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 enhance the handling of complex linguistic tasks efficiently [1]. Large-scale pre-trained models, such as BERT, have set new performance benchmarks across various tasks, including sentiment analysis, machine translation, and question answering, by leveraging self-attention for parallel data processing and high accuracy in relevance prediction [2].

However, deploying these models in resource-constrained environments poses significant challenges due to their high computational and memory requirements, particularly in edge computing scenarios involving mobile devices and embedded systems [3]. The large memory footprint of models like BERT complicates their practicality, resulting in increased latency in both local and cloud systems. Moreover, the substantial serving costs and environmental impact associated with these models highlight the urgent need for sustainable AI practices that reduce energy consumption and carbon emissions.

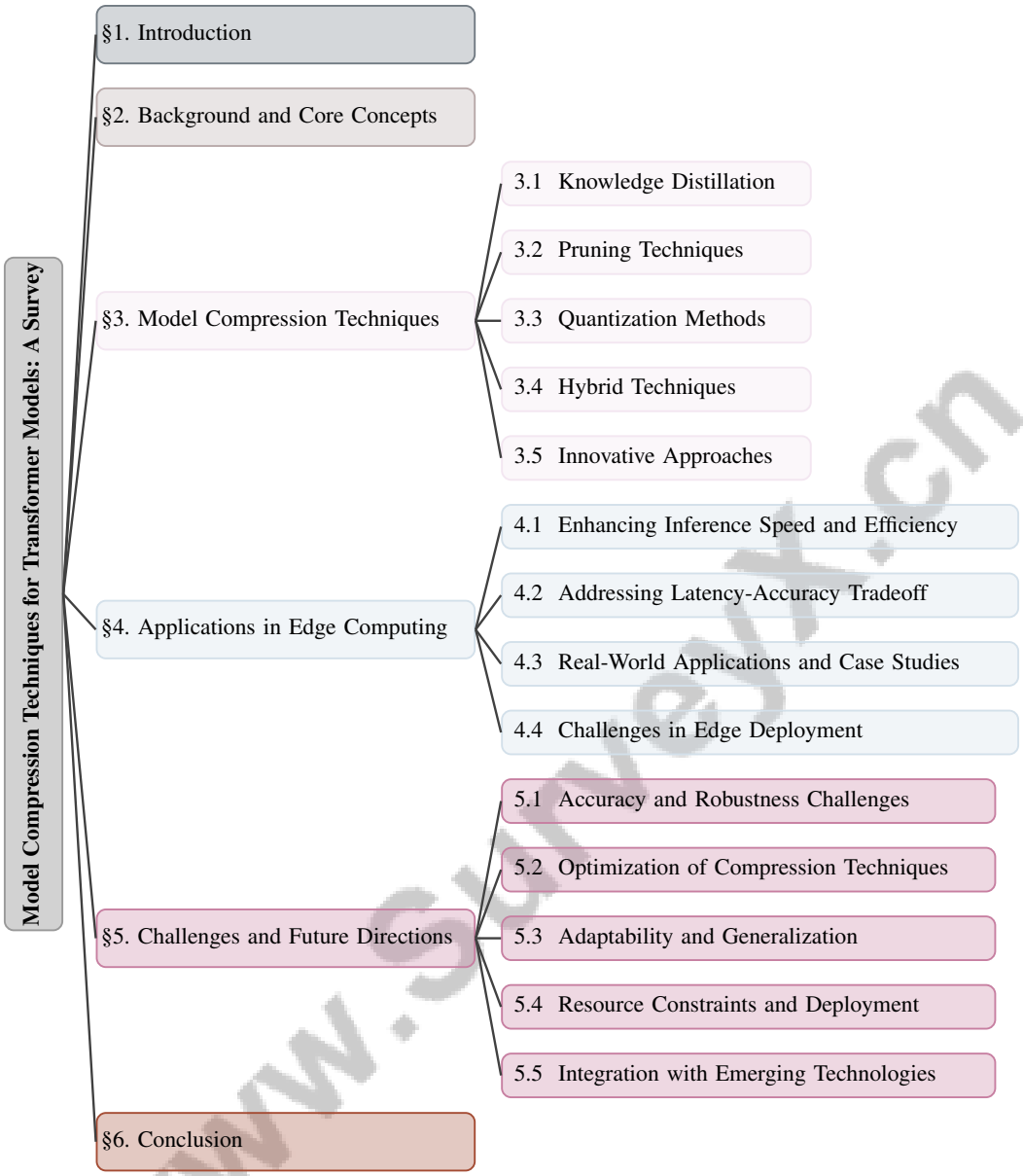


Figure 1: chapter structure

To address these challenges, model compression techniques, including knowledge distillation, have been developed to shrink models like BERT while retaining their performance, which is crucial for their deployment in resource-efficient applications [4]. The evolution of large language models (LLMs), such as ChatGPT and LlaMA, further underscores the necessity for optimizing model architectures to sustain their significant impact in the rapidly evolving field of generative AI [5]. Recent advancements in model compression specifically target LLMs to tackle their size and computational demands in resource-limited environments [6]. Innovative approaches like data-free knowledge distillation (DFKD) are emerging to mitigate the reliance on original training data, which may be unavailable due to privacy or regulatory issues [7]. These efforts are vital for ensuring the accessibility and sustainability of transformer models across diverse deployment contexts.

1.2 Challenges of Computational Demands

Deploying transformer models in resource-constrained environments, such as smartphones and IoT devices, presents significant challenges due to their high computational and memory demands.

Characterized by millions of parameters, these models complicate real-time applications on edge devices with limited resources [8]. The increasing size of large language models (LLMs) exacerbates these challenges, leading to heightened resource consumption and longer training times [6].

Existing model compression techniques, such as pruning and distillation, often struggle to achieve an optimal balance between speed and accuracy, particularly in edge computing scenarios [9]. Naive weight quantization, especially at low bit-widths, can cause significant accuracy degradation, complicating efforts to maintain model performance [10]. The limitations of Cross-Encoders in efficiently scoring numerous documents within constrained latency windows further emphasize the need for methods that enhance user satisfaction and energy efficiency in retrieval systems [11].

