Towards Building Efficient Sentence BERT Models using Layer Pruning

Anushka Shelke[1,3], **Riya Savant**[1,3], **Raviraj Joshi**[2,3]
[1]MKSSS Cummins College of Engineering for Women, Pune
[2]Indian Institute of Technology Madras
[3]L3Cube Labs, Pune
{anushkashelke020, riya.savant, ravirajoshi }@gmail.com

Abstract

This study examines the effectiveness of layer pruning in creating efficient Sentence BERT (SBERT) models. Our goal is to create smaller sentence embedding models that reduce complexity while maintaining strong embedding similarity. We assess BERT models like Muril and MahaBERT-v2 before and after pruning, comparing them with smaller, scratch-trained models like MahaBERT-Small and MahaBERT-Smaller. Through a two-phase SBERT fine-tuning process involving Natural Language Inference (NLI) and Semantic Textual Similarity (STS), we evaluate the impact of layer reduction on embedding quality. Our findings show that pruned models, despite fewer layers, perform competitively with fully layered versions. Moreover, pruned models consistently outperform similarly sized, scratch-trained models, establishing layer pruning as an effective strategy for creating smaller, efficient embedding models. These results highlight layer pruning as a practical approach for reducing computational demand while preserving high-quality embeddings, making SBERT models more accessible for languages with limited technological resources.

1 Introduction

Language models have evolved significantly in recent years. Although RNNs were once popular, they lack context embedding. Transformers (Vaswani et al., 2023) have emerged as superior, offering parallel processing for faster sequence handling and greater memory efficiency by utilizing position embeddings. Notably, BERT (Devlin et al., 2018), a leading language model, adopts the Transformer architecture, significantly improving performance across a range of NLP tasks by capturing deep contextual relationships within text.

BERT's architecture is built upon a multi-layer bidirectional Transformer encoder, drawing from the foundational framework of transformers (Vaswani et al., 2023).$BERT_{BASE}$ (Devlin et al., 2018) is endowed with 110 million parameters, whereas $BERT_{LARGE}$ boasts 340 million parameters. The deployment of BERT models remains challenging in resource-constrained environments typical of many low-resource languages due to their substantial computational demands.

While BERT excels at capturing contextualized word embeddings, it doesn't directly provide sentence-level representations. SBERT (Reimers and Gurevych, 2019) addresses this limitation by modifying BERT's architecture to efficiently generate sentence embeddings. SBERT accomplishes this through the use of siamese and triplet network structures.The modification introduced by SBERT makes the BERT model more complex by extending its capabilities beyond word-level embeddings to include sentence-level representations. This added complexity enables BERT to capture higher-level semantic information and relationships between entire sentences, enhancing its utility in a wider range of natural language processing tasks.

These fine-tuned BERT models, with their large number of parameters, present challenges for low-capability devices or applications with strict latency requirements due to their resource-intensive nature. Various model compression techniques, including pruning, quantization, knowledge distillation, and architectural modifications, have been employed on BERT (Ganesh et al., 2021) to decrease the model size and computational demands, thereby increasing computation latency.

Building on the efforts to address the challenges posed by resource-intensive BERT models, our research delves into reducing the complexity of SBERT models without compromising performance. Layer pruning, which involves selectively removing less critical parts of the neural network, offers a promising solution for enhancing the efficiency of SBERT models. This is especially important for processing languages within environments

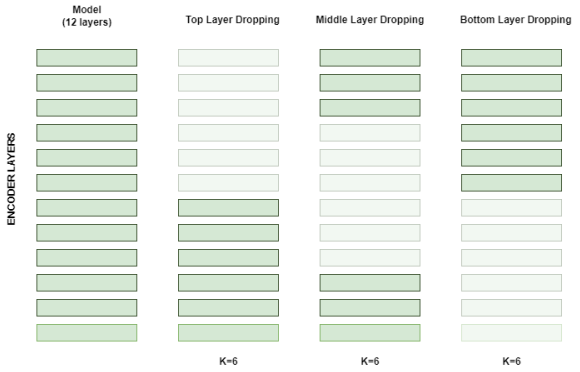

Figure 1: Layer Pruning Strategies.

constrained by limited computing infrastructure.

