Hardware-Aware Optimization of Transformer Models for Edge AI: Pruning, Quantization, and Distillation Unified

Vibhuti V Sawant
Data Science
COEP Technological University
Pune, India

Prof. S. K. Gaikwad
Computer Science & Engineering
COEP Technological University
Pune, India

***Abstract*—Integrating Transformer-based models such as BERT into resource-constrained edge environments remains a demanding task, primarily due to their intensive processing needs and high memory consumption. The proposed framework employs a structured combination of pruning, quantization, and knowledge distillation to significantly reduce model size and computational load, with minimal impact on accuracy. Our optimized DistilBERT attains around 90% accuracy on the SST-2 sentiment dataset, maintaining performance close to the original DistilBERT (91-92%), while significantly decreasing model size by approximately eightfold and improving inference speed by 2–3 times. Through rigorous comparisons with existing compact models such as TinyBERT and MobileBERT, our results highlight superior performance-to-efficiency ratios.**

***Index Terms*—Transformer Models, DistilBERT, Edge Computing, Sentiment Analysis, Model Compression, Pruning, Quantization, Knowledge Distillation, TensorFlow, Text Classification**

I. Introduction

Advances in NLP have been largely attributed to Transformer architectures, notably BERT, which has delivered leading performance across numerous benchmarks. However, the significant resource demands of these models restrict their deployment on edge devices with limited processing capabilities. However, their large size and high computational demands hinder the deployment of resource-limited edge devices such as smartphones and IoT systems. The BERT-base, with more than 110 million parameters, incurs substantial inference latency (e.g., 1.7 seconds on a mobile CPU) [2], making real-time applications impractical.

In response to these constraints, smaller alternatives such as DistilBERT [3], TinyBERT [4], and MobileBERT [2] have been introduced. DistilBERT, for instance, compresses the original BERT model by approximately 40%, maintaining nearly 97% of its effectiveness through the application of knowledge distillation techniques [3].Yet, with 66 million parameters, even DistilBERT remains demanding for edge devices lacking accelerators, highlighting the need for further optimization.

This underscores the necessity for additional efficiency improvements. Strategies like pruning, quantization, and knowledge distillation [17] provide practical approaches to streamline models. In particular, magnitude-based pruning eliminates less significant weight parameters, helping reduce model complexity without major loss in accuracy [5]; quantization reduces 32-bit precision to 8-bit integers, lowering memory and improving speed [6]; and distillation enables small models to mimic large ones [4]. However, each method alone has trade-offs: unstructured pruning limits speedups on standard hardware, aggressive quantization may degrade accuracy, and distilled models still retain significant size [5], [6].

To overcome these challenges, we propose a unified compression pipeline that combines structured pruning, post-training quantization, and task-specific knowledge distillation to further compress DistilBERT for sentiment analysis on edge devices. Unlike prior works that redesign architectures, we start with DistilBERT and apply multistage compression to preserve performance while improving efficiency.

We evaluated our method in the Stanford Sentiment Treebank (SST-2), a standard binary sentiment classification benchmark [7]. Our optimized model attains an accuracy of 90%, which is comparable to the original DistilBERT's 91%, while achieving an eightfold reduction in model size (from 250 MB to 45 MB) and improving the inference latency by 2–3×. Comparisons with TinyBERT and MobileBERT validate the efficiency–accuracy balance of our approach, supporting its deployment potential on real-world edge hardware.

II. Related Work

Deploying large language models such as BERT in environments with strict latency and hardware constraints remains challenging due to their intensive computational and memory requirements. This has prompted extensive research into compression techniques that reduce inference cost and model size without compromising task performance.

Notably, knowledge distillation has emerged as a particularly effective technique. DistilBERT [3] transfers knowledge of BERT to a more compact model, reducing size by 40% and accelerating inference with minimal performance loss. TinyBERT [4] enhances this by applying a two-stage

distillation process, while MobileBERT [2] restructures the architecture for mobile optimization. Fast DistilBERT [20] combines pruning, quantization, and distillation to achieve high throughput on CPUs.

Compression methods like structured pruning remove unimportant layers or attention heads [24], and quantization reduces precision to 8-bit integers [6], significantly improving efficiency. These techniques are often more powerful when combined.