The compute-bound nature of inference in large language models, particularly with large batch sizes or long sequences, results in considerable latency from intensive matrix calculations [12]. This reliance on substantial computational resources necessitates innovative compression techniques that effectively balance computational efficiency and model performance without sacrificing accuracy [8]. Addressing these computational challenges requires developing novel strategies that maintain model capacity while reducing size and load [6]. Compressing large models like BERT to lessen their computational requirements while preserving performance across NLP tasks remains a crucial research focus [13].

1.3 Need for Model Compression Techniques

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

The practical deployment of models on mobile devices continues to be challenging due to the difficulty in balancing model size, computational efficiency, and decoding speed [16]. Approaches like MLKD-BERT improve model compression by distilling both feature-level and relation-level knowledge, enhancing efficiency and effectiveness [17]. Despite these advancements, there remains a pressing need for breakthroughs that simplify the compression process while maintaining high performance [13]. Developing innovative compression strategies is critical to optimizing performance and ensuring the full realization of transformer models' benefits across diverse deployment settings.

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 are essential in this context, significantly reducing computational and memory demands, which facilitates the deployment of models on resource-constrained devices. The ability of compressed models to maintain high accuracy while operating under limited resources is crucial for ensuring fairness and effectiveness in NLP applications across various environments [18].

Innovative approaches, such as CompactifAI, exemplify the potential of model compression to enhance performance in edge computing by employing quantum-inspired Tensor Networks, focusing on model correlations rather than merely reducing neuron count, thus improving performance without extensive retraining or architectural changes [5]. This adaptability is critical for edge devices where resource efficiency is paramount. Techniques like Sparse Decomposed Quantization (SDQ) combine structured sparsity and quantization to enhance compute and memory efficiency, reflecting ongoing efforts to tailor compression methods to 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, enabling the efficient deployment of large NLP models on resource-constrained devices by optimizing compute and memory usage [20]. Application-Specific Compression (ASC) offers another promising strategy, allowing for the creation of tailored

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

Despite these advancements, challenges remain, particularly in the interplay between sparsity and quantization, which necessitates systematic exploration to optimize efficiency and performance [22]. Furthermore, techniques like TQCompressor demonstrate the capability 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, as benchmarked by frameworks like 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 meticulously structured to provide a comprehensive analysis of model compression techniques for transformer models, particularly in edge computing environments. It begins with an introduction that establishes the significance of transformer models in NLP and outlines the computational challenges they present, especially in resource-constrained settings. This section emphasizes the critical need for model compression techniques to enhance computational efficiency.

Following the introduction, the survey delves into the background and core concepts, offering a detailed overview of transformer models and the fundamental principles of model compression, including knowledge distillation, pruning, and quantization. This foundational section prepares the reader for 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 discusses hybrid techniques that combine these methods to optimize performance and highlights innovative approaches at the forefront of model compression research.

The section dedicated to 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. Challenges encountered in deploying compressed models in edge environments are also addressed.

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

Finally, the conclusion summarizes the key points discussed, reiterating the importance of model compression techniques in enhancing 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 evaluating and combining 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 is pivotal in reducing the size and computational demands of transformer models, facilitating their deployment on resource-limited devices. Techniques such as pruning, quantization, knowledge distillation, and efficient architecture design are tailored to transformers, which consist of attention and feedforward neural network modules. Given the impracticality of retraining large models on entire datasets, these methods' efficiency is crucial. Recent research highlights best practices for compressing models like BERT and suggests future directions for crafting lightweight, accurate NLP models for low-capability devices or latency-sensitive applications [25, 26, 27].

Knowledge distillation (KD) involves training a smaller student model to mimic the outputs of a larger teacher model, achieving similar performance with improved computational efficiency. DistilBERT is a notable example, offering a lighter, faster BERT variant [28]. Stagewise Knowledge Distillation (SKD) refines this by training the student model in stages, learning from the teacher's feature maps,

and minimizing data needs [29]. Combining task-specific structured pruning with distillation has produced compressed models that maintain high performance across various speed/accuracy trade-offs [30]. The KD-Lib library integrates multiple compression techniques, offering a modular design for algorithm support and hyperparameter tuning [24].

Pruning aims to remove redundant parameters and connections, enhancing model efficiency [18]. Post-training pruning eliminates unnecessary weights from a pre-trained model without retraining, aiding compression [31]. The SlimNets method, combining weight pruning, low-rank factorization, and knowledge distillation, demonstrates that compressed networks can retain high accuracy [18]. Application-Specific Compression (ASC) optimizes performance by pruning deep learning model components based on their task-specific data representation contributions [21].

Quantization reduces model weight and activation precision, decreasing memory usage and speeding up inference, crucial for deploying large models on edge devices with limited resources [32]. 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, as demonstrated by Suwannaphong et al., effectively reduces model size while maintaining performance under resource constraints [4]. Challenges remain, particularly in managing unique sparsity patterns and dynamic routing mechanisms in models like Mixture-of-Experts during post-training quantization [33].

Integrating these core compression techniques is essential for developing robust models that maintain high performance across diverse environments. Techniques such as XTC, which combine lightweight layer reduction with quantization-aware training, exemplify the potential for extreme compression of transformer models [20]. These methods' effectiveness is benchmarked across various deep learning models, highlighting their impact on image classification, object detection, language models, and generative tasks [34].