Model pruning, specifically layer pruning, seeks to address the inefficiencies related to the size and complexity of models like BERT, SBERT. The objective is to reduce the model's size and computational demands while maintaining or enhancing its performance. Techniques vary from removing individual neurons to whole layers. In the context of transformer-based models, a study (Fan et al., 2019) demonstrated that strategic layer removal could reduce model size substantially with minimal impact on performance.

In our research, we delve into recent developments in adapting Sentence-BERT (SBERT) models for low-resource languages, focusing particularly on Marathi and Hindi. The L3Cube-MahaSBERT and HindSBERT (Joshi et al., 2022) models were established as benchmarks for generating high-quality sentence embeddings in Marathi and Hindi, respectively. These specialized models are highlighted for their effectiveness in processing these low-resource languages. These models have been rigorously trained and evaluated across various NLP tasks, including text classification and semantic similarity.

Our research aims to extend these foundational models by applying layer-pruning techniques to enhance their efficiency without compromising the quality of the embeddings. By integrating layer pruning, we seek to reduce the computational demand and improve the operational feasibility of deploying SBERT models in real-world applications, making advanced NLP tools more accessible for languages that traditionally have fewer technological resources.

- A research (Sajjad et al., 2022) has showcased a range of layer pruning strategies, underscoring their effectiveness. These techniques maintain an impressive 98% of the original performance even after removing 40% of the layers from BERT, RoBERTa, and XLNet models.

- Expanding upon these findings, we applied several layer pruning methods such as top-layer pruning, middle-layer pruning, and bottom-layer pruning to SBERT models, as illustrated in the accompanying figure 1. In this context, the parameter "k" represents the number of layers removed from the original model.

- After evaluating all three approaches, we discovered that top-layer pruning yielded the best performance. Therefore, we chose top-layer pruning for our subsequent experiments. To further test the performance of these pruned models, we fine-tuned them using NLI+STS training.

- We compare 2-layer and 6-layer models created through layer pruning of MahaBERT-v2 with similar-sized models trained from scratch, such as MahaBERT-Small and MahaBERT-Smaller. Our observations show that the pruned models consistently outperform the scratch-trained models. Therefore, we recommend layer pruning followed by SBERT-like fine-tuning to create smaller embedding models, rather than training smaller models from scratch and then applying SBERT-like fine-tuning, which is highly computationally intensive.

- Remarkably, these fine-tuned pruned models demonstrate competitive performance compared to larger models, despite being 50% to 80% smaller in size.

2 Related Work

This section discusses the progression of transformer-based models, with a specific focus on their optimization for enhanced efficiency and application in resource-constrained environments.

Introduced by (Devlin et al., 2019) BERT revolutionized NLP tasks by employing a bidirectional training of Transformer, a novel architecture that was originally used in the paper (Vaswani et al., 2023) thereby encapsulating a deeper contextual understanding. The paper (Reimers and Gurevych,

2019) introduces Sentence-BERT (SBERT), a modification of the original BERT model that uses Siamese and triplet network structures to efficiently generate sentence embeddings for enhanced performance in semantic similarity tasks.

(Zhu and Gupta, 2017) evaluates the impact of different pruning techniques on neural network compression and performance across various models and tasks. As discussed in their (Fan et al., 2019), it has been shown that carefully targeted removal of layers can significantly decrease the size of a model while having only a minimal effect on its performance. Furthermore, the study by (Michel et al., 2019), titled "Are Sixteen Heads Really Better than One?" shows that many attention heads in transformers can be pruned without significant degradation in capabilities, highlighting the redundancy in these models.

We explore research aimed at enhancing the efficiency of transformer models, particularly through model compression techniques. Key studies in this area include (Hubara et al., 2016) and (Jiao et al., 2020), which provide valuable insights into designing more efficient models without significant loss in performance. The main goal of TinyBERT is to distill the knowledge from a large pre-trained language model, such as BERT, into a smaller model, while maintaining performance.

Additionally, we delve into the literature on layer pruning techniques, which specifically address methods for optimizing neural network architectures by identifying and removing redundant or less important layers. In this domain, valuable strategies have been employed for reducing the computational burden of neural network models through systematic layer pruning approaches (Liu et al., 2017). An iterative algorithm (Pietron and Wielgosz, 2020) is introduced for layer pruning, reducing storage demands in pre-trained neural networks. It selects layers based on complexity and sensitivity, applying reverse pruning if accuracy drops.

