Model Compression Techniques for Transformer Models: A Survey

Abstract

This survey paper provides a comprehensive analysis of model compression techniques essential for optimizing transformer models like TinyBERT in edge computing environments. The primary focus is on knowledge distillation, pruning, and quantization, which are pivotal in reducing the size and computational demands of these models while maintaining high performance in natural language processing (NLP) tasks. Despite the transformative impact of transformer models, their deployment in resource-constrained settings poses significant challenges due to substantial computational and memory requirements. The survey examines how these compression techniques address the latency-accuracy tradeoff, thereby enhancing computational efficiency. Knowledge distillation emerges as a particularly influential method, enabling smaller models to approximate the performance of larger ones through effective knowledge transfer. Pruning techniques contribute by removing redundant parameters, thus streamlining models for real-time applications. Quantization methods further reduce computational complexity by lowering the precision of model weights and activations. Additionally, the survey explores hybrid techniques that combine these methods for optimized performance. The paper highlights the importance of these techniques in edge computing environments, where real-time AI computation and resource efficiency are critical. Through detailed exploration of innovative approaches and real-world applications, the survey underscores the potential of model compression techniques to enhance the deployment of transformer models in diverse and resource-limited settings. Future research directions are suggested to further optimize these techniques, ensuring robust and efficient models across various applications.

1 Introduction

1.1 Significance of Transformer Models in NLP

Transformer models have transformed natural language processing (NLP) through attention mechanisms that efficiently manage complex linguistic tasks [1]. Large-scale pre-trained models like BERT have set new performance benchmarks, excelling in sentiment analysis, machine translation, and question answering. Their self-attention mechanisms enable parallel data processing, significantly enhancing relevance prediction accuracy [2].

However, deploying transformer models in resource-constrained environments is challenging due to their high computational and memory requirements. This issue is particularly acute in edge computing, where mobile and embedded systems have limited resources [3]. The substantial memory footprint of models like BERT complicates their practicality on such devices, increasing latency in both local and cloud setups. Additionally, the high operational costs and environmental impact highlight the urgent need for sustainable AI practices that reduce energy consumption and carbon emissions.

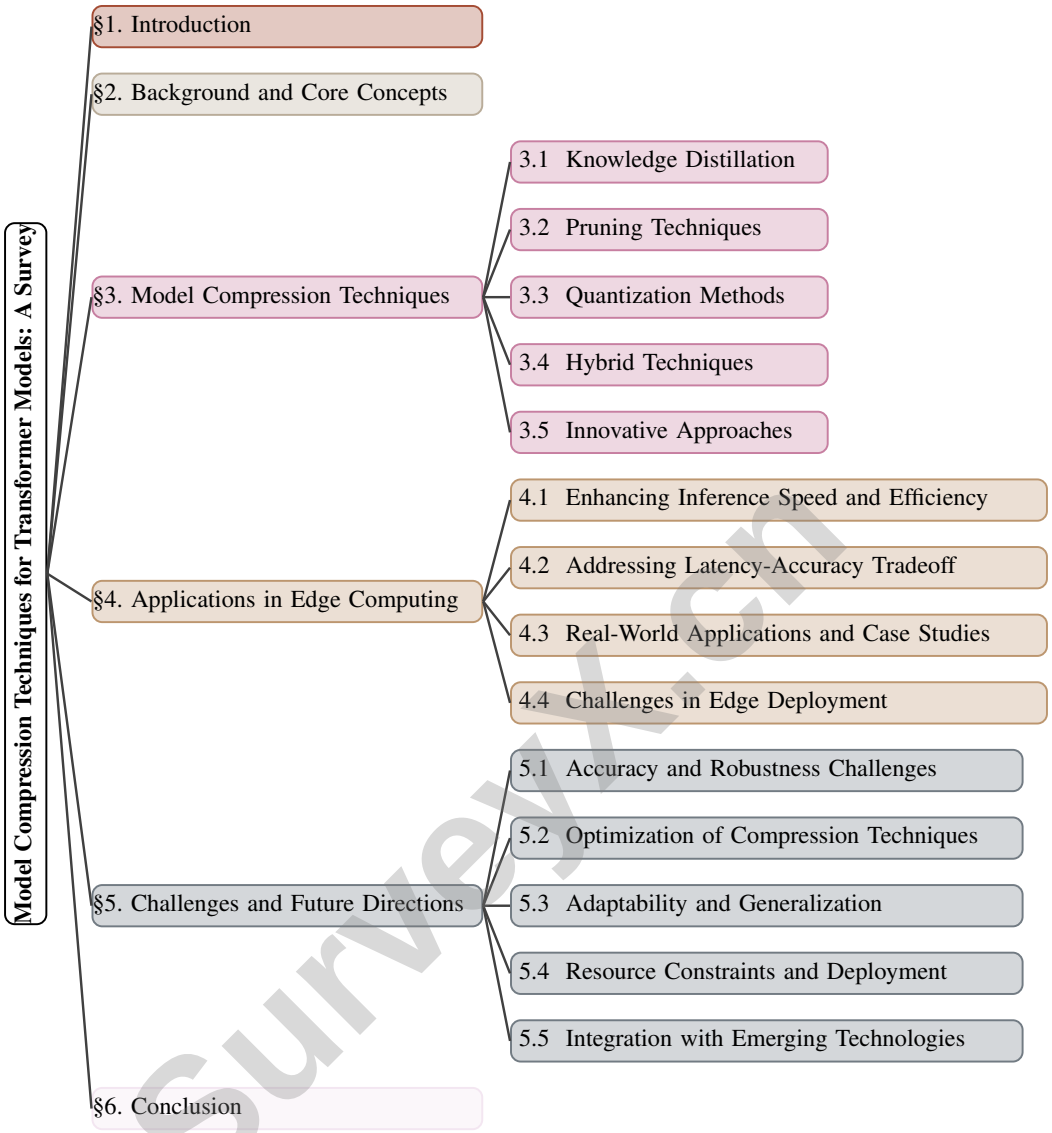


Figure 1: chapter structure

To address these challenges, various model compression techniques, including knowledge distillation, have emerged to reduce model size while maintaining performance. These techniques are vital for deploying transformer models effectively in resource-sensitive applications [4]. The evolution of large language models (LLMs) like ChatGPT and LlaMA further underscores the necessity for optimizing model architectures to sustain their transformative role in generative AI [5]. Recent advancements in model compression tailored for LLMs tackle the challenges of size and computational demands in limited-resource contexts [6]. Innovative strategies such as data-free knowledge distillation (DFKD) are being explored to mitigate reliance on original training data, which may be restricted due to privacy concerns [7]. These initiatives are essential for making the advantages of transformer models accessible and sustainable across diverse deployment scenarios.

1.2 Challenges of Computational Demands

Deploying transformer models in resource-constrained environments, such as smartphones and IoT devices, poses significant challenges due to their extensive computational and memory demands. These models, often comprising millions of parameters, create practical hurdles for real-time applications on edge devices where resources are limited [8]. The rapid expansion of large language models

(LLMs) exacerbates these challenges, resulting in heightened resource consumption and increased training times [6].

Existing model compression techniques, including pruning and distillation, often struggle to balance speed and accuracy, particularly in edge computing scenarios [9]. Applying weight quantization at low bit-widths can lead to significant accuracy loss, complicating efforts to maintain model performance [10]. Furthermore, the inefficiency of existing Cross-Encoders in scoring numerous documents within tight latency constraints highlights the urgent need for methods that enhance user satisfaction and energy efficiency in retrieval systems [11].

Additionally, the compute-bound nature of inference in large language models, especially with large batch sizes or lengthy sequences, leads to substantial latency due to intensive matrix operations [12]. This reliance on significant computational resources emphasizes the necessity for innovative compression techniques that effectively balance computational efficiency and model performance without sacrificing accuracy [8]. Addressing these computational challenges requires developing novel strategies that preserve model capacity while reducing size and computational load [6]. Compressing large models like BERT to alleviate their computational demands while maintaining performance across various NLP tasks remains a critical research focus [13].

1.3 Need for Model Compression Techniques

The rapid increase in the parameter size of transformer models, such as BERT, necessitates effective model compression techniques for deployment on resource-constrained devices [14]. The rising computational costs associated with deep neural networks (DNNs) further highlight the need for efficient compression methods, including pruning and quantization [12]. Traditional strategies like knowledge distillation, pruning, and quantization have been extensively examined to mitigate the challenges of deploying large models in practical settings [6]. However, these methods may require re-training or impose architectural constraints, indicating the need for innovative approaches that leverage large models without compromising performance [15].

The practical deployment of models on mobile devices is hindered by the difficulty in balancing model size, computational efficiency, and decoding speed [16]. Existing methods, such as the MLKD-BERT approach, enhance model compression by distilling both feature-level and relation-level knowledge, thus improving efficiency and effectiveness [17]. Despite these advancements, a breakthrough is necessary to simplify the compression process while retaining high performance [13]. Developing innovative compression strategies is critical to optimizing performance and ensuring that the advantages of transformer models are fully realized in various deployment contexts.

1.4 Importance in Edge Computing Environments

Integrating transformer models into edge computing environments is increasingly vital due to the demand for real-time AI computation and the management of latency and privacy concerns [1]. Model compression techniques play a crucial role in this context by significantly lowering computational and memory requirements, enabling model deployment on resource-constrained devices. Compressed models must sustain high accuracy while operating under limited resources to ensure fairness and effectiveness in NLP applications across diverse settings [18].