3 Model Compression Techniques

Category	Feature	Method
Knowledge Distillation	Layer and Level Extraction	MLKD-BERT[17]
	Loss Function Techniques	DistilBERT[28]
	Incremental Learning Strategies	QKD[4]
	Parameter and Sequence Optimization	KT[15]
Pruning Techniques	Automated and Data-Driven Evaluation	KCM[35], AEBERT[36]
	Hybrid and Iterative Techniques	PQK[37], KESI[38]
	Layer and Weight Reduction	GP[39], GLP[40]
Quantization Methods	Efficiency-Enhancing Techniques	LCQ[10], CVXQ[41]
	Training and Learning Integration	CA-QAT[42], QKD[43], W4A4[44]
	Precision and Sparsity Strategies	SQIM[22], AQ-BERT[45]
Hybrid Techniques	Compression Techniques	SNN[18], TQC[23], PTQ[46], CrAM[12], OMC[47], ASC[21], DFKD-T3[7], QATCT[48]
Innovative Approaches	Efficiency Optimization	DTB[8], TA-DCCL[14]
	Incremental Learning	Pro-KD[49]

Table 1: Summary of model compression techniques categorized into Knowledge Distillation, Pruning Techniques, Quantization Methods, Hybrid Techniques, and Innovative Approaches. Each category highlights specific features and corresponding methods, illustrating diverse strategies for optimizing transformer models for efficient deployment in resource-constrained environments.

Deploying transformer models in resource-constrained environments necessitates model compression techniques that optimize efficiency without compromising performance. Knowledge distillation (KD) is a key method that transfers knowledge from larger teacher models to smaller student models, thereby reducing computational demands and enhancing adaptability across diverse contexts. As illustrated in **??**, the hierarchical structure of model compression techniques categorizes them into Knowledge Distillation, Pruning Techniques, Quantization Methods, Hybrid Techniques, and Innovative Approaches. Each category is further divided into methods and applications, highlighting the diverse strategies and their specific implementations in optimizing transformer models for efficient deployment. Table 1 provides a comprehensive overview of various model compression techniques, detailing the features and methods within each category to enhance transformer model efficiency. Additionally, Table 4 presents a comparative overview of these techniques, detailing their features and methods to enhance transformer model efficiency. The following subsection delves into the methodologies and applications of knowledge distillation in model compression.

3.1 Knowledge Distillation

Method Name	Model Compression	Learning Techniques	Deployment Scenarios
DistilBERT[28]	Theseus Compression	Knowledge Distillation	Edge Applications
Pro-KD[49]	Knowledge Distillation	Progressive Knowledge Distillation	-
QKD[4]	Knowledge Distillation	Knowledge Distillation	Edge Devices
MLKD-BERT[17]	Mlkd-BERT	Multi-level Knowledge	Resource-limited Devices
KT[15]	Knowledge Translation	-	-
DTB[8]	Sequence-length Reduction	Transformer Distillation	Various Computational Budgets

Table 2: This table presents an overview of various model compression and knowledge distillation methods, highlighting their specific learning techniques and deployment scenarios. It provides a comparative analysis of approaches such as DistilBERT, Pro-KD, and MLKD-BERT, among others, illustrating their applicability in edge and resource-limited environments.

Knowledge Distillation (KD) enables a smaller student model to approximate the performance of a larger teacher model, significantly reducing size and computational requirements [28]. By learning from the teacher's outputs, which provide richer information than hard labels, the student model gains nuanced understanding, especially beneficial in scenarios with noisy labels and class imbalance [50, 51]. Progressive Knowledge Distillation (Pro-KD) enhances KD by allowing incremental learning from the teacher, refining the student's learning process and improving performance [49]. Additionally, integrating KD with quantization techniques optimizes transformer models for edge devices [4]. The KD-Lib library benchmarks various KD algorithms, aiding in selecting suitable methods for specific applications [24]. Table 2 provides a comparative overview of different model compression and knowledge distillation techniques, detailing their learning methodologies and potential deployment scenarios.

Innovative methods like Theseus Compression replace BERT modules with compact substitutes during training, offering novel model compression strategies [13]. The MLKD-BERT approach enhances performance by leveraging feature-level and relation-level knowledge, allowing flexible student model configurations [17]. Hybrid approaches combining KD with techniques like weight pruning and matrix factorization achieve significant model compression without sacrificing performance [15]. Dynamic sequence length adjustments during inference further exemplify KD's adaptability in optimizing efficiency while minimizing accuracy loss [8].

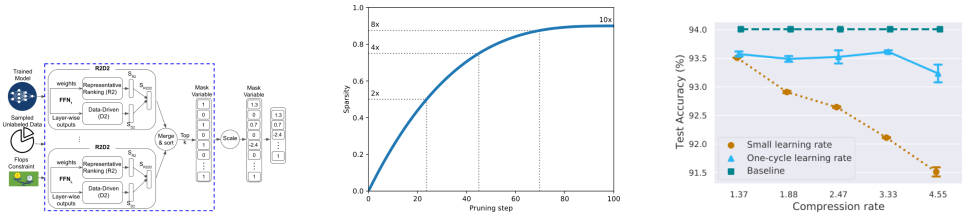
Recent innovations in KD include data-free techniques like DFKD-T3, which utilize generative language models to address data scarcity and privacy challenges by transforming general corpora into task-specific training data, outperforming state-of-the-art methods across various tasks [7, 29]. These advancements continue to enhance model efficiency and adaptability in diverse deployment environments.

3.2 Pruning Techniques

Pruning is a critical model compression technique that enhances transformer model efficiency by removing redundant parameters, thus reducing size and computational complexity while maintaining performance. This is particularly beneficial for deploying models on resource-constrained devices [9]. Unstructured pruning removes non-essential weights, allowing granular size reduction, while structured pruning eliminates entire neurons or channels for more predictable efficiency improvements [9]. SparseBERT exemplifies structured pruning, maintaining performance during fine-tuning [9].

Innovative approaches like Automatic Efficient BERT Pruning (AE-BERT) automate pruning, evaluating sub-networks without extensive fine-tuning, reducing computational costs [36]. Greedy Layer Pruning (GLP) offers a flexible performance-speed trade-off by pruning layers before fine-tuning [40]. Hybrid methods, such as PQK, combine pruning and quantization followed by knowledge distillation to enhance model performance [37].

The CoFi method integrates coarse-grained and fine-grained pruning with a layerwise distillation objective, significantly reducing model size while maintaining performance [9]. Despite challenges in optimizing pruning rates, pruning techniques are essential for optimizing transformer model deployment, particularly in ultra-low power devices, ensuring performance while reducing resource usage [18].



