iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

A Comparative Study of Small Language Models for Resource - Constrained AI Systems

Omkar Sanjay Pomendkar, Dr. Rashmi Thakur
Department of Computer Engineering Thakur College of Engineering & Technology, India, Mumbai

Abstract: ***Large Language Models (LLMs) have achieved remarkable success in natural language processing tasks, but their deployment requires high computational resources, large memory capacity, and significant energy consumption. These limitations make them unsuitable for deployment on resource-constrained devices such as mobile phones, embedded systems, and Internet of Things (IoT) devices. Small Language Models (SLMs) provide an efficient alternative by reducing the number of parameters while maintaining acceptable performance across many NLP tasks. This paper presents a comparative study of several small language models including Phi-3 Mini, Gemma, TinyLlama, and StableLM. The research analyzes architectural design, parameter efficiency, computational cost, and inference performance for deployment in constrained environments. The study highlights advantages of model compression techniques such as quantization, pruning, and knowledge distillation in improving efficiency without significantly affecting accuracy. Experimental analysis and literature insights indicate that SLMs provide a practical balance between performance and efficiency for edge-based AI systems.***

Keywords: Small Language Models (SLMs), Resource-Constrained AI Systems, Edge Artificial Intelligence, Natural Language Processing (NLP), Model Compression, Knowledge Distillation, Quantization, Transformer Architecture, Edge Computing, Efficient Language Models.

I. INTRODUCTION

The increasing digitization of electoral processes has led to the generation of large volumes of voter-related data, including demographic records, survey responses, and unstructured textual feedback. Efficient management and analysis of this data are critical for informed decision-making in modern Voter Management Systems (VMS). However, voter data—especially in countries like India—is often multilingual, inconsistent, and noisy, posing significant challenges for traditional data processing techniques.

Recent advancements in Natural Language Processing (NLP), particularly Large Language Models (LLMs), have demonstrated strong capabilities in handling unstructured text. Nevertheless, their high computational requirements, latency, and dependence on cloud infrastructure limit their applicability in real-time and resource-constrained environments. These limitations are particularly critical in field-level voter data collection systems that rely on mobile and edge devices.

Small Language Models (SLMs) offer a viable alternative by providing efficient language understanding with reduced computational overhead. Their lightweight architecture enables deployment on edge devices, making them suitable for real-time preprocessing and analysis of voter data. Despite this potential, existing research primarily focuses on model comparison, with limited exploration of system-level integration of SLMs in practical applications.

This paper addresses this gap by proposing an adaptive framework that integrates SLMs into the end-to-end pipeline of voter data preprocessing and analysis. The framework enables context-aware data cleaning, multilingual processing, and real-time sentiment and issue classification while maintaining low resource consumption. The proposed approach aims to enhance the efficiency, scalability, and responsiveness of VMS in resource-constrained environments.

II. RELATED THEORY

Language models have evolved significantly over the past decade. Early models were based on statistical methods such as n-gram models. Later, neural language models improved contextual understanding using recurrent neural networks.

The introduction of transformer architecture enabled large-scale training of language models. While these models achieved high accuracy, they also required massive computational resources.

Small Language Models address this challenge by reducing parameter size and improving training efficiency. Modern SLMs use techniques such as parameter sharing, efficient tokenization, and optimized training datasets to maintain strong performance despite smaller architectures.