Innovative methods, such as CompactifAI, showcase the potential of model compression to improve performance in edge computing by leveraging quantum-inspired Tensor Networks. This approach focuses on model correlations rather than merely reducing neuron count, enhancing performance without substantial retraining or architectural changes [5]. Such adaptability is essential for edge devices, where resource limitations and efficiency are paramount. Techniques like Sparse Decomposed Quantization (SDQ) combine structured sparsity and quantization to improve compute and memory efficiency, reflecting ongoing efforts to customize compression methods for the unique challenges of edge computing [19].

Quantization methods, particularly those utilizing ultra-low bit precision, have made significant strides in model compression for edge environments, allowing large NLP models to be efficiently deployed on resource-constrained devices through optimized compute and memory usage [20]. Application-Specific Compression (ASC) offers another promising avenue, enabling the creation of

tailored compressed models that maintain high performance for specific tasks, thereby addressing the limitations of traditional application-agnostic methods [21].

Despite these advancements, challenges remain, particularly in optimizing the interplay between sparsity and quantization, which requires systematic exploration to enhance efficiency and performance [22]. Techniques like TQCompressor demonstrate the capacity to significantly reduce model size without sacrificing performance, making them suitable for deployment in resource-constrained environments [23]. The continuous evolution of model compression techniques, benchmarked by frameworks such as KD-Lib, is essential for fully leveraging the capabilities of transformer models in edge computing, ensuring efficient and effective operation across a broad spectrum of applications [24].

1.5 Structure of the Survey

This survey is structured to provide a comprehensive analysis of model compression techniques for transformer models, with a focus on their application in edge computing environments. The introduction establishes the significance of transformer models in NLP and the computational challenges they present, particularly in resource-constrained settings, while emphasizing the critical need for model compression techniques to enhance computational efficiency.

Following the introduction, the survey delves into the background and core concepts, offering an overview of transformer models and foundational principles of model compression, including knowledge distillation, pruning, and quantization. This section serves as a foundation for understanding subsequent discussions on specific compression techniques.

The survey then explores various model compression techniques in detail, examining knowledge distillation, pruning methods, and quantization techniques in dedicated subsections. It also highlights hybrid techniques that combine these methods to optimize performance and showcases innovative approaches at the forefront of model compression research.

The section on applications in edge computing investigates how these compression techniques enhance inference speed and efficiency, address latency-accuracy tradeoffs, and present real-world applications and case studies. It also discusses the challenges encountered when deploying compressed models in edge environments.

The penultimate section addresses challenges and future directions in model compression for transformer models, including discussions on accuracy and robustness challenges, optimization of compression techniques, adaptability across tasks, and integration with emerging technologies.

Finally, the conclusion summarizes key points discussed in the paper, reiterating the importance of model compression techniques in improving the efficiency of transformer models for NLP tasks. The survey's organization reflects a comprehensive approach to understanding and advancing the field of model compression, informed by the evaluation and combination of methods such as weight pruning, low-rank factorization, and knowledge distillation [18].The following sections are organized as shown in Figure 1.

2 Background and Core Concepts

2.1 Core Concepts of Model Compression

Model compression aims to optimize transformer models by reducing their size and computational requirements, which is essential for deploying large language models (LLMs) and large vision models (LVMs) on devices with limited processing power. Techniques such as pruning, quantization, knowledge distillation, and efficient architecture design are tailored to the structure of transformers, addressing the high parameter counts in models like BERT to ensure efficiency in applications with strict latency and resource constraints [25, 26, 27, 28]. Key methods include knowledge distillation, pruning, and quantization, each enhancing model efficiency while maintaining performance.

Knowledge distillation (KD) involves training a smaller student model to replicate the outputs of a larger teacher model, achieving similar performance with improved computational efficiency. DistilBERT exemplifies KD by creating a smaller version of BERT [29]. Stagewise Knowledge Distillation (SKD) refines this by progressively training the student model on the teacher's feature

maps, reducing data needs [30]. Combining task-specific structured pruning with distillation creates compressed models that perform well across different speed/accuracy tradeoffs [31]. The KD-Lib library integrates multiple compression techniques, supporting various algorithms and hyperparameter tuning [24].

Pruning eliminates redundant parameters and connections, enhancing model efficiency [18]. Post-training pruning removes unnecessary weights from pre-trained models without retraining, streamlining compression [32]. Methods like SlimNets, combining weight pruning, low-rank factorization, and knowledge distillation, maintain high accuracy in compressed networks [18]. Adaptive Sparsity Compression (ASC) prunes components based on their task-specific data representation contribution, optimizing performance [21].

Quantization reduces model weight and activation precision, decreasing memory use and speeding up inference, crucial for deploying large models on edge devices with limited resources [33]. Quantization-Aware Training (QAT) and methods like LCQ, which uses a low-rank codebook for weight representation, optimize quantization to minimize loss [10]. Integrating quantization with knowledge distillation effectively reduces model size while maintaining performance under constraints [4]. Challenges remain in managing unique sparsity patterns and dynamic routing in models like Mixture-of-Experts during post-training quantization [34].

Integrating these compression techniques is vital for developing robust models that maintain high performance across environments. Techniques like XTC, combining lightweight layer reduction with quantization-aware training, show the potential for extreme transformer model compression [20]. These methods are benchmarked across various deep learning tasks, highlighting their impact on image classification, object detection, language modeling, and generative applications [25].

3 Model Compression Techniques

To effectively address the challenges associated with the deployment of transformer models in resource-constrained environments, various model compression techniques have been developed. These techniques aim to optimize model efficiency while maintaining performance, thereby facilitating the use of complex models in practical applications. Among these strategies, knowledge distillation emerges as a particularly influential method that enables the transfer of knowledge from larger, more complex models to smaller, more efficient counterparts. This process not only reduces the computational burden but also enhances the adaptability of models in diverse operational contexts.

Figure 2 illustrates the hierarchical structure of model compression techniques, highlighting primary categories such as Knowledge Distillation, Pruning Techniques, Quantization Methods, Hybrid Techniques, and Innovative Approaches. Each category is further divided into methodologies, applications, types, strategies, and developments, showcasing the comprehensive landscape of methods aimed at optimizing transformer models for efficiency and performance in resource-constrained environments. The following subsection delves into the intricacies of knowledge distillation, exploring its methodologies and applications in the realm of model compression.

3.1 Knowledge Distillation

Knowledge distillation (KD) is a pivotal model compression technique that facilitates the transfer of knowledge from a larger teacher model to a smaller student model, enabling the latter to approximate the teacher's performance while significantly reducing its size and computational demands [29]. The student model learns from the teacher's outputs, which provide richer information than hard labels, leading to a more nuanced learning process [35]. This process is particularly beneficial in scenarios with noisy labels and class imbalance, where capturing and transferring knowledge effectively is crucial [36].

Advanced methodologies have evolved to enhance the KD process. For instance, Progressive Knowledge Distillation (Pro-KD) allows the student model to learn progressively from the teacher during its training, rather than from a single, fully-trained teacher model [37]. This incremental learning approach helps in refining the student's learning process, thereby improving its performance.

Moreover, the integration of KD with quantization techniques, as demonstrated by Suwannaphong et al., illustrates the potential of KD to optimize transformer models for deployment in resource-

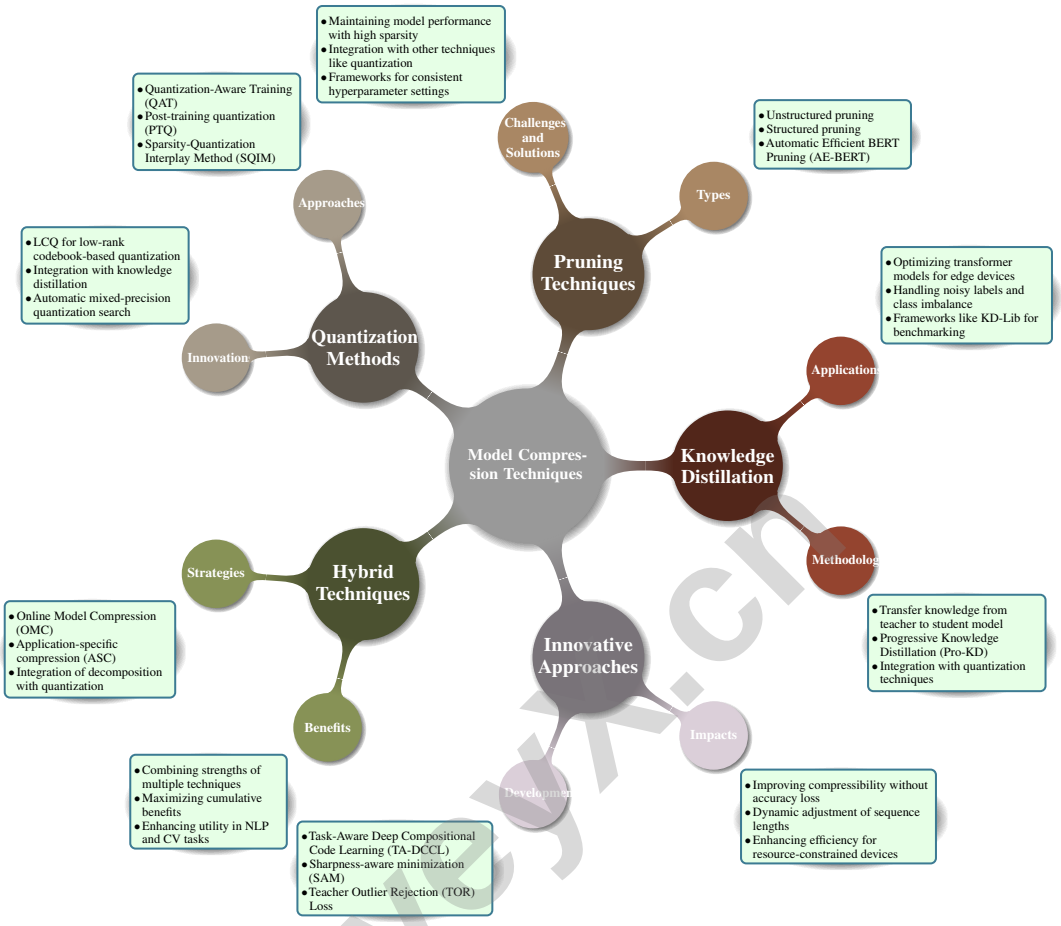


Figure 2: This figure illustrates the hierarchical structure of model compression techniques, highlighting primary categories such as Knowledge Distillation, Pruning Techniques, Quantization Methods, Hybrid Techniques, and Innovative Approaches. Each category is further divided into methodologies, applications, types, strategies, and developments, showcasing the comprehensive landscape of methods aimed at optimizing transformer models for efficiency and performance in resource-constrained environments.

constrained environments, such as edge devices [4]. The KD-Lib library offers a comprehensive benchmark of various KD algorithms, showcasing their efficacy across different tasks and datasets, which aids in selecting the appropriate method for specific applications [24].