(a) Masked Representation Learning for Efficient Unsupervised Learning[35]

(b) Sparse Sparsity[39]

(c) Comparison of Test Accuracy with Different Compression Rates and Learning Rates[38]

Figure 2: Examples of Pruning Techniques

As shown in Figure 2, pruning techniques are crucial for optimizing neural networks for efficient deployment without significantly compromising performance. The examples illustrate distinct pruning approaches, highlighting their impact on efficiency and accuracy.

3.3 Quantization Methods

Quantization methods are essential for reducing the computational complexity of transformer models by converting high-precision weights and activations into lower-precision formats, significantly decreasing memory usage and accelerating inference. This is crucial for deploying large models on resource-constrained devices like microcontroller units (MCUs) [52]. Quantization-Aware Training (QAT) integrates quantization into the training process, optimizing performance under distribution shifts. Methods like LCQ, employing a low-rank codebook for weight representation, exemplify this optimization [10]. Continuous approximations further enhance QAT by allowing accurate learning of quantization parameters [42].

Innovative techniques such as the Sparsity-Quantization Interplay Method (SQIM) combine sparsity and quantization to minimize computation errors [22]. The integration of quantization with knowledge distillation, as shown by QKD, improves accuracy over baseline methods, recovering full-precision accuracy at low-bit quantization levels [43]. Yang et al.'s method compresses embedding and linear layers of transformers into low-rank tensor cores while applying quantization-aware training to minimize size and runtime latency [48].

Post-training quantization (PTQ) methods optimize models after training for efficient deployment in NLP and CV tasks [27]. INT4 quantization, utilizing 4-bit precision for weights and activations, optimizes inference performance in transformer models [44]. The OMC framework achieves compression through quantization and efficient parameter management, facilitating lightweight operation during federated training [47].

Advanced quantization frameworks like CVXQ optimize bit depth and step size for each weight parameter, minimizing output distortion [41]. The AQ-BERT model exemplifies automatic mixed-precision quantization search, yielding significant performance improvements, particularly in low-compression scenarios [45]. Innovations such as FlattenQuant introduce structured formats for weight representation, enabling regular memory access patterns and high parallelism in decoding.

Quantization methods are vital for developing efficient transformer models, facilitating their deployment in environments with limited computational resources. The continuous evolution of these techniques promises to enhance the efficiency and applicability of transformers across various domains [53].

3.4 Hybrid Techniques

Hybrid techniques in model compression integrate multiple strategies such as pruning, quantization, and knowledge distillation to optimize transformer models. These methods leverage the strengths of individual techniques, achieving substantial reductions in model size and computational demands while maintaining high accuracy [18]. The Online Model Compression (OMC) framework reduces memory usage and communication costs, making it suitable for federated learning environments [47].

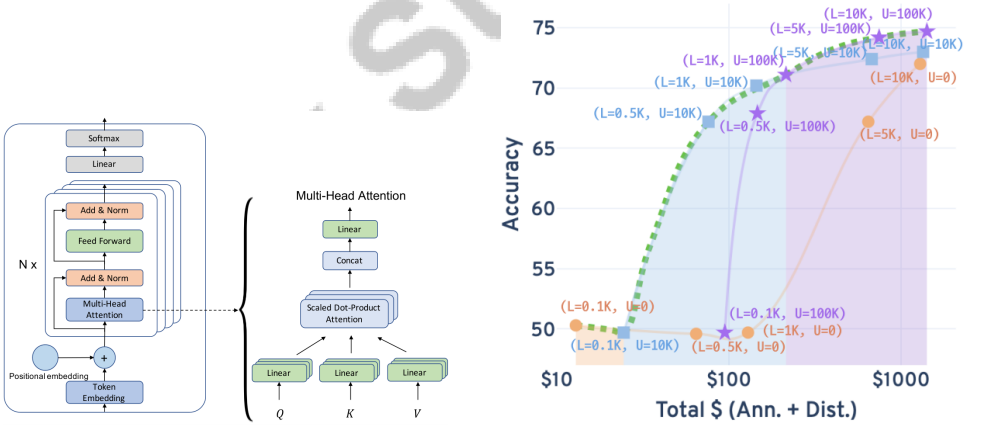
Method Name	Integration Strategies	Efficiency Optimization	Structural Enhancements
SNN[18]	Weight Pruning	Low-rank Factorization	Knowledge Distillation
OMC[47]	Quantization-aware Training	Reducing Memory Usage	Neuron Connection Optimization
ASC[21]	Pruning, Quantization	Application-specific Scenarios	Neuron Connection Optimization
TQC[23]	Knowledge Distillation	Model Size Reduction	Neuron Connection Optimization
QATCT[48]	Low-rank Tensor	Model Size Reduction	Tensor Decomposition
CrAM[12]	Pruning, Quantization	Computational Burden Reduction	Kronecker Decomposition
DFKD-T3[7]	Pruning, Quantization	Reducing Model Size	Neuron Connection Optimization
PTQ[46]	-	Quantization Resource Reduction	Quantization Technique Enhancement

Table 3: Summary of hybrid techniques in model compression, detailing the integration strategies, efficiency optimization methods, and structural enhancements employed by various methods. This table highlights the diverse approaches used to maintain model accuracy while reducing size and computational demands.

Application-specific compression (ASC) optimizes model efficiency by identifying layers that can be pruned without significantly impacting performance [21].

The TQCompressor enhances structural efficiency through neuron connection optimization and Kronecker decomposition [23]. Integrating low-rank tensor decomposition with quantization-aware training enables high compression ratios with minimal accuracy loss [48]. The CrAM framework advances hybrid compression methods by producing models robust to one-shot compression, allowing significant sparsity levels without retraining [12]. Data-free knowledge distillation (DFKD) frameworks, such as extensions to generative language models, enhance specificity and diversity in NLP tasks without relying on original training data [7].

Hybrid techniques represent a promising avenue for enhancing transformer model efficiency and performance. By integrating various compression techniques, these methodologies provide a comprehensive framework for optimizing model architectures, addressing challenges posed by contemporary machine learning models' increasing size and complexity [25, 53, 54, 21, 34]. Table 3 provides a comprehensive overview of the hybrid techniques employed in model compression, illustrating the integration strategies, efficiency optimizations, and structural enhancements utilized by different methods.



