CSE598: Data Intensive Systems for Machine Learning

An In-depth Assessment of Compression Methods on LLMs

Group - 22
Guruanjan Jaiprakash Singh (gsingh96@asu,edu)
Sarvesh Narendra Kapse (skapse@asu.edu)
Shubham Shashikant Rane (srane5@asu.edu)

Abstract: This paper evaluates compression techniques like quantization, pruning, and knowledge distillation on large language models (LLMs) such as BERT, ALBERT, and RoBERTa. Experiments on the Stanford Sentiment Treebank dataset[7] analyze trade-offs between accuracy, model size reduction, and inference speedup. Quantization and pruning significantly reduced model size while maintaining reasonable accuracy, with ALBERT exhibiting the best balance. Knowledge distillation retained high accuracy with moderate compression. Results highlight considering multiple factors like accuracy, size, and speed when selecting compression techniques for resource-constrained LLM deployment. The study provides insights into optimizing LLM performance and efficiency via model compression.

1) Introduction

We explore language model optimization in our project for the Data Intensive Machine Learning course with the goal of utilizing its revolutionary potential in the field of natural language processing. This project is in perfect harmony with the course's emphasis on using data-driven strategies to solve challenging machine learning problems. By carefully analyzing different improvement approaches, we want to shed light on the nuances of language model effectiveness and efficiency. In order to achieve a delicate balance between reducing the size of the model and maintaining crucial linguistic information, we will investigate various compression strategies, including quantization, pruning, knowledge distillation, and low-rank approximation. This project not only contributes to our understanding of cutting-edge AI advancements but also provides practical insights that are highly relevant to the principles and methodologies taught in the Data Intensive Machine Learning course.

This problem is interesting because it addresses a pressing need in the field of natural language processing, where the deployment of state-of-the-art language models is often limited by computational constraints and environmental factors. By investigating compression strategies, we not only contribute to the theoretical understanding of language model optimization but also pave the way for practical applications that can leverage the power of these models more efficiently. Furthermore, successful compression techniques could democratize access to advanced language models, enabling their use in a broader range of applications and devices, fostering innovation and driving progress in the field. The exploration of this problem holds the promise of pushing the boundaries of what is possible with language models, making them more accessible and environmentally sustainable while preserving their remarkable linguistic capabilities.

2) Related Work

To perform a literature survey for this paper, we include an examination of various compression algorithms for language models, such as quantization, pruning, knowledge distillation, and low-rank approximation. These methods offer insights into reducing the size of language models while maintaining their performance. Additionally, our review discusses the importance of compression for large language models, emphasizing the need for efficient algorithms to address computational and environmental challenges. Recent trends in language model compression, including the exploration of sparse models and new compression techniques, are also highlighted to contextualize the research. Our effort on language model compression greatly benefits from this paper, "Learning, Energy Efficient Machine,"[2] which was presented at the Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing in 2019. This work investigates methods and strategies to enhance the energy efficiency of machine learning models, which is consistent with our objective of creating computationally efficient compression algorithms for language models. We can learn how to create compression algorithms that minimize language model size while optimizing energy use by looking at energy-efficient machine learning techniques.

The next study by Zafrir et al., "Prune once for all: Sparse pre-trained language models," [3] offers a novel method for achieving sparsity in pre-trained language models by pruning. This research tackles the problem of shrinking language models without sacrificing performance, which is directly relevant to our work. By eliminating the need for repetitive pruning procedures, the authors suggest a way to drastically cut the amount of computing power needed for inference by pruning pre-trained language models in a single step.

Another relevant study is by Sathyendra et al., titled "Extreme model compression for on-device natural language understanding" [4]. This paper explores compression techniques specifically tailored for deploying natural language understanding models on resource-constrained devices, such as mobile phones and IoT devices. The authors propose a combination of quantization, pruning, and knowledge distillation to achieve extreme compression ratios while maintaining reasonable accuracy. They introduce a novel quantization-aware knowledge distillation method that enables effective knowledge transfer from a larger teacher model to a quantized student model. The results demonstrate significant reductions in model size, up to 40 times smaller than the original model, while retaining performance suitable for on-device deployment.