Innovative methods like Theseus Compression propose replacing modules of BERT with compact substitutes and training them together, highlighting a novel approach to model compression [13]. Additionally, the MLKD-BERT approach leverages both feature-level and relation-level knowledge to enhance model performance while allowing flexible configurations of the student model [17].

Hybrid approaches that combine KD with other compression techniques, such as weight pruning and matrix factorization, have been explored to enhance model efficiency. For example, the integration of KD with methods like GWK, which applies clustering based on weight sensitivity, demonstrates the potential for achieving significant model compression without compromising performance [15]. Furthermore, the method proposed by Guskin et al., which dynamically adjusts sequence lengths during inference, exemplifies the adaptability of KD in optimizing efficiency while minimizing accuracy loss [8].

As the field progresses, the exploration of data-free knowledge distillation techniques, such as DFKD-T3, which leverages generative language models to create task-specific training data, exemplifies the innovative directions in which KD is being applied to overcome challenges related to data availability and privacy. Recent advancements in Knowledge Distillation (KD) methodologies, such

as Knowledge Translation (KT) and Progressive Knowledge Distillation (Pro-KD), are paving the way for significant improvements in model efficiency and adaptability across various deployment environments. These innovative approaches not only enhance model compression by enabling the transfer of knowledge from larger models to more compact ones but also address challenges like label noise and capacity gaps. With the introduction of frameworks like KD-Lib, which provides modular implementations of KD, pruning, and quantization techniques, researchers can more effectively implement and tune these methods, ultimately leading to more robust models that perform well in real-world scenarios. [35, 24, 15, 37]

3.2 Pruning Techniques

Pruning is a pivotal model compression technique that enhances the efficiency of transformer models by strategically removing redundant parameters, thereby reducing model size and computational complexity while preserving performance. This approach is especially beneficial for deploying models on resource-constrained devices, ensuring effective operation in environments with limited computational resources [9].

Various pruning methodologies have been developed, each offering unique strategies to maintain model performance while achieving significant compression. Unstructured pruning, which involves the removal of individual weights considered non-essential, allows for a more granular reduction in model size. In contrast, structured pruning eliminates entire neurons or channels, leading to more predictable improvements in computational efficiency [9]. For instance, SparseBERT exemplifies structured pruning, where the model is pruned during fine-tuning on downstream tasks to maintain performance while reducing its size [9].

Innovative approaches such as the Automatic Efficient BERT Pruning (AE-BERT) automate the pruning process, evaluating sub-networks without the need for extensive fine-tuning, thereby minimizing the computational cost and time required for model optimization [38]. Similarly, the Greedy Layer Pruning (GLP) technique prunes layers just before fine-tuning, offering a flexible trade-off between performance and speed [39].

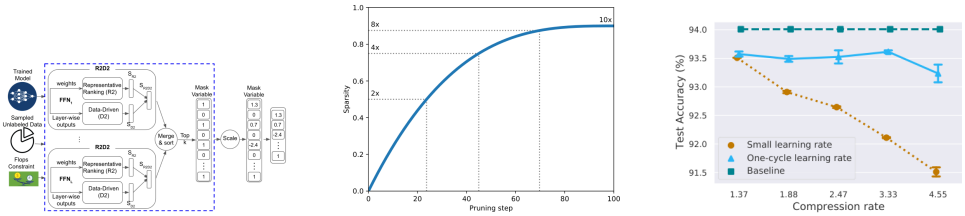
Hybrid methods that combine pruning with other compression techniques have also been explored. For instance, the PQK method operates in two phases: initially focusing on pruning and quantization to create a lightweight model, followed by employing knowledge distillation to enhance the model's performance [40]. This integration of techniques underscores the potential for achieving better compression ratios without significant accuracy degradation.

The CoFi method combines coarse-grained and fine-grained pruning with a layerwise distillation objective to optimize model compression [9]. This method highlights the potential of structured pruning to significantly reduce model size while maintaining performance, although achieving large speedups comparable to distillation remains challenging [9].

Challenges persist in leveraging the full potential of sparsity for performance gains, as existing methods often struggle to optimize the pruning rate effectively, which can lead to slower operations compared to dense matrix computations [41]. Furthermore, existing methods typically necessitate separate training or fine-tuning for each compression target, which is computationally expensive and prone to errors [12]. The application of consistent hyperparameter settings across different compression techniques, as facilitated by frameworks like KD-Lib, ensures fair comparisons and enhances the evaluation of pruning methods [24].

The application of pruning techniques is crucial for optimizing the deployment of transformer models, particularly in ultra-low power devices where maintaining performance while reducing resource usage is critical [42]. By addressing the challenges of maintaining model performance and parametric knowledge during compression, pruning techniques play a vital role in the efficient deployment of NLP models across various environments [18].

As shown in Figure 3, Model compression techniques, particularly pruning techniques, play a crucial role in enhancing the efficiency of machine learning models by reducing their complexity without significantly sacrificing performance. This concept is vividly illustrated through several examples, as depicted in the accompanying figure. The first example, "Masked Representation Learning for Efficient Unsupervised Learning," showcases a flowchart of a method designed to handle unlabeled data by leveraging a pre-trained model to create robust data representations, a



(a) Masked Representation Learning for Efficient Unsupervised Learning[43]

(b) Sparse Sparsity[44]

(c) Comparison of Test Accuracy with Different Compression Rates and Learning Rates[45]

Figure 3: Examples of Pruning Techniques

process particularly beneficial in scenarios with missing or noisy labels. The second example, "Sparse Sparsity," presents a graph detailing the relationship between the number of pruning steps and model sparsity, highlighting how incremental pruning can increase model sparsity, which is key to reducing model size and computational demands. Lastly, the "Comparison of Test Accuracy with Different Compression Rates and Learning Rates" graphically compares test accuracies of models subjected to varying compression rates, with different learning rate strategies, underscoring the trade-offs between model size reduction and performance. Together, these examples underscore the versatility and effectiveness of pruning techniques in model compression, offering insights into optimizing model architectures for better performance and resource efficiency. [**?**]nova2023gradientfreestructuredpruningunlabeled,zhu2017prunepruneexploringefficacy,le2020payingattentionsnapshotsiterative

3.3 Quantization Methods

Quantization methods play a crucial role in reducing the computational complexity of transformer models by converting high-precision weights and activations into lower-precision formats, thereby significantly decreasing memory usage and accelerating inference. This approach is particularly essential for deploying large models on resource-constrained devices, such as microcontroller units (MCUs), where computational resources are limited [42]. The process involves mapping continuous values to discrete levels, leading to substantial reductions in model size and computation time while maintaining acceptable levels of accuracy [46].

Quantization-Aware Training (QAT) is a prominent approach that incorporates quantization into the training process to mitigate the impact of precision loss, generally yielding models with better performance under distribution shifts compared to standard training . By implementing QAT, as demonstrated in methods like LCQ, which employs a low-rank codebook for weight representation, the quantization process is optimized to minimize accuracy loss [10]. Continuous approximations further enhance QAT by allowing more accurate learning of quantization parameters, thereby improving model performance [47].

Innovative quantization methods also include techniques such as the Sparsity-Quantization Interplay Method (SQIM), which optimally combines sparsity and quantization by applying sparsity before quantization to minimize computation errors [22]. The integration of quantization with knowledge distillation, as demonstrated by QKD, shows significant improvements in accuracy over baseline methods and effectively recovers full-precision accuracy at low-bit quantization levels [48]. Additionally, the method proposed by Yang et al. compresses the embedding and linear layers of transformers into low-rank tensor cores while applying quantization-aware training to minimize model size and runtime latency [49].

Post-training quantization (PTQ) methods offer another approach, focusing on optimizing models after training to achieve efficient deployment in NLP and CV tasks [28]. The application of INT4 quantization, which utilizes 4-bit precision for both weights and activations, has been investigated to optimize inference performance in transformer models [50]. Furthermore, the OMC framework achieves compression through quantization and efficient parameter management, allowing for lightweight operation during federated training [51].

AI-generated, for reference only.

Advanced quantization frameworks, such as CVXQ, optimize the bit depth and step size for each weight parameter to minimize output distortion, providing a novel perspective on quantization [52]. The AQ-BERT model exemplifies the potential of automatic mixed-precision quantization search to provide significant performance improvements over existing methods, particularly in low-compression scenarios [53]. The primary innovation in quantization methods, such as FlattenQuant, is the introduction of structured formats for weight representation that allow for regular memory access patterns and high parallelism in decoding .

Overall, quantization methods are essential for developing efficient transformer models, enabling their deployment in diverse environments where computational resources are limited. The continuous evolution of these techniques, supported by innovative frameworks and methodologies, promises to further enhance the efficiency and applicability of transformer models across various domains [54].