A. Transformer Architecture

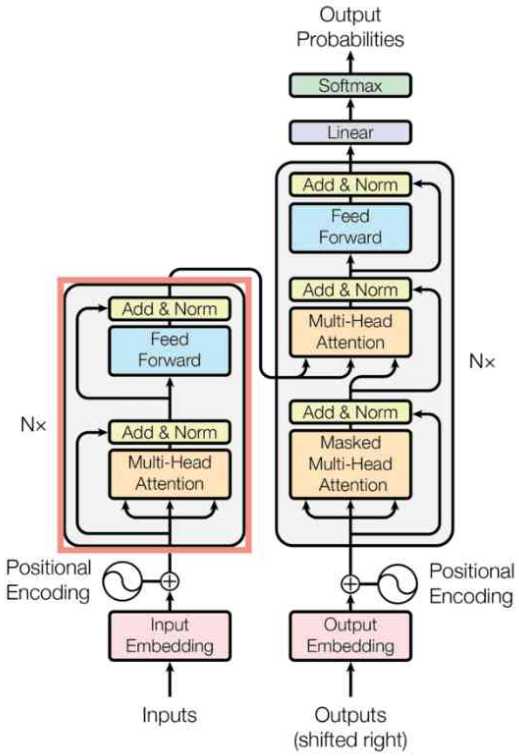


Fig. 1. Transformer Architecture

Most modern language models are built using the Transformer architecture, which was introduced to overcome limitations of recurrent neural networks (RNNs) and long short-term memory (LSTM) models. Transformers rely on a self-attention mechanism that allows the model to process entire sequences of text simultaneously rather than sequentially.

The key components of the transformer architecture include:

Token Embedding

Positional Encoding

Multi-Head Self-Attention

Feedforward Neural Networks

Layer Normalization

The self-attention mechanism enables the model to capture contextual relationships between words regardless of their distance within a sentence. This significantly improves the model's ability to understand long-range dependencies in language.

B. Large Language Models (LLMs)

Large Language Models are deep neural networks trained on massive datasets containing billions or trillions of words. These models typically contain billions of parameters and are capable of performing a wide range of language tasks with minimal task-specific training.

Examples of LLM capabilities include:

Text generation

Language translation

Code generation
Question answering
Document summarization
Although LLMs provide high accuracy and strong reasoning abilities, they require substantial computational resources for both training and inference. High GPU memory requirements and energy consumption make them difficult to deploy on mobile devices or edge computing platforms.

C. Small Language Models (SLMs)
Small Language Models are designed to provide efficient language processing capabilities while using significantly fewer parameters than traditional large models. Typically, SLMs contain between hundreds of millions to a few billion parameters.
SLMs focus on optimizing performance through:
Efficient model architectures
Data-centric training approaches
Parameter sharing techniques
Lightweight transformer implementations
Because of their smaller size, SLMs can be deployed on resource-constrained systems such as smartphones, embedded devices, and IoT platforms.

D. Model Compression Techniques
Several techniques are used to reduce the size and computational requirements of language models while preserving performance.

1) Quantization
Quantization reduces the numerical precision of model weights, typically converting 32-bit floating-point values into 8-bit or 4-bit representations. This significantly reduces memory usage and increases inference speed.

2) Pruning
Pruning removes redundant or less important parameters from a neural network. This technique reduces the overall complexity of the model while maintaining acceptable performance.

3) Knowledge Distillation
Knowledge distillation is a training approach where a smaller "student model" learns from a larger "teacher model." The student model imitates the output distribution of the teacher, allowing it to achieve similar performance with fewer parameters.

III. LITERATURE SURVEY

The evolution of Natural Language Processing (NLP) has been significantly influenced by the introduction of the transformer architecture by Vaswani et al. [1]. This model replaced traditional recurrent neural networks (RNNs) and long short-term memory (LSTM) networks with a self-attention mechanism, enabling efficient parallel processing and improved handling of long-range dependencies in textual data.
Subsequent advancements led to the development of Large Language Models (LLMs) such as GPT-3 by Brown et al. [2] and BERT by Devlin et al. [3]. These models demonstrated exceptional performance across a wide range of NLP tasks, including text generation, translation, and question answering. However, their reliance on billions of parameters results in high computational cost, significant memory requirements, and increased energy consumption, making them unsuitable for deployment in resource-constrained environments.
To improve efficiency, Touvron et al. introduced LLaMA [4], which utilizes optimized training strategies and architectural refinements to achieve competitive performance with relatively fewer parameters. Despite these improvements, such models still require considerable computational resources, limiting their practical applicability in edge-based systems.
Recent research has focused on the development of Small Language Models (SLMs) to address these limitations. Microsoft's Phi-3 models [5] demonstrate that strong reasoning capabilities can be achieved using smaller parameter sizes through data-centric training approaches.