Choudhary et al. present a comprehensive survey on model compression and acceleration techniques in their paper "A comprehensive survey on model compression and acceleration" [5]. This survey provides a thorough overview of various compression methods, including pruning, quantization, low-rank approximation, knowledge distillation, and neural architecture search. The authors discuss the theoretical foundations, implementation details, and trade-offs associated with each technique, making it a valuable resource for understanding the state-of-the-art in model compression. Additionally, they highlight the challenges and future directions in this field, such as the need for hardware-aware compression algorithms and the exploration of compression techniques for emerging model architectures like transformers and graph neural networks.

3) Dataset

For our project, we have employed the Stanford Sentiment Treebank (SST-2) dataset[7], a widely used corpus for binary sentiment classification tasks. The SST-2 dataset is derived from the Stanford Sentiment Treebank, a collection of movie reviews annotated with fine-grained sentiment labels. However, SST-2 simplifies the task by providing binary labels, classifying each sentence as either positive or negative sentiment. The SST-2 dataset comprises 67,349 sentences extracted from 11,855 movie reviews, offering a diverse range of sentiment examples. This diversity is crucial for training and evaluating language models, ensuring their robustness and generalization capabilities across various domains and linguistic styles. The dataset is conveniently split into three subsets: a training set with 6,920 sentences, a validation set with 872 sentences, and a test set with 1,821 sentences. This predefined split allows for systematic model training, validation, and testing, facilitating reliable performance evaluation and comparisons.

The SST-2 dataset presents several advantages for our compression experiments on large language models (LLMs). First, by reducing sentiment analysis to a binary classification problem, the dataset simplifies the training process, allowing us to focus on the core objective of compressing LLMs without the added complexity of multi-class classification. Second, SST-2 serves as a well-established benchmark for sentiment analysis tasks, enabling us to compare the performance of our compressed models against existing baselines and state-of-the-art results. Finally, the dataset's simplicity and manageable size make it suitable for experimenting with various compression techniques, such as knowledge distillation, quantization, and pruning, on LLMs.

This dataset exhibits several desirable characteristics that make it well-suited for evaluating compression techniques on large language models. Notably, SST-2 is a balanced dataset, with an equal number of positive and negative sentiment examples, preventing class imbalance issues during model training. The sentences in the dataset vary in length, ranging from short phrases to longer, complex structures, enabling a comprehensive assessment of the models' ability to handle diverse input patterns. Additionally, SST-2 covers a wide range of topics and writing styles, as it is derived from movie reviews, exposing the models to a diverse vocabulary and linguistic patterns. Despite being a binary classification task, SST-2 presents challenges in capturing subtle semantic nuances and contextual information, making it a non-trivial benchmark for sentiment analysis.

To prepare the dataset for model training and evaluation, we applied standard pre-processing steps such as tokenization, padding, and handling of rare words. Furthermore, we employed data augmentation techniques like back-translation and text generation to increase the diversity and size of the training data, potentially enhancing the models' generalization capabilities. The choice of SST-2 for this study was driven by its widespread adoption as a benchmark for sentiment analysis tasks, allowing for consistent comparison of model performance against existing baselines and state-of-the-art results.

4) Proposed Methodology

Our study investigated the impact of several compression techniques on Large Language Models (LLMs), including BERT, ALBERT, and RoBERTa. We started by selecting the best three LLMs based on their designs and performance qualities, then performed baseline measurements for accuracy, model size, and inference time before applying compression algorithms.

