhuyan, M. P. (2022). *Parts of Speech (PoS) and Universal Parts of Speech (UPoS) Tagging: A Critical Review with Special Reference to Low Resource Languages*. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 703–713, Goa University, Goa, India. NLP Association of India (NLPAI) https://aclanthology.org/2023.icon-1.70/