3.4 Hybrid Techniques

Hybrid techniques in model compression have emerged as a comprehensive approach to optimizing transformer models by integrating multiple strategies such as pruning, quantization, and knowledge distillation. These methods aim to enhance model performance and efficiency by leveraging the strengths of individual techniques to achieve a more substantial reduction in model size and computational demands while maintaining high accuracy [18].

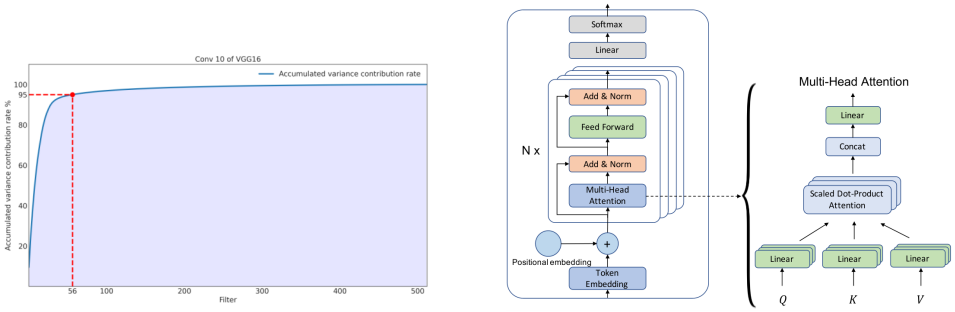
One innovative approach is the Online Model Compression (OMC) framework, which performs compression and decompression in every training iteration. This capability effectively reduces memory usage and communication costs, making it suitable for federated learning environments where preserving accuracy is crucial [51]. The application-specific compression (ASC) method exemplifies another hybrid strategy by performing a single forward pass over the training data to identify which layers can be pruned without significantly impacting performance, thus optimizing model efficiency for specific tasks [21].

The TQCompressor introduces a novel hybrid technique by optimizing neuron connections through permutations followed by Kronecker decomposition. This method achieves efficient model compression by enhancing the structural efficiency of neural networks [23]. Additionally, the integration of low-rank tensor decomposition with quantization-aware training, as demonstrated by Yang et al., enables high compression ratios with minimal accuracy loss, showcasing the potential of combining decomposition techniques with quantization for effective compression [49].

The CrAM framework advances hybrid compression methods by producing models robust to one-shot compression, allowing for significant sparsity levels without the need for retraining. This approach reduces the computational burden associated with model compression, making it an attractive option for scenarios requiring efficient deployment [12]. Furthermore, the exploration of data-free knowledge distillation (DFKD) frameworks, such as the extension to generative language models, highlights the potential of hybrid techniques to enhance specificity and diversity in NLP tasks without relying on original training data [7].

Hybrid techniques, which combine multiple efficiency methods such as pruning, quantization, and knowledge distillation, are emerging as a highly effective strategy for improving the efficiency and performance of transformer models. These techniques not only facilitate the implementation of large language and vision models on practical devices by reducing memory and computational costs, but also allow for the sequential application of various methods, maximizing their cumulative benefits while maintaining consistent performance outcomes. This multifaceted approach addresses the unique architectural challenges of transformers, thereby enhancing their utility in natural language processing and computer vision tasks. [55, 28]. By combining the strengths of various compression methods, these techniques offer a holistic approach to optimizing model architectures, ensuring they remain effective and efficient across diverse deployment scenarios.

As shown in Figure 4, In the realm of model compression techniques, hybrid techniques have emerged as a powerful approach to optimize deep learning models by combining different strategies to achieve both efficiency and performance. The examples illustrated in Figure 4 highlight two distinct applications of hybrid techniques. The first example focuses on the "Convolutional Layer 10 of VGG16: Accumulated Variance Contribution Rate," where the graph demonstrates how the accumulated variance contribution rate increases gradually as the filter size varies from 1 to 500. This visualization underscores the potential of leveraging variance contribution rates for effective



(a) Convolutional Layer 10 of VGG16: Accumulated Variance Contribution Rate[56]

(b) Multi-Head Attention Network[57]

Figure 4: Examples of Hybrid Techniques

model slimming, as suggested by Wang et al. . The second example showcases a "Multi-Head Attention Network," a sophisticated architecture frequently employed in natural language processing tasks. This network integrates several layers, including positional and token embedding layers, alongside a multi-head attention layer that consists of linear, concatenation, and scaled dot-product attention layers. As detailed by Wei et al. , such hybrid techniques are instrumental in enhancing the model's ability to process sequences of tokens and attend to different parts of the input data, thereby achieving a more balanced trade-off between computational efficiency and accuracy. [**?**]wang2022novelarchitectureslimmingmethod,wei2023greenerpowerfultaminglarge)

3.5 Innovative Approaches

Innovative approaches in model compression are continuously evolving to enhance the efficiency and performance of transformer models, addressing the challenges of resource consumption and accuracy. One notable advancement is the development of Task-Aware Deep Compositional Code Learning (TA-DCCL), which achieves a remarkable compression rate of 98.4

The integration of sharpness-aware minimization (SAM) with various model compression techniques represents a significant leap in achieving better compressibility without sacrificing accuracy. SAM's ability to stabilize the training process by smoothing the loss landscape enhances the robustness of compressed models, ensuring they maintain high performance across different tasks. Additionally, the introduction of the Teacher Outlier Rejection (TOR) Loss explicitly rejects outliers based on teacher model predictions, refining the distillation process and improving student model performance [36]. This innovation addresses the challenge of noisy labels and enhances the robustness of knowledge transfer.

Pro-KD, a progressive knowledge distillation method, mitigates both the capacity-gap and checkpoint-search problems by allowing the student to learn from multiple checkpoints of the teacher throughout the training process [37]. This incremental learning approach helps in refining the student's learning process, thereby improving its performance. The survey categorizes KD methods based on the aspects of the teacher they mimic, such as response distillation, representation space distillation, and relational knowledge distillation, offering a comprehensive framework for understanding and applying KD techniques [35].

Theseus Compression proposes replacing modules of BERT with compact substitutes and training them together, highlighting a novel approach to model compression that simplifies the training process by not requiring additional loss functions [13]. This method allows for a more direct interaction between the original and compressed models, facilitating efficient model adaptation.

Furthermore, the dynamic adjustment of sequence lengths during inference exemplifies the adaptability of model compression techniques in optimizing efficiency while minimizing accuracy loss. This method demonstrates the potential of dynamic configurations in achieving superior accuracy-speedup trade-offs [8].

These innovative approaches collectively underscore the evolving landscape of model compression research, which is crucial for optimizing transformer models used in natural language processing

(NLP) and computer vision (CV). By employing diverse strategies such as pruning, quantization, knowledge distillation, and modularization, researchers aim to significantly reduce the memory and computational demands of large-scale transformer models like BERT, T5, and BART. These methods are designed to enhance efficiency while maintaining high performance, thereby enabling the deployment of powerful transformer models on resource-constrained devices and in environments with strict latency requirements. The ongoing exploration of these techniques not only highlights their applicability across various tasks but also paves the way for future advancements in creating lightweight, adaptable, and high-performing models suitable for a wide range of applications. [58, 27, 28]

4 Applications in Edge Computing

The deployment of transformer models in edge computing involves overcoming computational limitations and meeting real-time performance requirements. Model compression techniques are crucial in enhancing model efficiency, enabling their use in resource-constrained environments. This section examines these techniques, emphasizing their role in improving inference speed and efficiency for complex algorithms on edge devices.

4.1 Enhancing Inference Speed and Efficiency

Improving inference speed and efficiency on edge devices is vital for real-time applications of transformer models, particularly in environments with limited computational resources. Techniques like pruning, quantization, and knowledge distillation are effective in reducing model size and complexity while maintaining performance. Strategies such as batching and model quantization are essential for decreasing the size and inference time of large language models (LLMs), thereby enhancing performance in edge computing where low-latency processing is critical [59, 60, 61].

Quantization methods significantly improve latency and energy efficiency for LLMs. Quantization-aware training reduces model size and computational costs, making models more suitable for edge deployment [49]. Heuristics for post-training quantization, especially for Mixture-of-Experts models, show enhanced performance over random allocation [34].

Pruning techniques also enhance inference speed. The CoFi method achieves over 10× speedups while maintaining accuracy, demonstrating structured pruning's effectiveness in reducing model size and computational demands without extensive unlabeled data [9]. Application-specific compression strategies, particularly for BERT models in tasks like Extractive QA, highlight optimized pipelines' impact on both accuracy and inference time [21].

Knowledge distillation, especially when combined with other compression techniques, improves model efficiency. DistilBERT exemplifies how knowledge distillation leads to smaller models with faster inference speeds, suitable for real-time mobile applications [29]. The Online Model Compression (OMC) framework effectively reduces memory usage and communication costs while maintaining accuracy in training large neural networks for federated learning, showcasing hybrid compression techniques' advantages [51].

Innovative approaches like TQCompressor demonstrate significant parameter reduction while preserving performance, as seen in compressing the GPT-2small model from 124 million to 81 million parameters [23]. The importance of evaluating inference speed alongside accuracy for practical model deployment is further emphasized [25].

Shallow Cross-Encoders maintain effectiveness even with CPU inference, making them viable for on-device applications [11]. Dynamic-TinyBERT shows notable improvements in inference efficiency while maintaining competitive accuracy on the SQuAD1.1 benchmark [8]. The KD-Lib library provides a unified platform for evaluating multiple model compression techniques, facilitating comparison and adoption [24].