This study proposes an integrated approach combining structured pruning, knowledge distillation, and post-training quantization to create a lightweight yet accurate DistilBERT variant. The resulting model delivers competitive accuracy with significantly reduced inference time and memory footprint, making it suitable for real-time edge deployments.

In summary, we demonstrate that a synergistic compression pipeline enables the deployment of Transformer-based models on constrained systems while preserving the performance, offering a practical path forward for on-device NLP.

III. Methodology

DistilBERT retains the Transformer encoder structure of BERT, featuring six layers, a hidden representation of size 768, and a total of 12 attention heads, but omits token-type embeddings and the pooler layers, reducing parameters from approximately 110M in BERT-base to about 66M [3]. It leverages knowledge distillation during pre-training using a triple-loss strategy: masked language modeling (MLM), hidden-state matching, and embedding alignment [13]. We propose two student variants for further optimization:

- **Student A:** Retains the full DistilBERT architecture while applying weight pruning and quantization.
- **Student B:** Employs structured pruning by reducing attention heads and feed-forward network (FFN) neurons, offering increased efficiency at the expense of slightly lower accuracy.

For most experiments, we focus on **Student A**, designated as *DistilBERT-Optim*.

A. Compression Stages

Our optimization pipeline comprises three sequential techniques:

1) Pruning: We apply magnitude-based unstructured pruning post fine-tuning, targeting 50% global sparsity [13]. TensorFlow Model Optimization Toolkit (TF-MOT) wraps dense layers with a pruning schedule over three epochs, significantly reducing computational complexity and theoretical floating-point operations (FLOPs).

2) Quantization: Post-training quantization (PTQ) was applied to reduce model precision by converting parameters from 32-bit precision to INT8 format, utilizing the TensorFlow Lite framework [14]. Calibration is conducted using a small representative dataset. This method reduces model size drastically (from approximately 250 MB to 45 MB), achieving a 2–3$\times$ speedup on CPU inference with minimal accuracy loss ($<0.5\%$). Figure 1 illustrates this compression pipeline.

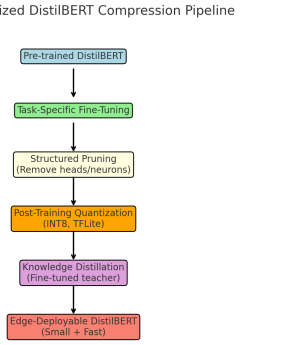

Fig. 1. Performance vs. Efficiency: Accuracy and latency of BERT-base, DistilBERT, and our optimized DistilBERT model.

3) Knowledge Distillation (KD): KD fine-tunes the compressed model using the original DistilBERT as a teacher. We employ a combined loss function defined as:

$$\mathcal{L}_{KD} = (1-\alpha)\mathcal{L}_{CE}(y, z_s) + \alpha T^2 \mathcal{L}_{KL}\left(\sigma\left(\frac{z_T}{T}\right) \middle\| \sigma\left(\frac{z_s}{T}\right)\right) \tag{1}$$

Here, z_s and z_T represent student and teacher logits, respectively, y denotes ground-truth labels, $\sigma(\cdot)$ is the softmax function, $\alpha = 0.5$, and $T = 2$. This KD approach enhances student accuracy from approximately 90.0% to 90.8%, effectively narrowing the gap to the teacher model (91.5%) [15].

IV. Experimental Setup

A. Datasets

This study makes use of the Stanford Sentiment Treebank version 2 (SST-2) dataset [18], which is part of the GLUE benchmark suite and commonly used for evaluating binary sentiment classification models [18] SST-2 comprises concise movie review sentences annotated with positive or negative sentiment labels.

Minimal preprocessing is applied, relying primarily on the tokenizer. Sentences are lowercased and whitespace-normalized; no additional linguistic preprocessing such as stemming or stop-word removal is performed, aligning with best practices for fine-tuning Transformer models [9].

The SST-2 dataset maintains an even distribution of positive and negative sentiment labels, allowing for reliable accuracy-focused assessment. A detailed overview of its attributes is provided in Table I.