Layer pruning reduces resource usage in CNNs by eliminating entire layers based on their importance estimated through PLS projection (Jordao et al., 2020). It can be followed by filter-oriented pruning for additional compression. Structured pruning (He and Xiao, 2024) encompasses a range of techniques such as filter ranking methods, dynamic execution, the lottery ticket hypothesis, etc. Layer-wise pruning ratios extend traditional weight pruning strategies by focusing on determining the optimal pruning rate for each layer.

Another method for layer-wise pruning based on feature representations (Chen and Zhao, 2019) is introduced. Unlike conventional methods that prune based on weight information, this approach identifies redundant parameters by examining the features learned in convolutional layers, operating at a layer level. A novel approach called layer-compensated pruning (Chin et al., 2018) incorporates meta-learning to address both how many filters to prune per layer and which filters to prune. Tests on ResNet and MobileNetV2 networks across multiple datasets validate the algorithm's effectiveness.

3 Methodologies

SBERT models are known for their complexity and large size. Fig. 2 depicts the process of training a smaller SBERT (Sentence-BERT) model using a technique known as layer pruning. Starting with the original SBERT base model, which consists of multiple layers, the process involves systematically removing certain layers to create a pruned version of the model. This layer-wise pruning aims to reduce the model's complexity without significantly compromising its performance.

Our initial experiments focused on identifying the most effective layer-pruning strategy to optimize the model's performance. We explored several pruning methods, including top-layer pruning, middle-layer pruning, and bottom-layer pruning as shown in 1, to evaluate their impact on model's efficiency and accuracy. Each strategy was tested by removing a specified number of layers, denoted by the parameter "k", from different positions in the model. This approach allowed us to systematically assess how the removal of layers affected the overall performance and computational requirements.

The pruned model is then fine-tuned through two specialized training phases: Natural Language Inference (NLI) training and Semantic Textual Similarity (STS) training. NLI training improves the model's ability to understand logical relationships between sentence pairs, categorizing them as entailment, contradiction, or neutral, whereas STS training focuses on assigning similarity scores to sentence pairs, enhancing the model's ability to gauge semantic closeness. By integrating NLI pre-training and STS fine-tuning, a robust training framework is established for SBERT models.

Following the fine-tuning, the pruned model

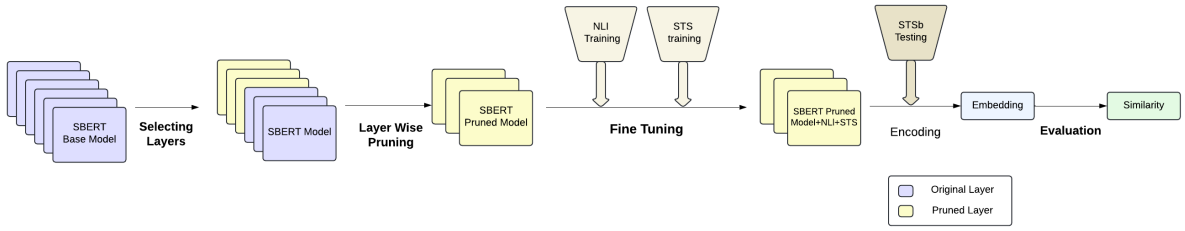

Figure 2: Layer Pruning on SBERT model

Training Methods	Top-layers pruning(1-6)	Middle-layers pruning(4-9)	Bottom-layers pruning(7-12)
NLI	0.7098	0.6912	0.6954

Table 1: Comparison of embedding similarity scores for various layer pruning strategies: Top, Middle, and Bottom layers during NLI training.

which is integrated with NLI and STS training is tested for its performance on the Semantic Textual Similarity benchmarks (STSb testing) dataset. This phase evaluates how effectively the model calculates the similarity between sentences. The final steps involve encoding these sentences into embeddings and evaluating their similarity and assessing the pruned model's accuracy and efficiency. Thus Fig.2 depicts a clear pathway from model complexity reduction through pruning to performance evaluation via embedding and similarity assessments.