MobileNMT's advantages include its small model size, low latency, and significant memory savings, ideal for mobile deployment [16]. The TA-DCCL method enhances inference speed and efficiency by compressing word embeddings during NLU task model training [14]. MLKD-BERT outperforms existing BERT distillation methods, effectively compressing the model while retaining competitive accuracy [17]. BERT-of-Theseus retains over 98

Strategic application of model compression techniques is essential for improving inference speed and efficiency in edge devices. Recent advancements in natural language processing (NLP) have led to techniques like Greedy-layer pruning and knowledge distillation, which significantly decrease inference times and resource consumption while maintaining high performance. Greedy-layer pruning allows for dynamic model size adjustments tailored to specific tasks, achieving a balance between speed and accuracy without additional pretraining. Furthermore, integrating adaptive optimization algorithms and model compression strategies enhances training efficiency and reduces memory usage, enabling complex NLP tasks to be executed effectively in real-time [60, 62, 39].

4.2 Addressing Latency-Accuracy Tradeoff

Balancing latency and accuracy in edge deployments is a critical challenge for transformer models, particularly in resource-limited environments. Advanced model compression techniques, including pruning, quantization, and knowledge distillation, have been pivotal in addressing this tradeoff, enabling efficient deployment without significant accuracy loss [63]. The APB method exemplifies this balance, achieving a superior accuracy/memory trade-off with notable improvements in CPU inference speed [63].

Quantization methods, particularly those utilizing low-bit precision, effectively reduce model size and enhance inference speed, crucial for maintaining performance in edge devices. Integrating quantization-aware training optimizes this process by minimizing precision loss during deployment [49], ensuring models remain performant even with reduced computational demands, thus effectively managing the latency-accuracy tradeoff.

Pruning techniques significantly contribute to optimizing this balance by removing redundant parameters, which reduces model complexity and inference time, essential for real-time applications on edge devices [9]. The CoFi method exemplifies structured pruning's potential to manage the latency-accuracy tradeoff effectively while achieving substantial speedups without sacrificing accuracy [9].

Knowledge distillation complements these strategies by transferring knowledge from larger models to smaller ones, enabling the latter to achieve comparable accuracy with lower computational requirements [29]. DistilBERT's advantages, including its smaller size and faster inference speed, highlight knowledge distillation's effectiveness in balancing latency and accuracy in edge deployments [29].

Hybrid compression techniques, integrating multiple strategies, provide a comprehensive approach to optimizing model performance in edge environments. These methods leverage the strengths of individual techniques to achieve significant reductions in model size and computational demands while maintaining high accuracy [18]. The ongoing evolution of these techniques underscores the importance of continuous innovation in managing the latency-accuracy tradeoff in edge computing scenarios.

4.3 Real-World Applications and Case Studies

The practical implementation of model compression techniques in real-world applications demonstrates their significant impact on enhancing the efficiency and performance of transformer models across various domains. The BERT2DNN framework in e-commerce search exemplifies the benefits of model compression by effectively enhancing performance, showcasing its suitability for large-scale applications [64]. This framework highlights knowledge distillation's potential in optimizing models for specific industry needs, ensuring efficient operation under real-world constraints.

The MS MARCO document ranking dataset serves as a benchmark for evaluating simplified TinyBERT models, which utilize knowledge distillation to achieve competitive performance with reduced computational demands [65]. This dataset provides a robust platform for assessing the scalability and adaptability of compressed models across diverse NLP tasks, reinforcing their applicability in information retrieval systems.

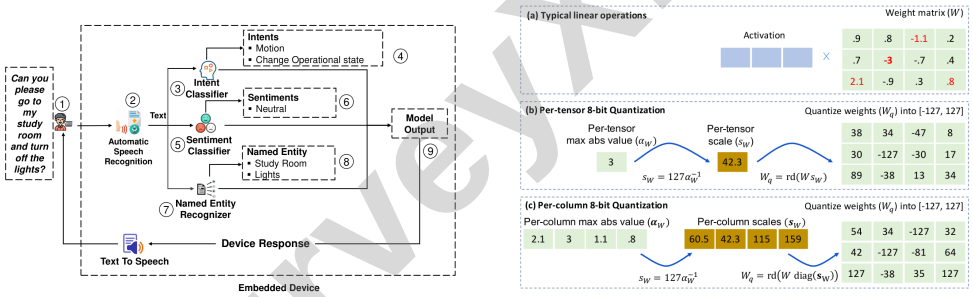
In NLP applications, the TT-embedding technique has shown efficacy in significantly reducing parameters in embedding layers while maintaining or enhancing model accuracy [66]. This method underscores the potential of tensorized approaches for deploying large-scale models in environments with limited computational resources.

Empirical studies on low-precision techniques, using datasets like CIFAR10 for image classification and Speech Commands for keyword spotting, illustrate the versatility of model compression methods across different tasks [67]. These use cases highlight the adaptability of compression strategies in optimizing model performance for a wide range of applications, from visual recognition to auditory processing.

SDQ, a technique combining structured sparsity and quantization, is particularly beneficial for applications requiring real-time inference from large language models, such as conversational agents and automated content generation [19]. This approach demonstrates the practical benefits of model compression in enhancing the responsiveness and efficiency of AI-driven systems in dynamic environments.

Data-free knowledge distillation methods, such as DFKD-T3, applied to benchmark datasets for text classification and named entity recognition, exemplify innovative compression techniques maintaining high performance without relying on original training data [7]. These experiments showcase compressed models' capability to deliver robust performance across various linguistic tasks, ensuring relevance in privacy-sensitive applications.

Moreover, the method developed by Takamoto et al. is particularly pertinent for applications demanding precise regression predictions, such as age estimation and gaze tracking, reinforcing the importance of model compression in enhancing predictive analytics' accuracy and efficiency [36]. These case studies collectively underscore the transformative impact of model compression techniques in optimizing transformer models for real-world applications, ensuring effective deployment across diverse industries and use cases.



(a) A flowchart illustrating a conversational assistant system[68]

(b) Quantization of Weight Matrices in Neural Networks[57]

Figure 5: Examples of Real-World Applications and Case Studies

As shown in Figure 5, real-world applications and case studies in edge computing provide valuable insights into technology's practical deployment and optimization. One example is a flowchart of a conversational assistant system, illustrating the process from a user's spoken request to executing an action, such as turning off the lights in a study room. This involves critical steps like speech recognition and intent classification, determining the user's intention. Another example explores the quantization of weight matrices in neural networks, a technique crucial for enhancing computational efficiency and reducing resource consumption in edge devices. The image outlines three distinct quantization methods—per-tensor, per-column, and a combination of both—aimed at compressing weight matrices into a limited range of values, optimizing neural network performance in resource-constrained environments. These examples underscore diverse applications and innovative strategies employed in edge computing to address real-world challenges effectively [68, 57].

4.4 Challenges in Edge Deployment

Deploying compressed transformer models in edge environments presents several challenges, particularly in balancing the trade-off between reducing model size and maintaining acceptable accuracy levels. While pruning techniques effectively reduce model size, they often struggle to preserve performance, which is crucial for real-time applications on edge devices [44]. SparseBERT demonstrates higher compression rates while maintaining performance, yet achieving such balance highlights the complexities of deploying these models in resource-constrained settings [62].

The heterogeneity of edge devices introduces significant complexity, as these devices vary greatly in computational power, memory capacity, and energy availability, hindering the deployment of resource-intensive deep neural networks (DNNs). This diversity necessitates advanced model compression techniques, such as pruning and quantization, to optimize DNNs for performance on limited-resource devices. Additionally, innovative mapping optimizers and incremental learning methods are essential for enhancing inference speed and maintaining model accuracy across different edge environments [59, 69, 70, 33, 61]. This variability requires adaptable compression techniques tailored to each deployment environment's specific constraints. Furthermore, the dynamic nature of edge environments, characterized by fluctuating network conditions and computational resources, demands robust and flexible models capable of handling variations without significant performance degradation.

The integration of compressed models with existing edge infrastructure poses significant technical challenges. Ensuring compatibility and seamless operation with diverse hardware and software ecosystems is crucial for successful deployment. Managing deployments in edge environments is further complicated by the need for ongoing updates and maintenance of large language models (LLMs) to mitigate security vulnerabilities and enhance performance. This complexity arises from the models' substantial resource demands and the necessity for fine-tuning on local datasets, often requiring memory-efficient techniques and model compression to reduce energy consumption and operational costs. Additionally, LLM inference's unique characteristics, including considerable size and computational requirements, necessitate tailored optimization strategies, such as batching and quantization, to maximize throughput while addressing latency and accuracy concerns [70, 59, 33, 61].

Privacy and data security are also critical considerations in edge deployments. Compressed models must be designed to handle sensitive data locally, minimizing the need for data transmission to centralized servers. To maintain user privacy while ensuring efficient model inference, comprehensive encryption methods and stringent data handling protocols must be implemented. These measures should effectively safeguard sensitive information without hindering complex models' speed and performance, particularly in resource-constrained environments where model compression techniques, such as pruning, can significantly reduce computational demands while preserving output quality [32, 44, 71].

To effectively tackle the challenges of deploying deep neural networks (DNNs) in edge computing environments, continuous innovation in model compression techniques—such as pruning and quantization—must be pursued, alongside a comprehensive strategy ensuring seamless integration of these optimized models into edge computing ecosystems. This involves enhancing runtime performance and accuracy while addressing energy efficiency and resource constraints inherent in edge devices [72, 25, 73, 61]. By overcoming these obstacles, the deployment of transformer models in edge environments can be optimized to deliver high performance while maintaining resource efficiency and data security.

5 Challenges and Future Directions

The landscape of transformer model compression is fraught with challenges that demand a nuanced understanding of optimization techniques. Key issues revolve around maintaining accuracy and robustness, as methods like pruning, quantization, and knowledge distillation can impact model performance. This section explores the critical challenges in preserving model integrity amidst these compression strategies.