(a) Multi-Head Attention Network[46]

(b) Cost-Performance Tradeoff in Large-Scale Image Classification[55]

Figure 3: Examples of Hybrid Techniques

As shown in Figure 3, hybrid approaches in model compression optimize the balance between computational efficiency and model performance. The examples illustrate the versatility and potential of hybrid techniques in model compression.

3.5 Innovative Approaches

Innovative approaches in model compression aim to enhance transformer models' efficiency and performance while addressing resource consumption and accuracy challenges. Task-Aware Deep

Compositional Code Learning (TA-DCCL) achieves a remarkable compression rate with minimal performance degradation [14]. The integration of sharpness-aware minimization (SAM) with various compression techniques stabilizes training by smoothing the loss landscape, enhancing the robustness of compressed models. Teacher Outlier Rejection (TOR) Loss refines the distillation process by rejecting outliers based on teacher model predictions, improving student model performance [51].

Pro-KD allows students to learn from multiple teacher checkpoints throughout training, refining the learning process and improving performance [49]. KD methods are categorized based on the aspects of the teacher they mimic, providing a framework for understanding and applying KD techniques [50]. Theseus Compression proposes replacing BERT modules with compact substitutes during training, simplifying the process [13]. Dynamic adjustment of sequence lengths during inference exemplifies model compression techniques' adaptability in optimizing efficiency while minimizing accuracy loss [8].

These innovative approaches collectively illustrate the evolving landscape of model compression research, presenting effective strategies such as pruning, quantization, and knowledge distillation essential for optimizing transformer models. These techniques aim to significantly reduce memory and computational demands, enhancing deployment across various applications and environments with limited resources and stringent latency requirements [26, 56, 27, 57, 34].

Feature	Knowledge Distillation	Pruning Techniques	Quantization Methods
Efficiency Improvement	Size Reduction	Redundant Removal	Precision Reduction
Deployment Context	Diverse Environments	Resource-constrained Devices	Mcus
Unique Feature	Teacher-student Model	Parameter Pruning	Low-bit Representation

Table 4: This table provides a comparative analysis of three prominent model compression techniques: Knowledge Distillation, Pruning Techniques, and Quantization Methods. It highlights key attributes such as efficiency improvement strategies, deployment contexts, and unique features of each method, offering insights into their application in optimizing transformer models for efficient deployment.

4 Applications in Edge Computing

Deploying transformer models in edge computing necessitates addressing computational constraints and real-time performance requirements. Model compression techniques are pivotal for enhancing efficiency in resource-limited settings. This section examines these techniques' applications, emphasizing improvements in inference speed and efficiency, crucial for operating complex algorithms on edge devices.

4.1 Enhancing Inference Speed and Efficiency

Improving inference speed and efficiency on edge devices is crucial for real-time transformer model applications. Pruning, quantization, and knowledge distillation reduce model size and complexity while maintaining performance. Techniques like batching and quantization significantly decrease model size and inference time, enhancing performance in edge computing environments. These strategies are vital for large language models (LLMs), addressing latency, accuracy, and energy consumption challenges [58, 59, 60, 61, 34].

Quantization methods notably improve LLMs' latency and energy efficiency. Quantization-aware training reduces model size and computational costs, making it suitable for diverse edge devices [48]. Post-training quantization heuristics, particularly for Mixture-of-Experts models, enhance performance beyond random allocation methods [33].

Pruning techniques like CoFi offer over 10× speedups while maintaining high accuracy, demonstrating structured pruning's potential to reduce model size and computational demands without extensive data [9]. Application-specific compression strategies evaluated on BERT models for tasks like Extractive QA illustrate optimized pipelines' impact on accuracy and inference time [21].

Knowledge distillation, combined with other compression techniques, significantly enhances model efficiency. DistilBERT exemplifies knowledge distillation's benefits, offering smaller size and faster inference speed, suitable for real-time mobile applications [28]. The Online Model Compression

(OMC) framework reduces memory usage and communication costs while maintaining accuracy in federated learning, showcasing hybrid compression advantages [47].

Innovative approaches like TQCompressor demonstrate significant parameter reduction while maintaining performance, compressing GPT-2 small model from 124 million to 81 million parameters [23]. Benchmarks emphasize evaluating inference speed alongside accuracy for practical deployment [34].

Shallow Cross-Encoders remain effective with CPU inference, making them viable for on-device applications [11]. Dynamic-TinyBERT improves inference efficiency while retaining competitive accuracy on the SQuAD1.1 benchmark [8]. The KD-Lib library offers a unified platform for evaluating multiple compression techniques, facilitating comparison and adoption [24].

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

Strategic application of compression techniques is crucial for enhancing inference speed and efficiency on edge devices. Recent NLP advancements have led to techniques like Greedy-layer pruning and knowledge distillation, significantly improving transformer models' efficiency and effectiveness in real-time applications. These methods address long inference times and high production costs by enabling dynamic model size adjustments tailored to specific tasks. Improvements in training efficiency through adaptive optimization algorithms, parallel computing, and mixed precision training further optimize resource utilization, ensuring swift and accurate performance of complex NLP tasks [61, 40].

4.2 Addressing Latency-Accuracy Tradeoff

Balancing latency and accuracy in edge deployments is a critical challenge for transformer models, especially in resource-limited environments. Advanced compression techniques, including pruning, quantization, and knowledge distillation, are instrumental in navigating this tradeoff, enabling efficient deployment without significant accuracy loss [62]. The APB method exemplifies this balance, achieving superior accuracy/memory trade-offs alongside notable CPU inference speed improvements [62].

Quantization methods, particularly those utilizing low-bit precision, effectively reduce model size and enhance inference speed, essential for edge device performance. Incorporating quantization-aware training optimizes this process by minimizing precision loss during deployment [48], ensuring models remain performant under reduced computational demands, effectively managing the latency-accuracy tradeoff.