Similarly, Google's Gemma models [6] are designed for efficient and open deployment, offering lightweight architectures suitable for scalable applications. Other models such as TinyLlama and StableLM further contribute to this domain by providing compact transformer-based solutions optimized for low-resource environments.

In addition to architectural advancements, model compression techniques have been widely explored to enhance efficiency. Quantization techniques, such as those proposed in QLoRA [8], reduce numerical precision to lower memory usage and improve inference speed. Pruning methods introduced by Han et al. [10] eliminate redundant parameters, thereby reducing model complexity. Knowledge distillation, proposed by Hinton et al. [9], enables smaller models to learn from larger models, achieving comparable performance with significantly fewer parameters.

Furthermore, Hoffmann et al. [11] emphasized the importance of compute-optimal scaling, highlighting the relationship between dataset size and model parameters for efficient training. Comprehensive surveys such as Minaee et al. [12] summarize advancements in deep learning for NLP; however, they provide limited focus on deployment in resource-constrained environments.

Despite these advancements, several challenges remain. Most existing studies prioritize model accuracy over practical deployment constraints such as latency, energy efficiency, and hardware limitations. Additionally, there is a lack of comprehensive comparative analysis of multiple SLMs under consistent evaluation conditions.

Therefore, this study aims to address these gaps by conducting a comparative analysis of Small Language Models, including Phi-3 Mini, Gemma, TinyLlama, and StableLM. The evaluation focuses on performance, computational efficiency, and suitability for deployment in resource-constrained AI systems.

Ref. No.	Author(s) & Year	Title / Framework	Methodology	Key Finding	Research Gaps / Limitation
[1]	Vaswani et al. (2017)	Attention Is All You Need	Transformer architecture with self-attention mechanism	Introduced transformer architecture that improved NLP performance	High computational cost for large-scale models
[2]	Brown et al. (2020)	Language Models are Few-Shot Learners	Large-scale transformer-based GPT-3 model	Demonstrated strong few-shot learning capability	Extremely large model requiring massive resources
[3]	Devlin et al. (2019)	BERT: Pre-training of Deep Bidirectional Transformers	Bidirectional transformer pretraining using masked language modeling	Improved contextual language understanding	Large model size limits edge deployment
[4]	Touvron et al. (2023)	LLaMA: Open and Efficient Foundation Language Models	Efficient transformer architecture trained with optimized datasets	Achieved high performance with fewer parameters compared to GPT models	Still computationally expensive for mobile devices
[5]	Microsoft Research (2024)	Phi-3 Technical Report	Data-centric training with small parameter models	Demonstrates strong reasoning performance using small models	Requires high-quality curated datasets
[6]	Google Research (2024)	Gemma: Open Models Based on Gemini	Lightweight transformer architecture optimized for efficiency	Designed for open AI deployment and efficient inference	Limited benchmark evaluation on edge devices
[7]	Hugging Face Research (2024)	SmolLM: Efficient Small Language Models	Training compact models using curated large-scale datasets	Achieves good NLP performance with low parameter size	Limited real-world deployment analysis
[8]	Dettmers et al. (2023)	QLoRA: Efficient Finetuning of Quantized LLMs	Quantization and low-rank adaptation techniques	Enables efficient fine-tuning on limited hardware	Still requires base LLM infrastructure
[9]	Hinton et al. (2015)	Knowledge Distillation for Neural Networks	Teacher-student training framework	Reduces model size while maintaining performance	Performance depends heavily on teacher model quality