5.1 Accuracy and Robustness Challenges

Maintaining accuracy and robustness post-compression remains a significant hurdle due to trade-offs inherent in methods such as pruning, quantization, and knowledge distillation. While effective in reducing model size, these techniques can degrade performance, requiring careful optimization. Pruning may compromise both general-purpose and task-specific knowledge, leading to subpar performance compared to smaller dense models [6]. Excessive pruning or aggressive quantization often results in accuracy losses, particularly in models with fewer parameters, highlighting the limited expressiveness of compressed student models [14].

Quantization, especially at low bit-widths, can cause significant accuracy losses. The interaction between quantization and knowledge distillation adds complexity, as distillation's regularization

effect may diminish quantized networks' performance [49]. Existing quantization methods often require manual tuning and lack fine-grained subgroup-wise support, complicating their application [10]. Experiments show that pruning and quantization can affect adversarial sample transferability, posing security vulnerabilities [12].

Challenges in knowledge distillation include effectively transferring task-relevant signals. Distilled models' performance often hinges on the teacher model's quality; a weak teacher can produce an underperforming student [36]. The capacity gap between teacher and student models necessitates effective distillation strategies [35]. Activation record choice also significantly impacts compressed models' accuracy [8].

The robustness of compressed models must address fairness and explainability post-compression. Biases and toxicity introduced during compression demand innovative solutions to balance efficiency and performance [37]. The high computational complexity of large models remains a challenge, as many methods require substantial resources during training, making them impractical for some applications [16]. Current studies often overlook the trade-offs between model size and performance, with many compressed models underperforming compared to their uncompressed counterparts [6].

Addressing these challenges involves refining compression techniques and exploring new methodologies to ensure robust and accurate models for diverse applications. Developing greener and more efficient methods holds promise for maintaining much of the original model's accuracy, although some quantization settings may still result in minor performance drops [14]. As research advances, integrating collaborative learning approaches, such as combining knowledge distillation with mutual learning, offers promising pathways for enhancing performance and efficiency. However, methods like BERT-of-Theseus may face difficulties when predecessor and successor module architectures differ significantly, potentially affecting performance [13].

5.2 Optimization of Compression Techniques

Optimizing compression techniques for transformer models is crucial for enhancing computational efficiency while preserving performance across diverse tasks. Future research should prioritize refining quantization schemes to maintain model accuracy at low bit depths, addressing trade-offs between compression levels and predictive accuracy [74]. This includes investigating the application of LCQ to multi-modal models and optimizing its methods to improve performance and applicability [10]. Additionally, exploring the generalization of advanced hierarchical optimization (AHO) across various models and datasets, and integrating it with other optimization frameworks, could further enhance capabilities [75].

Pruning strategies require further exploration, particularly the application of CoFi for upstream pruning to create task-agnostic models, alongside investigating other potential optimizations in the pruning process [9]. The development of tuning-free compression techniques and hybrid approaches that combine multiple methods could yield optimal results in efficiency and performance [1]. Future research could also focus on optimizing the distillation process, indicating potential improvements in compression techniques [29].

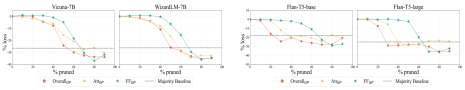
Integrating knowledge distillation with mutual learning presents a promising avenue for enhancing performance and efficiency, addressing challenges posed by current methods that either necessitate complete re-training of the compressed model or are constrained by architectural limitations [15]. Additionally, exploring prolonged CrAM training could improve model performance and robustness while investigating methods to reduce computational complexity in the CrAM update [12].

Frameworks like KD-Lib serve as valuable resources for evaluating multiple model compression techniques, although they may not encompass all possible algorithms or variations, potentially limiting their applicability in specific scenarios [24]. By pursuing these research directions, advancements in compression techniques will collectively contribute to the development of optimized models that are efficient and broadly applicable.

As shown in Figure 6, the challenges and future directions in optimizing compression techniques are illuminated through mathematical models and empirical comparisons. The first example, represented as a mathematical equation, emphasizes the complexity inherent in optimizing compression strategies. This equation, expressed in LaTeX, illustrates a summation over a range of variables, highlighting the intricate balance between different components necessary for effective compression. The second

$$\mathcal{L}(\phi) = \underbrace{\sum_{i=1}^{N} \mathbb{E}_{\mathbf{w}\sim q_\phi(\mathbf{w})} [\log p(y_n|x_n, \mathbf{w})]}_{L_{\mathcal{D}}(\phi)} - D_{KL}(q_\phi(\mathbf{w}) || p(\mathbf{w}))$$

(a) The image shows a mathematical equation.[76]



(b) Comparison of Loss Reduction Across Different Pruning Strategies for Various Models[77]

Figure 6: Examples of Optimization of Compression Techniques

example provides a comparative analysis of loss reduction across various pruning strategies applied to models such as Vicuna-7B, WizardLM-7B, and Flan-T5 variants. This comparison, visualized through a graph, reveals the impact of different pruning strategies on model performance, with the x-axis indicating the percentage of weights pruned and the y-axis showing the resultant percentage of loss reduction. Together, these examples underscore ongoing challenges in refining compression techniques and suggest potential avenues for future exploration, including the development of more sophisticated mathematical models and empirical evaluations of diverse pruning strategies to enhance model efficiency and performance [76, 77].

5.3 Adaptability and Generalization

The adaptability and generalization of compressed models across diverse tasks and datasets are crucial for effective deployment in various applications. Future research should prioritize mixed-precision approaches that optimize quantization-aware pruning (QAP) methods, enhancing model adaptability and validation across different contexts [78]. Exploring heterogeneous sparsity and quantization schemes, along with their interactions, could further optimize model performance, ensuring robustness and adaptability in varying computational environments [22].

Developing advanced techniques for knowledge distillation and further optimizations for quantization is essential to enhance model performance on highly constrained devices, such as edge computing platforms [4]. Additionally, exploring alternative methods for distilling intermediate representations could improve the adaptability of compressed models across various tasks and datasets, facilitating applications in diverse linguistic and computational contexts [79].

Improving quantization methods for ultra-low precision and developing frameworks that support lower precision computations are critical for advancing the generalization capabilities of compressed models, particularly in resource-limited scenarios [67]. Future research could also focus on optimizing vocabulary selection and combining these methods with other distillation techniques to enhance adaptability across different NLP tasks [80].

The adaptability of methods like TA-DCCL is demonstrated by their potential application to various NLP tasks beyond natural language understanding (NLU), including machine translation and language modeling, indicating their versatility across different domains [14]. Enhancing algorithm robustness against network fluctuations and exploring applicability in areas such as real-time data analytics could further extend the generalization capabilities of compressed models [81].

Finally, investigating the application of Theseus Compression to other neural network types and integrating this approach with dynamic acceleration methods could enhance efficiency and adaptability [13]. Future work should also explore enhancements for multilingual translation and improve compatibility with various AI accelerators beyond ARM CPUs, ensuring compressed models can adapt to a wide range of hardware and software environments [16]. Addressing these research directions can significantly enhance the adaptability and generalization of compressed models, ensuring their effectiveness in diverse and dynamic environments.

5.4 Resource Constraints and Deployment

Deploying compressed transformer models in resource-constrained environments poses significant challenges, particularly regarding processor, memory, and battery limitations of embedded devices. Balancing computational efficiency with accuracy is critical, as these models often execute complex natural language processing tasks under stringent resource constraints [6]. A major hurdle is the extensive retraining often necessary to maintain model performance post-compression, which can be

resource-intensive and demand substantial computational power and time, posing feasibility issues for certain deployment scenarios [15].

The computational overhead associated with certain optimization methods, such as Sharpness-Aware Minimization (SAM), complicates deployment by requiring more resources and time compared to standard techniques [49]. Additionally, the compatibility of compressed models with existing infrastructure is critical, as the fixed data precision assumed by some methods may not fully capture the benefits of lower precision computations across all scenarios, potentially limiting their applicability [35].

Innovative approaches like Quantization-Aware Training (QAT) offer promising solutions to these challenges by addressing memory constraints that previously hindered such operations. However, these methods may still encounter challenges in extremely low-rank settings, potentially leading to performance degradation [49]. Furthermore, multiplexing with model compression can lead to information leakage between different instances, raising privacy concerns if used in public APIs.

Future research should focus on developing more efficient algorithms for pruning and quantization that maintain accuracy while further reducing computational requirements. Enhancing distillation techniques, optimizing model adaptation, and improving robustness against dataset shifts are also crucial areas for exploration [6]. Additionally, exploring hybrid models and independent layer compression techniques could fully leverage transformer architectures for efficient compression [15].

By addressing these challenges, the deployment of compressed transformer models can be optimized to deliver high performance while adhering to the resource constraints inherent in edge and embedded computing environments. The cost-effectiveness of methods such as BERT2DNN, which distills knowledge from BERT into simpler feed-forward networks, significantly lowers serving costs—achieving a latency that is 150 times faster than BERT-Base and 15 times faster than TinyBERT—making advanced machine learning techniques more accessible to organizations with limited resources. This democratization of technology underscores the potential for wider application across various sectors, even for those constrained by budgetary limitations [82, 64, 83, 62, 38].

5.5 Integration with Emerging Technologies

Integrating model compression techniques with emerging technologies is crucial for advancing the efficiency and applicability of transformer models across various domains. As the demand for more efficient AI models grows, leveraging advanced technologies can significantly optimize the performance and deployment of compressed models. Future research should focus on exploring the application of Modular Transformers to other architectures and larger models, as well as investigating further optimizations for different resource constraints [58]. Additionally, optimizing compression algorithms and exploring the application of Online Model Compression (OMC) in different federated learning scenarios could enhance model efficiency and adaptability [51].