Pruning techniques significantly contribute to optimizing this balance by eliminating redundant parameters, reducing model complexity and inference time, critical for real-time applications on edge devices [9]. The CoFi method exemplifies structured pruning's effectiveness in managing the latency-accuracy tradeoff while achieving substantial speedups with high accuracy [9].

Knowledge distillation complements these strategies by transferring knowledge from larger models to smaller ones, allowing the latter to perform comparably with lower computational requirements [28]. DistilBERT's advantages, including its smaller size and faster inference speed, underscore knowledge distillation's effectiveness in balancing latency and accuracy in edge deployments [28].

Hybrid compression techniques, integrating multiple strategies, offer a comprehensive approach to optimizing model performance in edge environments. These methods leverage individual techniques' strengths to achieve substantial reductions in model size and computational demands while maintaining high accuracy [18]. The ongoing evolution of these techniques highlights the necessity for 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 scenarios demonstrates their significant impact on enhancing transformer models' efficiency and performance across various

domains. In e-commerce search, the BERT2DNN framework exemplifies model compression benefits, enhancing performance and proving suitable for large-scale applications [63]. 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 use knowledge distillation to achieve competitive performance with reduced computational demands [64]. This dataset provides a robust platform for assessing compressed models' scalability and adaptability across diverse NLP tasks, reinforcing their applicability in information retrieval systems.

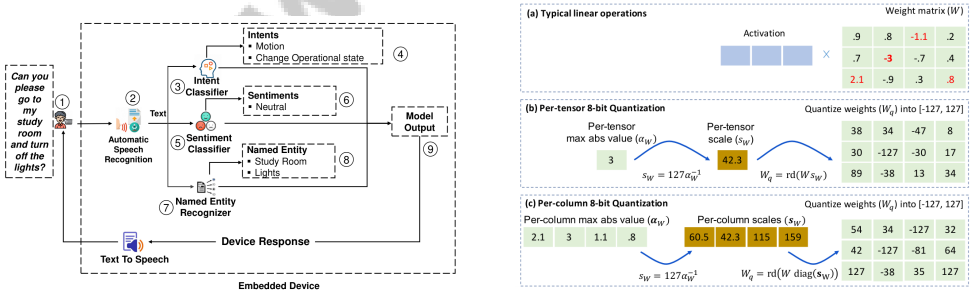
In NLP applications, the TT-embedding technique significantly reduces parameters in embedding layers while maintaining or enhancing model accuracy [65]. This method underscores the potential of tensorized approaches in practical applications, enabling efficient deployment of large-scale models in resource-limited environments.

An empirical study on low-precision techniques, using datasets like CIFAR10 for image classification and Speech Commands for keyword spotting, illustrates model compression methods' versatility across different tasks [66]. These use cases highlight compression strategies' adaptability in optimizing model performance for a wide range of applications, from visual recognition to auditory processing.

SDQ, which combines structured sparsity and quantization, is particularly advantageous for applications requiring real-time inference from large language models, such as conversational agents and automated content generation [19]. This approach illustrates model compression's practical benefits in enhancing AI-driven systems' responsiveness and efficiency 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 reliance 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 relevant for applications demanding precise regression predictions, such as age estimation and gaze tracking, reinforcing model compression's importance in enhancing predictive analytics' accuracy and efficiency [51]. These case studies collectively underscore model compression techniques' transformative impact 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[67]

(b) Quantization of Weight Matrices in Neural Networks[46]

Figure 4: Examples of Real-World Applications and Case Studies

As shown in Figure 4, real-world applications and case studies in edge computing demonstrate the transformative potential of this technology across various domains. One example is a flowchart of a conversational assistant system, showcasing the seamless integration of natural language processing in everyday tasks. This system efficiently processes a user's spoken request, such as turning off the lights in a study room, by converting speech to text and accurately classifying intent, exemplifying edge computing's practical application in smart home automation. Another compelling example is the quantization of weight matrices in neural networks, crucial for optimizing computational efficiency

and resource management in edge devices. Employing methods like per-tensor and per-column quantization ensures that neural networks operate effectively within edge environments' constraints, enabling powerful and greener computational capabilities [67, 46].

4.4 Challenges in Edge Deployment

Deploying compressed transformer models in edge environments presents challenges, particularly in balancing model size reduction and maintaining acceptable accuracy levels. While effective in reducing model size, pruning techniques often struggle to preserve performance, which is critical for real-time applications on edge devices [39]. SparseBERT, for example, achieves higher compression rates while maintaining performance, yet the complexities of achieving this balance highlight the challenges of deploying these models in resource-constrained settings [68].

The heterogeneity of edge devices adds complexity, as these devices vary widely in computational power, memory capacity, and energy availability. This variability necessitates adaptable compression techniques tailored to each deployment environment's specific constraints. The dynamic nature of edge environments, characterized by fluctuating network conditions and varying computational resources, requires robust and flexible models capable of adapting to changes without significant performance degradation. This is particularly crucial as large language models (LLMs) and deep neural networks (DNNs) face resource constraints, where traditional deployment methods may fall short. Advanced model compression techniques, such as structured pruning and filter pruning, are being explored to create lightweight, efficient models that maintain accuracy while significantly reducing memory and energy requirements. Innovative approaches like knowledge transfer enable on-device models to learn incrementally from new data and adapt to unseen categories, enhancing resilience in edge computing scenarios [59, 32, 60].

Integrating compressed models with existing edge infrastructure presents considerable technical challenges, primarily due to the substantial computing, memory, and energy requirements of LLMs and DNNs. These models often necessitate specialized compression techniques—such as pruning, quantization, and knowledge distillation—to reduce resource demands while maintaining performance. Additionally, traditional fine-tuning methods require significant GPU memory, exceeding mainstream hardware capabilities, complicating deployment on edge devices. As the demand for efficient deployment of multi-modal foundation models grows, innovative solutions like structured pruning and memory-efficient fine-tuning methods are essential to address these challenges and enable sustainable advancements in artificial intelligence [59, 26, 60, 27]. Ensuring compatibility and seamless operation with diverse hardware and software ecosystems is crucial for successful deployment. Furthermore, continuous updates and maintenance of these models to address security vulnerabilities and improve performance over time add to the complexity of managing deployments in edge environments.