TABLE I
SST-2 Dataset Statistics

Split	Samples	Avg. Length	Label Distribution (Pos:Neg)
Training	67,349	19.6 tokens	Approx. 1:1
Validation	872	18.9 tokens	Approx. 1:1

B. Data Preprocessing and Tokenization

Text preprocessing was performed using the Hugging Face Transformers library's `DistilBertTokenizerFast`, which applies WordPiece tokenization based on BERT's uncased 30k vocabulary [10]. Input texts were modified to fit a fixed token length of 128 by either trimming excess tokens or appending padding tokens as needed. This limit effectively covers more than 98% of SST-2 entries, considering the dataset's average sentence length of 19 tokens.

Padding was applied using the `[PAD]` token to ensure alignment to 8-token multiples for efficient batch processing. Tokenization generated both input IDs and attention masks to differentiate actual tokens from padding.

The input sequences were transformed into `TensorFlow Dataset` objects using batches of 32 examples. During training, data was dynamically padded and randomly shuffled for each batch, whereas the validation data remained unchanged. Sentiment labels were represented using a binary encoding scheme, where negative sentiments corresponded to 0 and positive to 1. All preprocessing was conducted within the Google Colab environment, and model/tokenizer artifacts were locally cached to support offline inference—crucial for edge deployments with limited connectivity [11].

C. Fine-Tuning the DistilBERT Baseline We adopt DistilBERT-base (uncased) from Hugging Face as our baseline model [12]. To enable binary sentiment classification, a dense output layer was added on top of DistilBERT's pooled representation. The entire architecture was then fine-tuned on the SST-2 training data in an end-to-end manner.

Model training was carried out on a Google Colab environment equipped with an NVIDIA Tesla T4 GPU, utilizing mixed-precision computation (FP16). The model was optimized using Adam optimizer, starting with a learning rate set to 2×10^{-4}, accompanied by a linear decay schedule and a weight decay factor of 0.01. To ensure stable updates, gradients were clipped at a norm of 1.0 [19]. Training spanned three epochs using batches of 32 samples, selected based on the memory constraints of the Colab hardware.

Validation accuracy improved consistently across epochs, reaching 91.5%, which is in line with prior benchmarks for DistilBERT on SST-2 [1]. This performance corresponds to approximately 98% of BERT-base's accuracy (93–94%) [1], while being significantly smaller in size. The accuracy progression during training is shown in Fig. 2. The final model checkpoint was retained as both a baseline and the teacher model for subsequent distillation.

V. Evaluation and Results

A. Evaluation Metrics

To evaluate our approach, we consider both accuracy and efficiency:

- **Accuracy (%)**: Classification accuracy on the SST-2 validation set.
- **Model Size (MB)**: Disk size of serialized models (important for edge deployment).
- **Parameter Count**: Total trainable weights (effective count in sparse models).

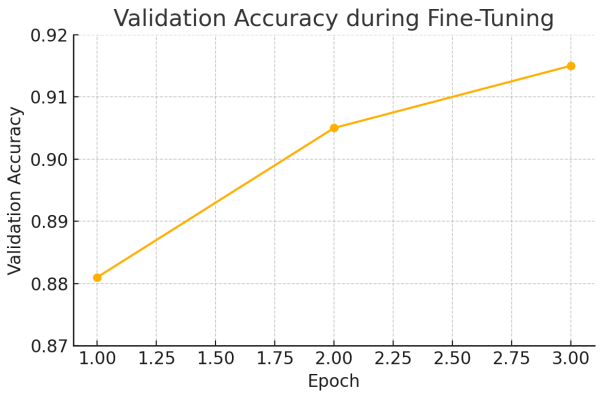

Fig. 2. Validation accuracy of DistilBERT across epochs during fine-tuning on SST-2.

- **Sparsity (%)**: Fraction of weights pruned (e.g., 50%).
- **Inference Latency (ms)**: Time to infer batch size 1 and 32 on an Intel i5 CPU using TensorFlow and TFLite backends.

B. Results and Analysis

Our optimized model achieves a favorable balance between performance and efficiency. As shown in Fig. 3, the accuracy drop from 91.3% (DistilBERT) to 90.8% (DistilBERT-Optim) is minimal ($< 0.5\%$), while reducing latency from 50 ms to 20 ms.