Ref. No.	Author(s) & Year	Title / Framework	Methodology	Key Finding	Research Gaps / Limitation
[10]	Han et al. (2015)	Deep Compression of Neural Networks	Pruning and quantization techniques	Significant reduction in model size and computation	May affect model accuracy if overly compressed
[11]	Hoffmann et al. (2022)	Training Compute-Optimal Language Models	Optimal scaling between dataset size and model parameters	Improved training efficiency and model performance	Focused mainly on large-scale models
[12]	Minaee et al. (2021)	Deep Learning Based Text Classification: A Survey	Comprehensive review of deep learning models for NLP	Summarized advancements in NLP architectures	Limited focus on resource-constrained systems

IV. EXISTING SYSTEM

Traditional language models rely on extremely large architectures containing billions or trillions of parameters.
These models typically require cloud-based GPU clusters for both training and inference.
The major limitations of existing systems include:

- High computational cost
- Large memory requirements
- High energy consumption
- Difficulty deploying on edge devices

These limitations restrict the usability of large models in real-time applications such as mobile assistants, offline AI tools, and embedded systems.

V. PROPOSED SYSTEM

The proposed research focuses on analyzing and comparing multiple Small Language Models to determine their suitability for deployment in resource-constrained environments.

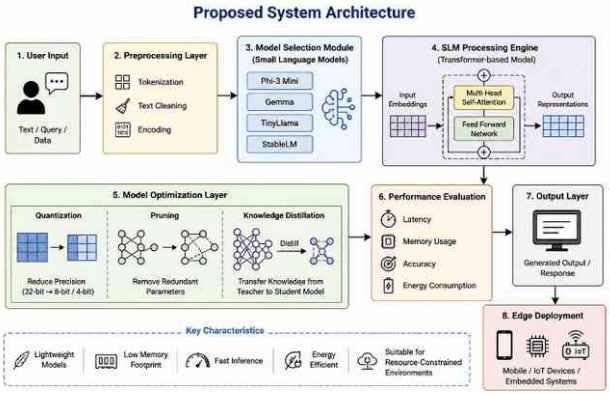


Fig. 2. Proposed System

The study evaluates models based on the following criteria:

- Model size (number of parameters)
- Inference latency
- Memory consumption
- Energy efficiency
- Task performance

This work extends the Adaptive Edge-Aware Processing Framework (AEAPF) to the domain of Voter Management Systems (VMS) by integrating Small Language Models (SLMs) into the end-to-end data processing pipeline. The proposed system is designed to efficiently handle large-scale, multilingual, and unstructured voter data in resource-constrained environments.

Unlike conventional approaches that rely on centralized cloud-based processing, the proposed framework leverages lightweight SLMs for on-device data preprocessing and real-time analysis, thereby reducing latency, bandwidth usage, and computational cost.

The novelty of this work lies in the integration of Small Language Models not only for analytical tasks but also for intelligent data preprocessing within a unified adaptive framework. This approach enables efficient, real-time voter data intelligence, which is not adequately addressed in existing literature.

VI. CONCEPTUAL FRAMEWORK

The conceptual framework of the proposed Smart Voter Management System (VMS) is designed to illustrate the interaction between data sources, preprocessing mechanisms, Small Language Models (SLMs), and analytical outputs within an adaptive edge-based environment. The framework emphasizes the role of SLMs in transforming raw, unstructured voter data into meaningful insights for decision-making.

Conceptual Framework Overview

The system operates through a sequence of interconnected components, starting from data acquisition to final decision support. The framework integrates data preprocessing, SLM-based processing, and analytical intelligence into a unified pipeline.

VII. METHODOLOGY

The research methodology consists of four major steps:

1) Model Selection – Popular SLMs are selected based on open availability and research relevance.
2) Performance Evaluation – Each model is evaluated using NLP benchmark tasks such as text classification and question answering.
3) Resource Analysis – Memory usage, latency, and energy consumption are compared.
4) Comparative Study – Results are analyzed to determine trade-offs between model size and performance.

This methodology allows identification of the most efficient models for edge-based AI applications.