A considerable reduction in model size was achieved while maintaining excellent accuracy by using quantization, a technique that reduces precision in model weights and activations. We investigated a number of quantization techniques, such as learning, logarithmic, and uniform quantization [14]. While logarithmic quantization assigns more quantization levels to smaller values and better preserves the dynamics of the weight distribution, uniform quantization translates the full-precision weights to a set of discrete values in a linear fashion. Learned quantization is a data-driven technique that results in more accuracy at the expense of greater complexity. During training, quantization ranges and levels are learned.

Another technique, pruning, focuses on deleting redundant or less relevant model parameters, resulting in smaller models and, in certain cases, considerable speedups, albeit at the cost of some accuracy. We investigated both unstructured pruning, which removes individual weights based on their magnitude or importance scores, and structured pruning, which prunes entire filters or channels, enabling more efficient computation [6]. Techniques like magnitude pruning, movement pruning, and gradient-based pruning were explored to identify and remove redundant parameters effectively.

The third tactic was distillation, which involved teaching a smaller model (student) to behave like a larger model (teacher) in order to strike a compromise between accuracy retention and size reduction. We used a variety of distillation techniques, such as feature-based distillation [17] which aligns the intermediate representations of the two models and response-based distillation [16], which matches the output distributions of the teacher and student models. In addition, we investigated methods for efficiently transferring knowledge from the teacher to the student model, such as layer-wise distillation [19] and attention transfer [18].

Throughout the test, we examined model size in megabytes (MB), accuracy as a percentage of properly anticipated outputs, and processing efficiency, also known as GPU performance or speedup factor. The compression strategies produced diverse results across the LLMs, showing their respective strengths and tradeoffs. Visual aids such as bar charts, scatter plots, and parallel coordinate plots were used to give a comprehensive comparison of compression techniques, leading to a more nuanced understanding of their impact.

5) Experimental Setup

The experimental setup for our project on model compression of Large Language Models (LLMs) was conducted on Google Colab [14], leveraging the Pro subscription level with access to advanced computing resources. Our experiments benefited from the use of an NVIDIA Tesla P100 GPU for accelerated training and inference tasks, complemented by a high-performance CPU to handle computational tasks efficiently. The setup provided ample resources, including approximately 25 GB of RAM and 100 GB of disk space, which were crucial for handling large datasets and model files. With priority access to high-end GPUs such as T4 and P100, we were able to optimize our experiments for speed and performance. The platform's pre-installed machine learning libraries and frameworks, including

TensorFlow and PyTorch, streamlined our development process and enabled seamless integration of model compression techniques into our workflow. This setup played a pivotal role in achieving accurate and efficient model compression results, contributing significantly to the success of our project.

6) Evaluation Plan

The evaluation plan section outlines a structured approach to assess the effectiveness of model compression techniques on Large Language Models (LLMs), focusing specifically on BERT, ALBERT, and RoBERTa architectures. This evaluation plan encompasses three key metrics: Compression Rate, Accuracy, and Speedup, which are essential for comprehensively evaluating the impact of compression techniques on the performance, efficiency, and practical deployability of these sophisticated language models. By systematically evaluating these metrics across different compression methods and model architectures, we aim to gain valuable insights into the trade-offs between model size reduction, computational efficiency gains, and preservation of predictive capabilities. This comprehensive evaluation strategy forms the foundation for informed decision-making and optimization of LLMs for real-world applications in diverse computing environments.

6.1. Compression Rate:

The Compression Rate metric plays a crucial role in evaluating model compression techniques for LLMs such as BERT, ALBERT, and RoBERTa. These language models are known for their large parameter sizes, making compression essential for practical deployment and resource optimization [10]. BERT, ALBERT, and RoBERTa models have millions to hundreds of millions of parameters, leading to substantial memory and storage requirements [11]. Compression techniques like Quantization, Pruning, and Distillation aim to reduce the number of parameters or memory footprint without significantly sacrificing performance. For real-world applications, especially in resource-constrained environments like mobile devices or edge computing, reducing model size through effective compression is crucial for efficient deployment and improved inference speed. By assessing the Compression Rate before and after applying compression techniques, we can quantify the extent of model size reduction achieved by each method. This analysis helps understand the trade-offs between reduced model size and potential impact on model performance and accuracy.