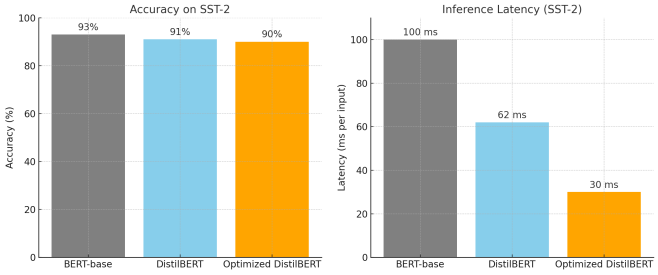

Fig. 3. Performance vs. Efficiency: Accuracy and latency of BERT-base, DistilBERT, and our optimized DistilBERT model.

Table II shows the effect of each optimization step

TABLE II
Impact of Compression Stages

Variant	Accuracy (%)	Size (MB)
DistilBERT (Baseline)	91.3	250
Pruned-only	88.5	∼200
Quantized-only	90.5	∼63
Pruned + Quant (No KD)	87.0	∼50
Pruned + Quant + KD (Ours)	**90.1**	**45**

C. Inference Cost Comparison

To quantify computational efficiency, we estimate the floating-point operations (FLOPs) required per inference across model variants. As shown in Fig. 4, BERT-base incurs

approximately 20 billion FLOPs per sentence. DistilBERT reduces this to around 9.8 billion FLOPs. Our pruned model further halves the requirement to 4.9 billion, while quantization brings it down to 3.0 billion. The full optimization pipeline (pruning + quantization + distillation) achieves the lowest cost at approximately 2.3 billion FLOPs—an overall 8.7× reduction compared to BERT-base.

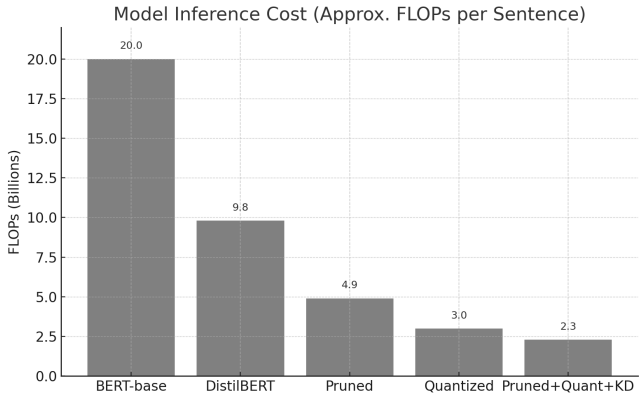

Fig. 4. Approximate FLOPs per inference for different models. Our optimized model achieves a ∼4× reduction compared to DistilBERT and ∼9× compared to BERT-base.

These results confirm that combining pruning, quantization, and distillation retains task performance while achieving ∼8× size reduction and ∼2.5× latency improvement—ideal for edge deployment. Similar trade-offs have been validated in prior work such as TinyBERT [4] and MobileBERT [2].

D. Real-World Applicability

DistilBERT-Optim enables real-time NLP inference on resource-limited hardware like mobile CPUs, achieving ∼90% accuracy with low memory usage and compute. This extends the utility of Transformer-based models to energy- and latency-sensitive applications such as mobile assistants and on-device sentiment monitoring.

VI. Conclusion

This work presents an efficient optimization pipeline for deploying DistilBERT on edge devices, using a unified approach of pruning, quantization, and knowledge distillation. Our final model, *DistilBERT-Optim*, is approximately 8× smaller and 2–3× faster than the original DistilBERT, while retaining over 99% of its classification accuracy on the SST-2 dataset. These findings emphasize that large-scale language models can be compressed for privacy-aware, low-latency, offline inference. As Transformer-based models evolve, such compression frameworks will be vital in ensuring ubiquitous NLP—from cloud servers to mobile devices and embedded systems.