The development of specialized pruning techniques for text-to-image models and exploring the implications of information encoding mechanisms in these models present promising avenues for future research [32]. Further optimizations in the clustering process and the applicability of these methods to a wider range of model architectures should also be explored [84]. Moreover, optimizing generative models further and applying the framework to additional NLP tasks and settings could enhance the versatility and performance of compressed models [7].

Enhancements to the Teacher Outlier Rejection (TOR) Loss and its application to more complex regression tasks or real-world datasets could improve the robustness and accuracy of compressed models [36]. Integrating dynamic sequence-length with sparsity and low-bit quantization methods offers another promising research direction to further optimize performance [8]. The scalability of Progressive Knowledge Distillation (Pro-KD) to larger models and datasets, as well as its applicability to other domains, should be explored to enhance its effectiveness and generalizability [37].

Further research could focus on optimizing the training process and extending methods like MLKD-BERT to other types of tasks beyond natural language understanding, thereby broadening the scope and impact of model compression techniques [17]. Additionally, developing more advanced model compression techniques, scaling up existing methods from smaller models to large language models (LLMs), improving the efficiency of compressed models during deployment, and integrating AutoML techniques to reduce reliance on manual design are critical areas for future exploration [6].

By integrating model compression techniques—such as weight quantization, parameter and network pruning, and knowledge distillation—with emerging technologies, researchers can create highly efficient and adaptable AI models that not only reduce size and computational requirements but also maintain high performance. This integration is crucial for meeting the demands of modern AI applications, particularly in resource-constrained environments like edge devices, where the proliferation of connected devices necessitates models that can operate effectively with limited compute capacity and power availability. Furthermore, leveraging optimized data augmentation strategies can enhance model performance during the compression process, allowing for tailored approaches that align with the specific characteristics of various model sizes [25, 85].

6 Conclusion

Model compression techniques are pivotal in enhancing the efficiency of transformer models for NLP, particularly in environments with limited computational resources. Techniques such as knowledge distillation, pruning, and quantization are central to reducing the size and complexity of models while maintaining their performance. Knowledge distillation, in particular, has proven effective in improving model efficiency, as demonstrated by various frameworks that facilitate the implementation and assessment of diverse compression strategies.

Pruning, especially when applied in a stage-wise manner, has shown significant improvements in accuracy and effectiveness, achieving notable results in deep neural networks. The combination of pruning with hybrid compression frameworks offers a balanced approach to managing model size and accuracy, which is crucial in resource-limited scenarios. Quantization, including extreme quantization methods, provides viable solutions for achieving substantial compression ratios without compromising accuracy.

Emerging methods like FITCompress and MaQD illustrate the progress in model compression, effectively balancing efficiency and inference accuracy. Techniques involving tensor decompositions and permutations, as seen in advanced compressors, suggest promising avenues for future research in neural network compression. Additionally, the use of shallow Cross-Encoders for low-latency applications highlights the potential for CPU-only inference, enhancing retrieval effectiveness in practical settings.

The strategic application of model compression techniques is essential for the successful deployment of transformer models in NLP tasks. These innovations enable the execution of complex natural language processing functions efficiently and in real-time, meeting the demands of modern applications while optimizing resource use. Ongoing development and integration of these techniques are crucial for advancing the field, potentially leading to significant enhancements in model efficiency and applicability.

References

[1] Gousia Habib, Tausifa Jan Saleem, and Brejesh Lall. Knowledge distillation in vision transformers: A critical review, 2024.

[2] Jing Jin, Cai Liang, Tiancheng Wu, Liqin Zou, and Zhiliang Gan. Kdlsq-bert: A quantized bert combining knowledge distillation with learned step size quantization, 2021.

[3] Yao Qiang, Supriya Tumkur Suresh Kumar, Marco Brocanelli, and Dongxiao Zhu. Adversarially robust and explainable model compression with on-device personalization for text classification, 2021.

[4] Thanaphon Suwannaphong, Ferdian Jovan, Ian Craddock, and Ryan McConville. Optimising tinyml with quantization and distillation of transformer and mamba models for indoor localisation on edge devices, 2024.

[5] Andrei Tomut, Saeed S. Jahromi, Abhijoy Sarkar, Uygar Kurt, Sukhbinder Singh, Faysal Ishtiaq, Cesar Muñoz, Prabdeep Singh Bajaj, Ali Elborady, Gianni del Bimbo, Mehrazin Alizadeh, David Montero, Pablo Martin-Ramiro, Muhammad Ibrahim, Oussama Tahiri Alaoui, John Malcolm, Samuel Mugel, and Roman Orus. Compactifai: Extreme compression of large language models using quantum-inspired tensor networks, 2024.

[6] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models, 2024.

[7] Zheyuan Bai, Xinduo Liu, Hailin Hu, Tianyu Guo, Qinghua Zhang, and Yunhe Wang. Data-free distillation of language model by text-to-text transfer, 2023.

[8] Shira Guskin, Moshe Wasserblat, Ke Ding, and Gyuwan Kim. Dynamic-tinybert: Boost tinybert's inference efficiency by dynamic sequence length, 2021.

[9] Mengzhou Xia, Zexuan Zhong, and Danqi Chen. Structured pruning learns compact and accurate models, 2022.

[10] Wen-Pu Cai, Ming-Yang Li, and Wu-Jun Li. Lcq: Low-rank codebook based quantization for large language models, 2025.

[11] Aleksandr V. Petrov, Sean MacAvaney, and Craig Macdonald. Shallow cross-encoders for low-latency retrieval, 2024.

[12] Alexandra Peste, Adrian Vladu, Eldar Kurtic, Christoph H. Lampert, and Dan Alistarh. Cram: A compression-aware minimizer, 2023.

[13] Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. Bert-of-theseus: Compressing bert by progressive module replacing, 2020.

[14] Kanthashree Mysore Sathyendra, Samridhi Choudhary, and Leah Nicolich-Henkin. Extreme model compression for on-device natural language understanding, 2020.

[15] Wujie Sun, Defang Chen, Jiawei Chen, Yan Feng, Chun Chen, and Can Wang. Knowledge translation: A new pathway for model compression, 2024.

[16] Ye Lin, Xiaohui Wang, Zhexi Zhang, Mingxuan Wang, Tong Xiao, and Jingbo Zhu. Mobilenmt: Enabling translation in 15mb and 30ms, 2023.

[17] Ying Zhang, Ziheng Yang, and Shufan Ji. Mlkd-bert: Multi-level knowledge distillation for pre-trained language models, 2024.

[18] Ini Oguntola, Subby Olubeko, and Christopher Sweeney. Slimnets: An exploration of deep model compression and acceleration, 2018.

[19] Geonhwa Jeong, Po-An Tsai, Stephen W. Keckler, and Tushar Krishna. Sdq: Sparse decomposed quantization for llm inference, 2024.

[20] Xiaoxia Wu, Zhewei Yao, Minjia Zhang, Conglong Li, and Yuxiong He. Extreme compression for pre-trained transformers made simple and efficient, 2022.

[21] Rohit Raj Rai, Angana Borah, and Amit Awekar. Application specific compression of deep learning models, 2024.

[22] Simla Burcu Harma, Ayan Chakraborty, Elizaveta Kostenok, Danila Mishin, Dongho Ha, Babak Falsafi, Martin Jaggi, Ming Liu, Yunho Oh, Suvinay Subramanian, and Amir Yazdanbakhsh. Effective interplay between sparsity and quantization: From theory to practice, 2025.

[23] V. Abronin, A. Naumov, D. Mazur, D. Bystrov, K. Tsarova, Ar. Melnikov, I. Oseledets, S. Dolgov, R. Brasher, and M. Perelshtein. Tqcompressor: improving tensor decomposition methods in neural networks via permutations, 2024.

[24] Het Shah, Avishree Khare, Neelay Shah, and Khizir Siddiqui. Kd-lib: A pytorch library for knowledge distillation, pruning and quantization, 2020.

[25] Aayush Saxena, Arit Kumar Bishwas, Ayush Ashok Mishra, and Ryan Armstrong. Comprehensive study on performance evaluation and optimization of model compression: Bridging traditional deep learning and large language models, 2024.

[26] Arhum Ishtiaq, Sara Mahmood, Maheen Anees, and Neha Mumtaz. Model compression, 2021.

[27] Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Hassan Sajjad, Preslav Nakov, Deming Chen, and Marianne Winslett. Compressing large-scale transformer-based models: A case study on bert, 2021.

[28] Yehui Tang, Yunhe Wang, Jianyuan Guo, Zhijun Tu, Kai Han, Hailin Hu, and Dacheng Tao. A survey on transformer compression, 2024.

[29] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[30] Akshay Kulkarni, Navid Panchi, Sharath Chandra Raparthy, and Shital Chiddarwar. Data efficient stagewise knowledge distillation, 2020.

[31] J. S. McCarley, Rishav Chakravarti, and Avirup Sil. Structured pruning of a bert-based question answering model, 2021.

[32] Samarth N Ramesh and Zhixue Zhao. Efficient pruning of text-to-image models: Insights from pruning stable diffusion, 2024.

[33] Kaiqi Zhao, Yitao Chen, and Ming Zhao. Enabling deep learning on edge devices through filter pruning and knowledge transfer, 2022.

[34] Pingzhi Li, Xiaolong Jin, Yu Cheng, and Tianlong Chen. Examining post-training quantization for mixture-of-experts: A benchmark, 2024.

[35] Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. Knowledge distillation beyond model compression, 2020.