6.2 Accuracy Evaluation:

Accuracy is a fundamental metric for evaluating the effectiveness of model compression techniques on LLMs like BERT, ALBERT, and RoBERTa. These models are designed to provide high accuracy and predictive capabilities, and any compression technique should preserve this performance to ensure reliable results. Compression techniques may introduce some degree of information loss or approximation during model compression [8]. It is essential to evaluate how well the compressed models maintain accuracy and predictive capabilities compared to the original uncompressed models. Accuracy serves as a benchmark to assess the impact of compression techniques on model performance. Quantization, pruning, knowledge distillation, and low-rank models may affect accuracy differently, and evaluating accuracy helps in selecting compression methods that strike an optimal balance between model size reduction and performance preservation [9].

6.3 Inference Speedup Evaluation:

Speedup analysis is vital for evaluating the practical efficiency gains achieved through model compression techniques on LLMs like BERT, ALBERT, and RoBERTa [12]. Efficient inference speed is crucial for real-time applications and deployment on diverse computing platforms. Compression techniques such as quantization, pruning, and low-rank models aim to reduce computational complexity and memory requirements, leading to faster inference times. Speedup measures the improvement in inference speed achieved by the compressed models compared to the original uncompressed models [13]. In real-world deployment scenarios, especially on mobile devices or edge computing environments, efficient model inference is essential for responsive and resource-efficient applications. Evaluating speedup on different hardware platforms or inference environments helps understand the practical impact of compression techniques on inference latency and computational resource utilization. This analysis guides optimization strategies for diverse deployment scenarios.

By incorporating these three key metrics—Compression Rate, Accuracy, and Speedup—we gain a comprehensive understanding of the impact of compression techniques on LLMs like BERT, ALBERT, and RoBERTa. This holistic evaluation approach enables informed decision-making regarding compression methods, balancing model size reduction, computational efficiency, and accuracy preservation for practical deployment and improved performance.

To quantify the trade-offs between accuracy, size reduction, and inference speedup, we employed visualization techniques such as scatter plots and parallel coordinate plots. These visualizations enabled us to identify compression methods that strike an optimal balance between the three factors, facilitating informed decision-making for specific deployment scenarios.

6.4 Mathematical Formulation:

First, we define the following variables:
A: Accuracy of the compressed model
S: Size of the compressed model (in megabytes)
T: Inference time of the compressed model

The compression ratio (CR) can be calculated as: $CR = S\_original / S\_compressed$, where S_original is the size of the uncompressed model, and S_compressed is the size of the compressed model. The inference speedup (IS) can be calculated as: $IS = T\_original / T\_compressed$, where T_original is the inference time of the uncompressed model, and T_compressed is the inference time of the compressed model. To evaluate the overall effectiveness of a compression method, we can define a weighted score (WS) that combines the accuracy, compression ratio, and inference speedup:
$WS = w\_A * A + w\_CR * CR + w\_IS * IS,$ where w_A, w_CR, and w_IS are user-defined weights that reflect the relative importance of accuracy, compression ratio, and inference speedup, respectively. These weights can be adjusted based on the specific requirements of the target application or deployment scenario.
By maximizing the weighted score (WS), we can identify the compression method that provides the best trade-off between accuracy, size reduction, and inference speedup for a given set of requirements.