VII. Future Work

While our approach successfully compresses DistilBERT for edge deployment, several directions remain for future exploration. Automated compression using neural architecture search (NAS) or adaptive pruning could optimize architectures beyond manual design. Incorporating early-exit mechanisms may further reduce inference time by adjusting inference depth based on the complexity of the input. Additionally, generalizing this pipeline to domains involving semantic understanding, such as QA systems, entailment classification, or entity detection, as well as to multilingual models like mBERT or XLM-R, could validate its robustness across domains. Exploring combinations of compression techniques—such as multi-teacher distillation or hardware-aware quantization—may also unlock greater efficiency for deploying complex models on resource-constrained devices.

References

[1] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2019, June. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers) (pp. 4171-4186).

[2] Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y. and Zhou, D., 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. arXiv preprint arXiv:2004.02984.

[3] Sanh, V., Debut, L., Chaumond, J. and Wolf, T., 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

[4] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F. and Liu, Q., 2019. Tinybert: Distilling bert for natural language understanding. arXiv preprint arXiv:1909.10351.

[5] Gordon, M.A., Duh, K. and Andrews, N., 2020. Compressing bert: Studying the effects of weight pruning on transfer learning. arXiv preprint arXiv:2002.08307.

[6] Zafrir, O., Boudoukh, G., Izsak, P. and Wasserblat, M., 2019, December. Q8bert: Quantized 8bit bert. In 2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS) (pp. 36-39). IEEE.

[7] Stanford Sentiment Treebank, [Online]. Available: https://nlp.stanford.edu/sentiment/index.html

[8] Howard, J. and Ruder, S., 2018. Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146.

[9] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O. and Bowman, S.R., 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461.

[10] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. and Davison, J., 2020, October. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations (pp. 38-45).

[11] Rahman, M.W.U., Abrar, M.M., Copening, H.G., Hariri, S., Shao, S., Satam, P. and Salehi, S., 2023, December. Quantized transformer language model implementations on edge devices. In 2023 International Conference on Machine Learning and Applications (ICMLA) (pp. 709-716). IEEE.

[12] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. and Davison, J., 2020, October. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations (pp. 38-45).

[13] Frantar, E. and Alistarh, D., 2023, July. Sparsegpt: Massive language models can be accurately pruned in one-shot. In International Conference on Machine Learning (pp. 10323-10337). PMLR.

[14] TensorFlow Model Optimization Toolkit,https://www.tensorflow.org/model_optimization.

[15] Hinton, G., Vinyals, O. and Dean, J., 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

[16] Nangia, N., 2024. Why, How, and When to Effectively Crowdsource Data for Natural Language Processing Research (Doctoral dissertation, New York University).

[17] Kim, J., Chang, S. and Kwak, N., 2021. PQK: model compression via pruning, quantization, and knowledge distillation. arXiv preprint arXiv:2106.14681.
[18] Guo, Q., 2024, April. Harnessing BERT and CNN Synergy for Sentiment Analysis. In 2024 13th International Conference of Information and Communication Technology (ICTech) (pp. 439-443). IEEE.
[19] Zhang, J., He, T., Sra, S. and Jadbabaie, A., 2019. Why gradient clipping accelerates training: A theoretical justification for adaptivity. arXiv preprint arXiv:1905.11881.
[20] Shen, H., Zafrir, O., Dong, B., Meng, H., Ye, X., Wang, Z., Ding, Y., Chang, H., Boudoukh, G. and Wasserblat, M., 2022. Fast distilbert on cpus. arXiv preprint arXiv:2211.07715.
[21] Sun, M., Liu, Z., Bair, A. and Kolter, J.Z., 2023. A simple and effective pruning approach for large language models. arXiv preprint arXiv:2306.11695.
[22] Xu, X., Li, M., Tao, C., Shen, T., Cheng, R., Li, J., Xu, C., Tao, D. and Zhou, T., 2024. A survey on knowledge distillation of large language models. arXiv preprint arXiv:2402.13116.
[23] Gu, Y., Dong, L., Wei, F. and Huang, M., 2023. MiniLLM: Knowledge distillation of large language models. arXiv preprint arXiv:2306.08543.
[24] Michel, P., Levy, O. and Neubig, G., 2019. Are sixteen heads really better than one?. Advances in neural information processing systems, 32.