Privacy and data security are critical considerations in edge deployments. Compressed models must handle sensitive data locally, minimizing the need for data transmission to centralized servers. To safeguard user privacy while maintaining model inference efficiency and speed, comprehensive encryption techniques and stringent data handling protocols must be implemented. These measures should protect sensitive information while ensuring complex models, such as those used in text-to-image generation, remain optimal, even when subjected to pruning and distillation processes aimed at reducing model size and enhancing computational efficiency [31, 69].

To effectively tackle the challenges of deploying deep neural networks (DNNs) in edge computing environments, continuous advancements in model compression techniques, such as structured pruning and quantization, must be prioritized alongside a comprehensive strategy that integrates optimized models into diverse edge device ecosystems, ensuring compliance with stringent energy efficiency, computational resource, and performance metrics [59, 56, 34, 70]. Overcoming these obstacles will optimize transformer models' deployment in edge environments, delivering high performance while maintaining resource efficiency and data security.

5 Challenges and Future Directions

5.1 Accuracy and Robustness Challenges

The accuracy and robustness of transformer models post-compression face significant challenges due to trade-offs inherent in techniques such as pruning, quantization, and knowledge distillation.

While these methods effectively reduce model size, they can lead to performance degradation that necessitates careful optimization. For instance, pruning may inadvertently compromise both general-purpose and task-specific knowledge, resulting in performance that lags behind smaller dense models [6]. Excessive pruning or aggressive quantization often leads to substantial accuracy drops, particularly in compressed models with limited expressiveness [14].

Quantization, especially at very low bit-widths, poses risks of significant accuracy loss. The interplay between quantization and knowledge distillation complicates matters further, as the regularization effect of knowledge distillation may adversely affect the performance of quantized networks [48]. Moreover, existing compression methods, particularly quantization, frequently require manual hyper-parameter tuning and lack support for fine-grained subgroup-wise quantization, adding to the complexity [10]. Experiments indicate that both pruning and quantization can influence the transferability of adversarial samples, suggesting that attacks on compressed models may also affect uncompressed counterparts, thereby exposing potential security vulnerabilities [12].

Knowledge distillation methods struggle with effectively transferring task-relevant signals, risking performance drops in distilled models relative to their teacher models. The reliance on the teacher model's quality is a notable limitation; a subpar teacher can yield a poorly performing distilled model [51]. Additionally, the capacity gap between teacher and student models necessitates robust distillation strategies to facilitate effective learning [50]. The accuracy of compressed models also varies significantly based on the chosen activation record strategy, complicating post-compression accuracy maintenance [8].

Maintaining fairness and explainability in compressed models further challenges their robustness, as biases and toxicity may emerge, necessitating innovative solutions to balance efficiency and performance [49]. The high computational demands of large models remain a primary hurdle, with many methods requiring substantial resources during training, limiting their practical applicability [16]. Current studies often inadequately address the trade-offs between model size and performance, with many compressed models still underperforming relative to their uncompressed equivalents [6].

Addressing these challenges requires refining compression techniques and exploring novel methodologies to ensure robust and accurate models for diverse applications. The development of more efficient methods shows promise in preserving original model accuracy, although certain quantization settings may still lead to minor performance drops [14]. As research advances, integrating collaborative learning approaches, such as combining knowledge distillation with mutual learning, may enhance performance and efficiency. Additionally, approaches like BERT-of-Theseus may face challenges in scenarios where predecessor and successor module architectures differ significantly, potentially impacting performance [13].

5.2 Optimization of Compression Techniques

Optimizing compression techniques for transformer models is essential for enhancing computational efficiency while preserving performance across diverse tasks. Future research should aim to refine quantization schemes to maintain model accuracy at low bit depths, addressing the trade-offs between compression levels and predictive accuracy [71]. This includes investigating the application of LCQ to multi-modal models and optimizing its methodology to improve performance and applicability [10]. Furthermore, exploring the generalization of advanced hierarchical optimization (AHO) across various model types and datasets, and integrating it with other optimization frameworks, could enhance its capabilities [72].

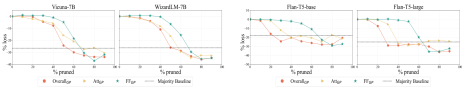
Pruning strategies also warrant further exploration, particularly the application of CoFi for upstream pruning to create task-agnostic models, alongside investigating additional optimizations in the pruning process [9]. The development of tuning-free compression techniques and hybrid approaches that combine multiple compression methods could yield optimal results in terms of efficiency and performance [1]. Future research might also focus on optimizing the distillation process, indicating potential improvements in compression techniques [28].

Integrating knowledge distillation with mutual learning presents a promising avenue for improving performance and efficiency, overcoming the primary challenge of current methods that require complete re-training of compressed models or are constrained by architectural limitations [15]. Additionally, exploring prolonged CrAM training could enhance model performance and robustness while investigating methods to reduce computational complexity in the CrAM update [12].

Frameworks like KD-Lib provide valuable resources for evaluating multiple model compression techniques, though 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 applicable across various domains.

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

(a) The image shows a mathematical equation.[73]



(b) Comparison of Loss Reduction Across Different Pruning Strategies for Various Models[74]

Figure 5: Examples of Optimization of Compression Techniques

As illustrated in Figure 5, optimizing data compression techniques to enhance efficiency and effectiveness is a critical research area. The first subfigure presents a mathematical equation fundamental to understanding the principles of compression optimization, expressed in LaTeX to demonstrate a summation over a set of variables, providing a theoretical foundation for evaluating and improving compression strategies. The second subfigure compares loss reduction across different pruning strategies applied to various models, including Vicuna-7B, WizardLM-7B, Flan-T5-base, and Flan-T5-large, visually representing the impact of pruning on model performance. The x-axis indicates the percentage of weights pruned, while the y-axis shows the corresponding percentage of loss reduction. Analyzing these graphical representations can yield insights into the challenges and potential future directions for optimizing compression techniques, ultimately leading to more efficient and effective data management solutions [73, 74].