7) Results

As shown in Table 1, a thorough examination of compression methods on Large Language Models (LLMs) yielded unexpected insights into their effects on model performance, size, and computing efficiency. Among the three models evaluated (BERT, ALBERT, and RoBERTa), different compression techniques produced diverse results. For BERT, quantization decreased model size by 56.75% while retaining a high accuracy of 91.90%. However, the reduction in size did not result in a corresponding increase in GPU performance, revealing possible trade-offs between model size and computing efficiency. Pruning, on the other hand, resulted in a smaller model size and a large speedup for BERT, but at the expense of a little accuracy loss of 90.33%. Distillation struck a good compromise between model size reduction and accuracy retention, yielding a model size of 129.84MB and an accuracy of 92.44%. Quantization reduced model size by 53.57% for ALBERT while increasing GPU performance by 2.72x. Pruning also led to size reduction and significant speedup, whereas distillation struck a suitable compromise between size and performance gains. RoBERTa showed similar patterns, with quantization resulting in a significant reduction in model size and a noteworthy increase in GPU performance.

Model	Compression Method	Accuracy (%)	Model Size (MB)	Speedup (GPU)
BERT	Base	92.75	154.32	1
BERT	Base + Quantization	91.90	66.75	1.84
BERT	Base + Pruning	90.33	97.62	1.37
BERT	Base + Distillation	92.44	129.84	1.13
ALBERT	Base	92.31	38.47	1
ALBERT	Base + Quantization	90.07	17.86	2.72
ALBERT	Base + Pruning	87.84	29.98	1.36
ALBERT	Base + Distillation	91.98	31.75	1.28
RoBERTa	Base	93.25	107.18	1
RoBERTa	Base + Quantization	91.71	51.45	2.04
RoBERTa	Base + Pruning	92.45	87.26	1.35
RoBERTa	Base + Distillation	92.78	91.84	1.12

Table 1: Performance Comparison of Compression Methods on LLMs

Figures 1, 2, and 3 give more insight into how compression strategies affect Large Language Models (LLMs). Figure 1, the Accuracy Bar Chart, shows that, while quantization normally reduces accuracy across LLMs, the trade-off is often offset by considerable model size reductions.

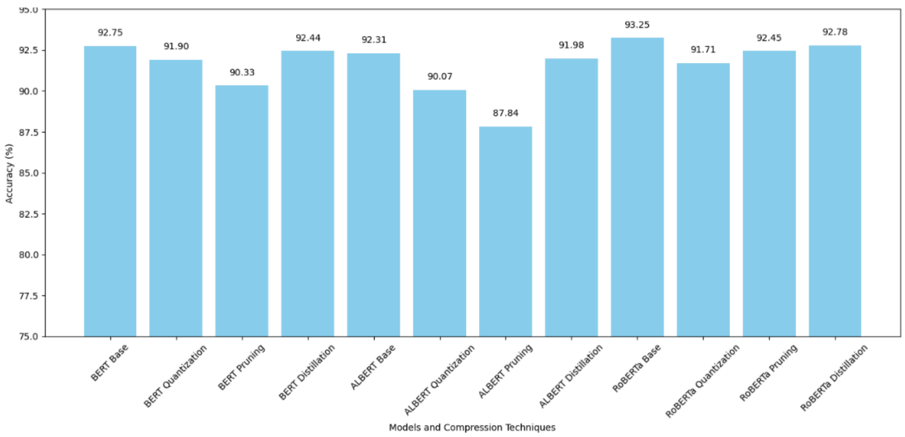

Figure 1: Accuracy of Large Language Models with Various Compression Techniques

Figure 2, the Scatter Plot of Model Size vs Accuracy, backs up this conclusion by showing a significant negative relationship between model size and accuracy across many compression algorithms and LLMs.