VIII. CONCLUSION

Model	Parameters	Organization	Key Features
Phi-3 Mini	3.8B	Microsoft	High reasoning capability
Gemma	2B–9B	Google	Open model optimized for efficiency
TinyLlama	1.1B	Open Research	Lightweight LLaMA architecture
Qwen 2	0.5B–7B	Alibaba	Multilingual capabilities
StableLM Zephyr	3B	Stability AI	Instruction tuned model

This research presented a comparative study of Small Language Models for deployment in resource-constrained AI systems. The analysis demonstrated that modern SLMs provide strong performance while significantly reducing computational cost compared to large language models.

Models such as Phi-3 Mini and TinyLlama demonstrate promising efficiency for mobile and edge deployments. Techniques including quantization, pruning, and knowledge distillation further improve performance on limited hardware.

Future research will focus on improving reasoning capabilities of small models and developing multimodal SLMs capable of handling text, images, and speech simultaneously.

IX. ACKNOWLEDGEMENTS

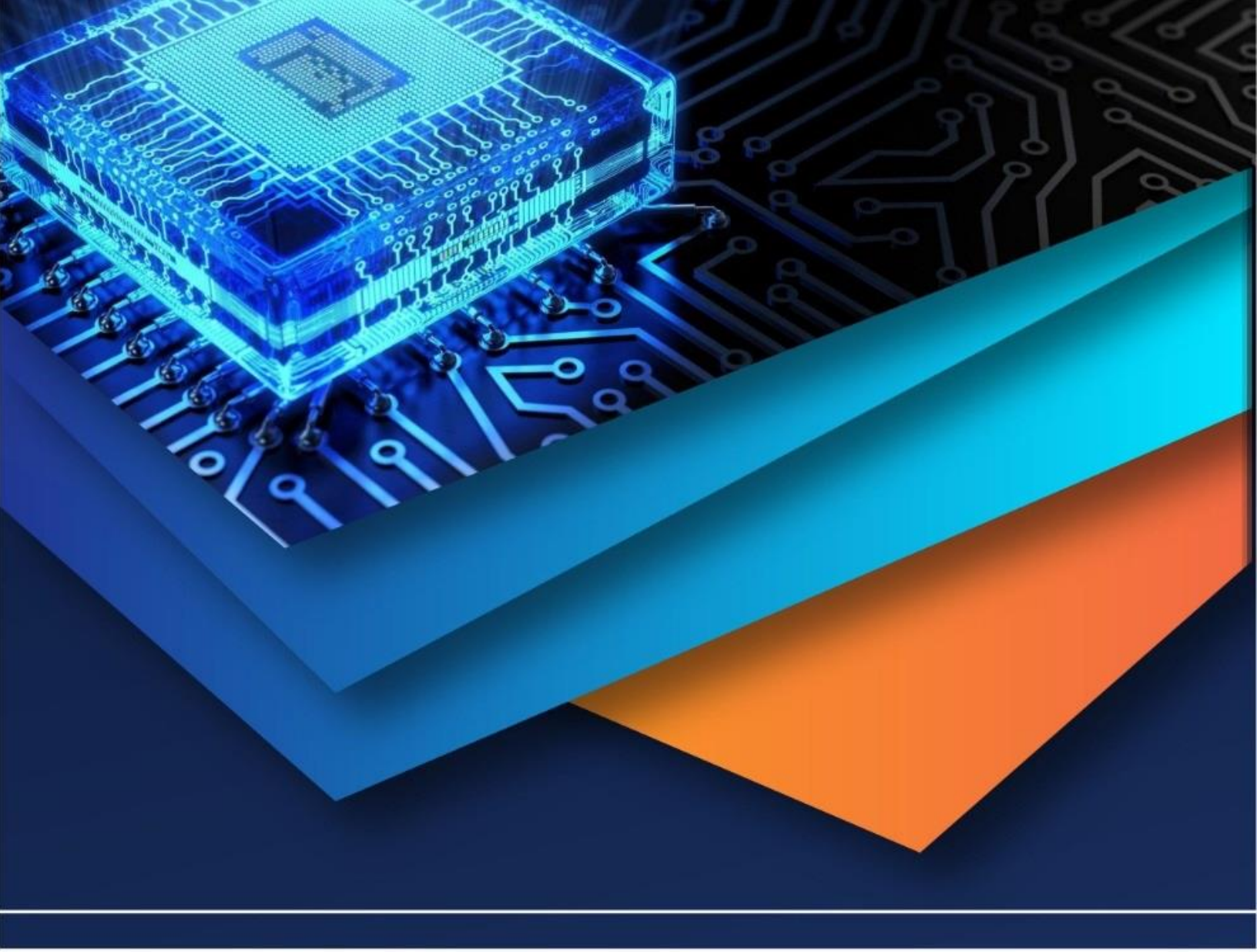
The Department of Computer Engineering provided the infrastructure and technical assistance required to conduct this research, for which the authors are deeply grateful. We also thank our project supervisor for his advice and insightful recommendations, which were crucial to finishing this study .