Dataset

3.0.1 IndicXNLI [1]

IndicXNLI5 comprises data from the English XNLI dataset that has been translated into eleven Indic languages including Marathi.(Aggarwal et al., 2022) This includes translation of the training (392,702 entries), validation (2,490 entries), and evaluation sets (5,010 entries) from English into each of the eleven languages. From the IndicXNLI dataset, the training samples specific to each language are used to train the MahaSBERT models.

3.0.2 STS benchmark(STSb) [2]

It comprises data from the English XNLI dataset that has been translated into eleven Indic languages including Marathi. This includes translation of the training (392,702 entries), validation (2,490 entries), and evaluation sets (5,010 entries) from English into each of the eleven languages. From the IndicXNLI dataset, the training samples specific to each language are used to train the MahaSBERT models. It has been made publicly accessible.[3]

In our experiments, we specifically utilized the translated Marathi dataset to fine-tune the pruned SBERT models, ensuring the models were optimized for the Marathi language. This approach allowed us to directly target language-specific nuances and enhance the model's performance on tasks relevant to Marathi.

3.1 EXPERIMENT

Referring to the procedures outlined in Fig.2 our experiment evaluates the performance of several SBERT models Muril, MahaBert v2, MahaBert Small, and MahaBert Smaller both before and after the application of layer pruning.

3.1.1 Best Layering Strategy Selection

To identify the most effective pruning strategy, we systematically evaluated the performance of each pruned model configuration using multiple criteria, including accuracy, model size, and computational efficiency. By experimenting with various layer combinations such as the first 6 layers, the middle 6 layers, and the bottom 6 layers we aimed to balance the trade-offs between reducing model complexity and preserving performance. Each combination was assessed on the 12-layer MahaBert v2 model using a validation set, focusing on its impact on natural language understanding tasks in Marathi through NLI training. The top-layers pruning strategy yielded the highest accuracy scores compared to other configurations. Based on these results,

[1] https://github.com/divyanshuaggarwal/IndicXNLI

[2] https://huggingface.co/datasets/stsb_multi_mt

[3] https://github.com/l3cube-pune/MarathiNLP

Training Language/Model	Original layers	No. of layers after pruning	NLI	NLI+STS
MahaBert-small	6	2	0.6659	0.7362
MahaBert-smaller	2	2	0.6563	0.7308
MahaBert-v2	12	2	0.6760	**0.7447**
Muril	12	2	0.6880	0.7284
MahaBert-small	6	6	0.6693	0.7422
MahaBert-v2	12	6	0.7098	**0.7878**
Muril	12	6	0.6849	0.7742
MahaBert-v2	12	12	0.7720	**0.8320**
Muril	12	12	0.7488	0.8165

Table 2: Embedding similarity scores from two-step NLI+STS Training on SBert Models

we selected the top-layer pruning strategy for our further experiments.

3.1.2 Layer Pruning

Layer pruning was conducted on the base models Muril, MahaBERT, MahaBERT-Small, and MahaBERT-Smaller to explore various layer combinations and analyze the resulting changes in model performance and complexity. For models like Muril and MahaBERT consisting of 12 layers, we considered different layer subset combinations such as 2, 6 and 12 layers.

3.1.3 Fine Tuning

After obtaining the pruned SBERT model we fine-tuned the model in two phases of training. We first performed NLI training on the model using the Marathi dataset of IndicXNLI and then used the translated STSb train dataset as the second step for training. Thus the pruned model was trained using two steps to obtain the fine-tuned model targeting the Marathi language.

3.1.4 Evaluation

For evaluating the pruned SBERT model which has undergone NLI+STS training we find the embedding similarity scores using Translated STSb Marathi test dataset. On the obtained embeddings we apply the KNN Classifier algorithm to obtain Similarity scores. For classification, we use the IndicNLP News Article Classification dataset targeting the Marathi language.

4 Results

Following layer pruning and two-step NLI+STS training on SBert models, Table 2 shows the embedding similarity scores obtained from various models. The outcomes display similarity scores between 0.72 and 0.83 for different combinations of layers. Notably, the pruned MahaBert-Small model (2 layers) achieved performance comparable to the base model (6 layers), indicating that layer reduction does not necessarily compromise embedding quality. Additionally, the application of NLI+STS fine-tuning greatly enhances similarity scores for all models.