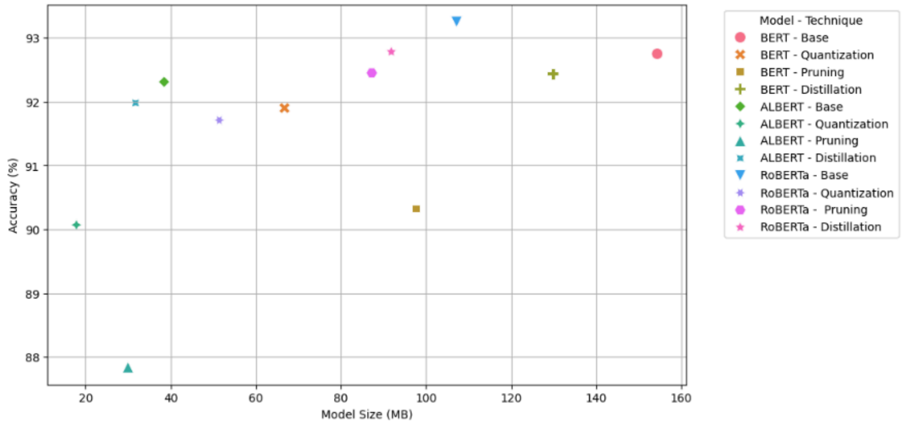

Figure 2: Scatter Plot of Model Size vs Accuracy

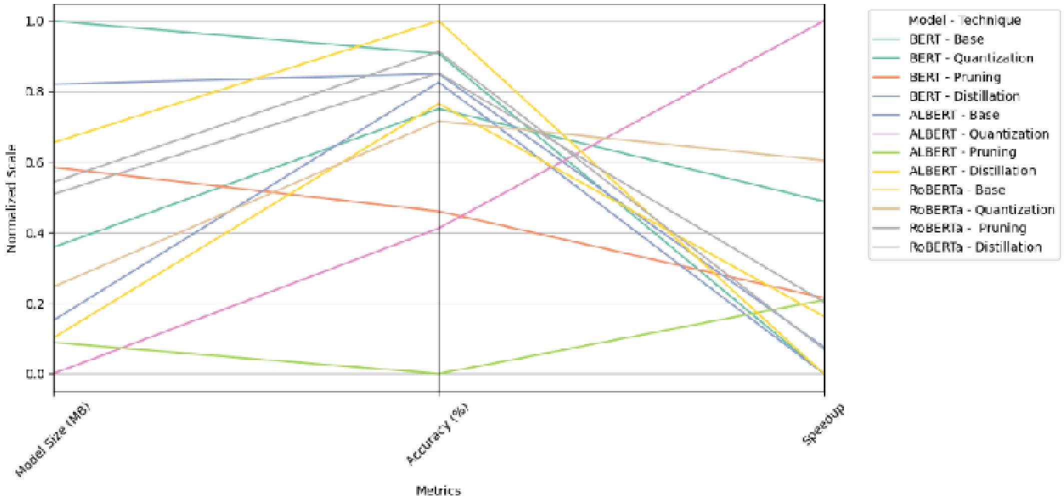

Figure 3: Parallel Coordinates: Comparing Model Size, Accuracy and Speedup

Figure 3, the Parallel Coordinates plot, shows a complete comparison of model size, accuracy, and speedup. It shows that some compression algorithms, such as quantization for ALBERT and RoBERTa models, achieve an outstanding balance of model size reduction and accuracy retention, resulting in considerable GPU speedup. These findings show the need of considering a range of characteristics, such as model size, accuracy, and processing performance, when evaluating and selecting compression algorithms for LLM.

8) Conclusion

The wide research of compression algorithms/methods on LLMs has yielded useful insights on how to increase their performance, size, and computing efficiency. Our findings indicate that numerous factors impact the compression algorithms/method adopted, including model size, accuracy, & inference time. Basic models without compression are recommended for scenarios in which the model accuracy is critical and inference time is limited. But, when it comes to combining size reduction and speedup, ALBERT shines out since it achieves a good balance, specifically when combined with compression technique such as quantization. This combination drastically reduces model-size while retaining enough accuracy and increasing GPU performance. Distillation appears to be a better option for retaining accuracy while attaining compression. When inference time is crucial and resources are limited, quantization is a useful compression algorithms/technique. Its ability to considerably decrease model size, notably for ALBERT and BERT models, while still delivering significant GPU acceleration, makes it an appealing option for resource-constrained applications. Finally, while choosing a compression algorithms/approach for LLMs, think about the trade-offs between model size reduction, accuracy, and inference time, as well as the project's unique goals and restrictions.