5.3 Adaptability and Generalization

The adaptability and generalization of compressed models across diverse tasks and datasets are vital for their effective deployment. Future research should explore mixed-precision approaches that optimize quantization-aware pruning (QAP) methods, enhancing model adaptability and validation across various tasks and datasets [75]. Investigating heterogeneous sparsity and quantization schemes, along with their interactions, could also contribute to optimizing model performance, ensuring compressed models maintain robustness and adaptability in varying computational environments [22].

Developing advanced techniques for knowledge distillation and further optimizing quantization are essential for enhancing model performance on constrained devices, such as edge computing platforms [4]. Additionally, exploring alternative methods for distilling intermediate representations could enhance the adaptability of compressed models across various tasks and datasets, facilitating their application in diverse linguistic and computational contexts [76].

Improving quantization methods for ultra-low precision and developing frameworks that support lower precision computations are crucial for advancing the generalization capabilities of compressed models, particularly in resource-limited scenarios [66]. Moreover, future research could optimize vocabulary selection and investigate combining these methods with other distillation techniques to enhance model adaptability across different NLP tasks [77].

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 [14]. Enhancing the algorithm's robustness against network fluctuations and exploring applicability in other domains, such as real-time data analytics, could further extend the generalization capabilities of compressed models [78].

Finally, exploring 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 focus on enhancing multilingual translation and improving compatibility with various AI accelerators beyond ARM CPUs, ensuring compressed models can adapt to diverse hardware and software environments [16]. Addressing these research directions can significantly enhance the adaptability and generalization of compressed models, ensuring their effectiveness in dynamic environments.

5.4 Resource Constraints and Deployment

Deploying compressed transformer models in resource-constrained environments presents significant challenges, particularly concerning processor, memory, and battery limitations of embedded devices. Balancing computational efficiency with accuracy is critical, as these models often need to execute complex natural language processing tasks under stringent resource constraints [6]. A major hurdle is the extensive retraining often required 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 necessitating more resources and time compared to standard techniques [48]. Additionally, the compatibility of compressed models with existing infrastructure is critical; for instance, the fixed data precision some methods assume may not fully leverage the benefits of lower precision computations across all scenarios, potentially limiting their applicability [50].

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 [48]. 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]. Moreover, exploring hybrid models and independent layer compression techniques could fully leverage transformer architectures for efficient compression [15].

By effectively tackling the challenges associated with resource limitations in edge and embedded computing environments, the deployment of compressed transformer models can be significantly enhanced. This optimization not only improves performance but also ensures that these models, crucial for applications in natural language processing and computer vision, remain practical for devices with limited computational capacity. Recent advancements in model compression techniques, such as pruning, quantization, and knowledge distillation, provide viable strategies to reduce the memory and computational demands of large transformer models, enabling their successful implementation in real-world scenarios where efficiency and speed are paramount [26, 56, 34, 27]. The reduced serving costs of approaches like BERT2DNN enhance accessibility for organizations with limited resources, highlighting the broader applicability of these models despite resource constraints.

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 explore the application of Modular Transformers to other architectures and larger models, as well as investigate further optimizations for different resource constraints [57]. Additionally, optimizing compression algorithms and examining the application of Online Model Compression (OMC) in various federated learning scenarios could enhance model efficiency and adaptability [47].

Developing 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 [31]. Further optimizations in the clustering process and the applicability of these methods to a broader range of model architectures should also be explored [79]. Moreover, optimizing generative models 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 [51]. 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 [49].

Further research could focus on optimizing the training process and extending methods like MLKD-BERT to tasks beyond natural language understanding, broadening the scope and impact of model compression techniques [17]. Additionally, developing 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 pruning, and knowledge distillation with emerging technologies, researchers can create more efficient and adaptable AI models that reduce size and energy consumption while enhancing performance on resource-limited devices. This approach addresses the growing demands of modern AI applications, enabling the deployment of high-performing models across diverse environments, including edge devices, while maintaining accuracy and inference speed. Leveraging data augmentation strategies tailored to model size can further optimize performance, ensuring that compressed models meet the evolving requirements of various applications, from image classification to language processing [53, 80, 34].

6 Conclusion

The investigation into model compression techniques underscores their pivotal role in refining transformer models for NLP tasks, effectively managing computational challenges and resource limitations. Techniques such as knowledge distillation, pruning, and quantization are essential in reducing model size and complexity while maintaining performance. Knowledge distillation, in particular, holds promise for enhancing model efficiency, as illustrated by frameworks that facilitate the implementation and assessment of various compression strategies. Pruning, especially through stage-wise methods, has significantly advanced accuracy and effectiveness, achieving leading performance in deep neural networks. The integration of pruning with hybrid frameworks exemplifies an ideal balance between model size and accuracy, particularly in constrained environments. Quantization, including extreme methods, provides efficient solutions for achieving high compression ratios without sacrificing accuracy. Innovative approaches like tensor decompositions and permutations in compression techniques point to promising future directions in neural network optimization. The strategic use of these compression methods is crucial for the effective deployment of transformer models in NLP, enabling real-time processing of complex tasks while optimizing resource use. As research progresses, the continuous refinement and integration of these techniques will be critical in advancing the field, offering substantial potential for improving 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] Arhum Ishtiaq, Sara Mahmood, Maheen Anees, and Neha Mumtaz. Model compression, 2021.

[26] 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.

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

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

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

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

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

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

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

[34] 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.

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

[36] 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.

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

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

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

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

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

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

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

[44] 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.

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

[46] 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.

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

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

[49] 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.

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

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

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

[53] 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.

[54] Yerlan Idelbayev and Miguel Á. Carreira-Perpiñán. A flexible, extensible software framework for model compression based on the lc algorithm, 2020.

[55] Junmo Kang, Wei Xu, and Alan Ritter. Distill or annotate? cost-efficient fine-tuning of compact models, 2023.

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

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

[58] 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.

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

[60] 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.

[61] 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.

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

[63] 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.

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

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

[66] 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.

[67] 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.

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

[69] 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.

[70] 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.

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

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

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

[74] 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.

[75] 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.

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

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

[78] 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.

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

[80] 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.