Our experiments demonstrated that models with fewer layers, achieved through layer pruning, can still yield competitive embedding similarity scores. For instance, models with just 2 or 6 layers performed comparably to their fully layered counterparts after undergoing two-phase fine-tuning (NLI followed by STS training). This indicates that there is no necessity to train large, computationally intensive models when pruned models can offer similar performance. These findings suggest that layer pruning is an effective technique for enhancing model efficiency without compromising the quality of embeddings. This approach helps achieve better accuracy while leveraging the advantages of model pruning.

5 Conclusion

Our primary aim was to identify layering configurations that reduce complexity while maintaining strong performance in terms of embedding similarity scores. Our experiments demonstrated that pruned SBERT models, with fewer layers, can achieve performance comparable to their fully layered counterparts. Thus with comparative scores obtained from pruned models we can conclude that pruned models have outperform models i.e. MahaBERT-Small and MahaBERT-Smaller, which are built from scratch. Therefore, instead of developing new models from the ground up, it is more

effective to start with a larger model and apply pruning techniques.

By reducing computational demand and maintaining high-quality embeddings, our approach makes advanced NLP tools more accessible and operationally feasible, particularly for languages with fewer technological resources.

In the long term, this work highlights the potential for layer-pruned SBERT models to be adapted for diverse NLP tasks, such as text classification, question answering and even more complex tasks such as Information Retrieval with Retrieval-Augmented Generation(RAG). By integrating RAG,the pruned models are not only more computational efficient but also capable of retrieving relevant information dynamically. This combined approach of pruning and augmentation extends the model's applicability across a broad range of tasks, making advanced NLP capabilities more accessible and adaptable to real-world, resource constrainted applications.

Acknowledgements

We gratefully acknowledge the L3Cube Mentorship Program, Pune for providing the platform for this research. We express our sincere thanks to our mentors for their guidance and encouragement throughout the project.

References

Divyanshu Aggarwal, Vivek Gupta, and Anoop Kunchukuttan. 2022. Indicxnli: Evaluating multilingual inference for indian languages. *Preprint*, arXiv:2204.08776.

Shi Chen and Qi Zhao. 2019. Shallowing deep networks: Layer-wise pruning based on feature representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):3048–3056.

Ting-Wu Chin, Cha Zhang, and Diana Marculescu. 2018. Layer-compensated pruning for resource-constrained convolutional neural networks. *Preprint*, arXiv:1810.00518.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. *Preprint*, arXiv:1909.11556.

Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Hassan Sajjad, Preslav Nakov, Deming Chen, and Marianne Winslett. 2021. Compressing large-scale transformer-based models: A case study on bert. *Transactions of the Association for Computational Linguistics*, 9:1061–1080.

Yang He and Lingao Xiao. 2024. Structured pruning for deep convolutional neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2900–2919.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Quantized neural networks: Training neural networks with low precision weights and activations. *Preprint*, arXiv:1609.07061.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. *Preprint*, arXiv:1909.10351.

Artur Jordao, Maiko Lie, and William Robson Schwartz. 2020. Discriminative layer pruning for convolutional neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 14(4):828–837.

Ananya Joshi, Aditi Kajale, Janhavi Gadre, Samruddhi Deode, and Raviraj Joshi. 2022. L3cube-mahasbert and hindsbert: Sentence bert models and benchmarking bert sentence representations for hindi and marathi. *Preprint*, arXiv:2211.11187.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. 2017. Learning efficient convolutional networks through network slimming. *Preprint*, arXiv:1708.06519.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Preprint*, arXiv:1905.10650.

Marcin Pietron and Maciej Wielgosz. 2020. Retrain or not retrain? – efficient pruning methods of deep cnn networks. *Preprint*, arXiv:2002.07051.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2022. On the effect of dropping layers of pre-trained transformer models. *Preprint*, arXiv:2004.03844.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Michael Zhu and Suyog Gupta. 2017. To prune, or not to prune: exploring the efficacy of pruning for model compression. *Preprint*, arXiv:1710.01878.