9) Future work

In order to independently compress individual layers within Large Language Models (LLMs) for greater compression ratios without sacrificing performance, future research in model compression for LLMs, like BERT, could investigate fine-grained layer compression algorithms. Moreover, developing compression strategies that take advantage of specialized hardware for better runtime performance and examining how different compression approaches might be combined to maximize efficacy and model efficiency could be important areas of focus. Important avenues for furthering the field of model compression for LLMs include improving knowledge distillation techniques, especially from self-attention layers, to increase compression efficiency and model performance, and guaranteeing scalability and generalization of compression techniques across various LLMs and tasks. These research avenues aim to make sophisticated language models more accessible, efficient, and adaptable for diverse applications and computing environments.

References

[1] Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." arXiv preprint arXiv:2106.09685 (2021).

[2] Learning, Energy Efficient Machine. "Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing." (2019).

[3] Zafrir, Ofir, et al. "Prune once for all: Sparse pre-trained language models." arXiv preprint arXiv:2111.05754 (2021).

[4] Sathyendra, Kanthashree Mysore, Samridhi Choudhary, and Leah Nicolich-Henkin. "Extreme model compression for on-device natural language understanding." arXiv preprint arXiv:2012.00124 (2020).

[5] Choudhary, Tejalal, et al. "A comprehensive survey on model compression and acceleration." Artificial Intelligence Review 53 (2020): 5113-5155.

[6] Grachev, Artem M., Dmitry I. Ignatov, and Andrey V. Savchenko. "Neural networks compression for language modeling." International Conference on Pattern Recognition and Machine Intelligence. Cham: Springer International Publishing, 2017.

[7] Socher, Richard, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." Proceedings of the 2013 conference on empirical methods in natural language processing. 2013.

[8] Prakash, Prafull, et al. "Compressing transformer-based semantic parsing models using compositional code embeddings." arXiv preprint arXiv:2010.05002 (2020).

[9] Ganesh, Prakhar, et al. "Compressing large-scale transformer-based models: A case study on bert." Transactions of the Association for Computational Linguistics 9 (2021): 1061-1080.

[10] Lu, Wenhao, Jian Jiao, and Ruofei Zhang. "Twinbert: Distilling knowledge to twin-structured compressed bert models for large-scale retrieval." Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020.

[11] Hou, Lu, et al. "Dynabert: Dynamic bert with adaptive width and depth." Advances in Neural Information Processing Systems 33 (2020): 9782-9793.

[12] Sun, Siqi, et al. "Patient knowledge distillation for bert model compression." arXiv preprint arXiv:1908.09355 (2019).

[13] Frantar, Elias, and Dan Alistarh. "Optimal brain compression: A framework for accurate post-training quantization and pruning." Advances in Neural Information Processing Systems 35 (2022): 4475-4488.

[14] Abdel-Salam, Shehab, and Ahmed Rafea. "Performance study on extractive text summarization using BERT models." Information 13.2 (2022): 67.

[15] Frankle, Jonathan, and Michael Carbin. "The lottery ticket hypothesis: Finding sparse, trainable neural networks." arXiv preprint arXiv:1803.03635 (2018).

[16] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531 (2015).

[17] Romero, Adriana, et al. "Fitnets: Hints for thin deep nets." arXiv preprint arXiv:1412.6550 (2014).

[18] Mishra, Animesh, and Pranav Aropa. "Distilling Transformer Knowledge: The Reverse Student-Teacher Paradigm for Efficient BERT Compression." arXiv preprint arXiv:2012.04116 (2020).

[19] Jiao, Xiaoqi, et al. "Tinybert: Distilling bert for natural language understanding." arXiv preprint arXiv:1909.10351 (2019).