[36] Makoto Takamoto, Yusuke Morishita, and Hitoshi Imaoka. An efficient method of training small models for regression problems with knowledge distillation, 2020.

[37] Mehdi Rezagholizadeh, Aref Jafari, Puneeth Salad, Pranav Sharma, Ali Saheb Pasand, and Ali Ghodsi. Pro-kd: Progressive distillation by following the footsteps of the teacher, 2021.

[38] Shaoyi Huang, Ning Liu, Yueying Liang, Hongwu Peng, Hongjia Li, Dongkuan Xu, Mimi Xie, and Caiwen Ding. An automatic and efficient bert pruning for edge ai systems, 2022.

[39] David Peer, Sebastian Stabinger, Stefan Engl, and Antonio Rodriguez-Sanchez. Greedy-layer pruning: Speeding up transformer models for natural language processing, 2022.

[40] Jangho Kim, Simyung Chang, and Nojun Kwak. Pqk: Model compression via pruning, quantization, and knowledge distillation, 2021.

[41] Se Jung Kwon, Dongsoo Lee, Byeongwook Kim, Parichay Kapoor, Baeseong Park, and Gu-Yeon Wei. Structured compression by weight encryption for unstructured pruning and quantization, 2020.

[42] Minh Tri Lê, Pierre Wolinski, and Julyan Arbel. Efficient neural networks for tiny machine learning: A comprehensive review, 2023.

[43] Azade Nova, Hanjun Dai, and Dale Schuurmans. Gradient-free structured pruning with unlabeled data, 2023.

[44] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression, 2017.

[45] Duong H. Le, Trung-Nhan Vo, and Nam Thoai. Paying more attention to snapshots of iterative pruning: Improving model compression via ensemble distillation, 2020.

[46] Dongsoo Lee and Byeongwook Kim. Retraining-based iterative weight quantization for deep neural networks, 2018.

[47] He Li, Jianhang Hong, Yuanzhuo Wu, Snehal Adbol, and Zonglin Li. Continuous approximations for improving quantization aware training of llms, 2024.

[48] Jangho Kim, Yash Bhalgat, Jinwon Lee, Chirag Patel, and Nojun Kwak. Qkd: Quantization-aware knowledge distillation, 2019.

[49] Zi Yang, Samridhi Choudhary, Siegfried Kunzmann, and Zheng Zhang. Quantization-aware and tensor-compressed training of transformers for natural language understanding, 2023.

[50] Xiaoxia Wu, Cheng Li, Reza Yazdani Aminabadi, Zhewei Yao, and Yuxiong He. Understanding int4 quantization for transformer models: Latency speedup, composability, and failure cases, 2023.

[51] Tien-Ju Yang, Yonghui Xiao, Giovanni Motta, Françoise Beaufays, Rajiv Mathews, and Mingqing Chen. Online model compression for federated learning with large models, 2022.

[52] Sean I. Young. Foundations of large language model compression – part 1: Weight quantization, 2024.

[53] Changsheng Zhao, Ting Hua, Yilin Shen, Qian Lou, and Hongxia Jin. Automatic mixed-precision quantization search of bert, 2021.

[54] Angie Boggust, Venkatesh Sivaraman, Yannick Assogba, Donghao Ren, Dominik Moritz, and Fred Hohman. Compress and compare: Interactively evaluating efficiency and behavior across ml model compression experiments, 2024.

[55] Ji Xin, Raphael Tang, Zhiying Jiang, Yaoliang Yu, and Jimmy Lin. Building an efficiency pipeline: Commutativity and cumulativeness of efficiency operators for transformers, 2022.

[56] Dongqi Wang, Shengyu Zhang, Zhipeng Di, Xin Lin, Weihua Zhou, and Fei Wu. A novel architecture slimming method for network pruning and knowledge distillation, 2022.

[57] Xiaokai Wei, Sujan Gonugondla, Wasi Ahmad, Shiqi Wang, Baishakhi Ray, Haifeng Qian, Xiaopeng Li, Varun Kumar, Zijian Wang, Yuchen Tian, Qing Sun, Ben Athiwaratkun, Mingyue Shang, Murali Krishna Ramanathan, Parminder Bhatia, and Bing Xiang. Greener yet powerful: Taming large code generation models with quantization, 2023.

[58] Wangchunshu Zhou, Ronan Le Bras, and Yejin Choi. Modular transformers: Compressing transformers into modularized layers for flexible efficient inference, 2023.

[59] Xinyuan Zhang, Jiang Liu, Zehui Xiong, Yudong Huang, Gaochang Xie, and Ran Zhang. Edge intelligence optimization for large language model inference with batching and quantization, 2024.

[60] Taiyuan Mei, Yun Zi, Xiaohan Cheng, Zijun Gao, Qi Wang, and Haowei Yang. Efficiency optimization of large-scale language models based on deep learning in natural language processing tasks, 2024.

[61] Bailey J. Eccles, Leon Wong, and Blesson Varghese. Rapid deployment of dnns for edge computing via structured pruning at initialization, 2024.

[62] Dongkuan Xu, Ian E. H. Yen, Jinxi Zhao, and Zhibin Xiao. Rethinking network pruning – under the pre-train and fine-tune paradigm, 2022.

[63] Franco Maria Nardini, Cosimo Rulli, Salvatore Trani, and Rossano Venturini. Neural network compression using binarization and few full-precision weights, 2023.

[64] Yunjiang Jiang, Yue Shang, Ziyang Liu, Hongwei Shen, Yun Xiao, Wei Xiong, Sulong Xu, Weipeng Yan, and Di Jin. Bert2dnn: Bert distillation with massive unlabeled data for online e-commerce search, 2020.

[65] Xuanang Chen, Ben He, Kai Hui, Le Sun, and Yingfei Sun. Simplified tinybert: Knowledge distillation for document retrieval, 2021.

[66] Oleksii Hrinchuk, Valentin Khrulkov, Leyla Mirvakhabova, Elena Orlova, and Ivan Oseledets. Tensorized embedding layers for efficient model compression, 2020.

[67] Shaojie Zhuo, Hongyu Chen, Ramchalam Kinattinkara Ramakrishnan, Tommy Chen, Chen Feng, Yicheng Lin, Parker Zhang, and Liang Shen. An empirical study of low precision quantization for tinyml, 2022.

[68] Souvika Sarkar, Mohammad Fakhruddin Babar, Md Mahadi Hassan, Monowar Hasan, and Shubhra Kanti Karmaker Santu. Processing natural language on embedded devices: How well do transformer models perform?, 2024.

[69] Murray Kornelsen. Low-latency bert inference for heterogeneous multi-processor edge devices. 2023.

[70] Yanjie Dong, Haijun Zhang, Chengming Li, Song Guo, Victor C. M. Leung, and Xiping Hu. Fine-tuning and deploying large language models over edges: Issues and approaches, 2024.

[71] Mengnan Du, Subhabrata Mukherjee, Yu Cheng, Milad Shokouhi, Xia Hu, and Ahmed Hassan Awadallah. Robustness challenges in model distillation and pruning for natural language understanding, 2023.

[72] Canwen Xu and Julian McAuley. A survey on model compression and acceleration for pretrained language models, 2022.

[73] Vinu Joseph, Shoaib Ahmed Siddiqui, Aditya Bhaskara, Ganesh Gopalakrishnan, Saurav Muralidharan, Michael Garland, Sheraz Ahmed, and Andreas Dengel. Going beyond classification accuracy metrics in model compression, 2021.

[74] Grant P. Strimel, Kanthashree Mysore Sathyendra, and Stanislav Peshterliev. Statistical model compression for small-footprint natural language understanding, 2018.

[75] Xinyi Wang, Haiqin Yang, Liang Zhao, Yang Mo, and Jianping Shen. Refbert: Compressing bert by referencing to pre-computed representations, 2021.

[76] Marco Federici, Karen Ullrich, and Max Welling. Improved bayesian compression, 2017.

[77] Satya Sai Srinath Namburi, Makesh Sreedhar, Srinath Srinivasan, and Frederic Sala. The cost of compression: Investigating the impact of compression on parametric knowledge in language models, 2023.

[78] Benjamin Hawks, Javier Duarte, Nicholas J. Fraser, Alessandro Pappalardo, Nhan Tran, and Yaman Umuroglu. Ps and qs: Quantization-aware pruning for efficient low latency neural network inference, 2021.

[79] Alina Kolesnikova, Yuri Kuratov, Vasily Konovalov, and Mikhail Burtsev. Knowledge distillation of russian language models with reduction of vocabulary, 2022.

[80] Sanqiang Zhao, Raghav Gupta, Yang Song, and Denny Zhou. Extremely small bert models from mixed-vocabulary training, 2021.

[81] Shaokun Zhang, Xiawu Zheng, Chenyi Yang, Yuchao Li, Yan Wang, Fei Chao, Mengdi Wang, Shen Li, Jun Yang, and Rongrong Ji. You only compress once: Towards effective and elastic bert compression via exploit-explore stochastic nature gradient, 2021.

[82] Jin Xu, Xu Tan, Renqian Luo, Kaitao Song, Jian Li, Tao Qin, and Tie-Yan Liu. Nas-bert: Task-agnostic and adaptive-size bert compression with neural architecture search, 2021.

[83] Cheng Chen, Yichun Yin, Lifeng Shang, Zhi Wang, Xin Jiang, Xiao Chen, and Qun Liu. Extract then distill: Efficient and effective task-agnostic bert distillation, 2021.

[84] Madhumitha Sakthi, Niranjan Yadla, and Raj Pawate. Deep learning model compression using network sensitivity and gradients, 2022.

[85] Muzhou Yu, Linfeng Zhang, and Kaisheng Ma. Revisiting data augmentation in model compression: An empirical and comprehensive study, 2023.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.