We would like to express our gratitude to the field workers and agricultural specialists who helped gather and validate the cotton leaf picture data. Understanding disease traits and pest symptoms was made possible by their subject expertise .

Lastly, we would like to express our gratitude to our author, institution, and guide for supporting research endeavors and offering an academic setting that enabled the effective completion of this work..

REFERENCES

[1] A. Vaswani et al., “Attention Is All You Need,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008.
[2] T. B. Brown et al., “Language Models are Few-Shot Learners,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 1877–1901.
[3] H. Touvron et al., “LLaMA: Open and Efficient Foundation Language Models,” arXiv preprint arXiv:2302.13971, 2023.
[4] Microsoft Research, “Phi-3 Technical Report: Efficient Small Language Models,” Microsoft AI Research, 2024.
[5] Google Research, “Gemma: Open Models Based on Gemini Research and Technology,” Google DeepMind Technical Report, 2024.
[6] Y. Bengio, A. Courville, and P. Vincent, “Representation Learning: A Review and New Perspectives,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.
[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.
[8] T. Wolf et al., “Transformers: State-of-the-Art Natural Language Processing,” in Proc. EMNLP System Demonstrations, 2020, pp. 38–45.
[9] N. Shazeer et al., “Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer,” in Proc. ICLR, 2017.
[10] J. Hoffmann et al., “Training Compute-Optimal Large Language Models,” DeepMind Chinchilla Paper, arXiv:2203.15556, 2022.
[11] A. Radford et al., “Improving Language Understanding by Generative Pre-Training,” OpenAI Technical Report, 2018.
[12] OpenAI, “GPT-4 Technical Report,” arXiv preprint arXiv:2303.08774, 2023.
[13] Hugging Face Research, “SmolLM: Efficient Small Language Models for Edge AI,” Hugging Face Technical Report, 2024.
[14] A. Paszke et al., “PyTorch: An Imperative Style High-Performance Deep Learning Library,” in Proc. NeurIPS, 2019.
[15] T. Dettmers et al., “QLoRA: Efficient Finetuning of Quantized LLMs,” in Proc. NeurIPS, 2023.
[16] E. Frantar and D. Alistarh, “GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers,” arXiv:2210.17323, 2022.
[17] S. Han, J. Pool, J. Tran, and W. Dally, “Learning Both Weights and Connections for Efficient Neural Networks,” in Proc. NeurIPS, 2015.
[18] G. Hinton, O. Vinyals, and J. Dean, “Distilling the Knowledge in a Neural Network,” arXiv:1503.02531, 2015.
[19] M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in Proc. ICML, 2019, pp. 6105–6114.
[20] S. Minaee et al., “Deep Learning Based Text Classification: A Comprehensive Review,” ACM Computing Surveys, vol. 54, no. 3, pp. 1–40, 2021.
[21] IBM Research, “Small Language Models for Efficient Edge AI Deployment,” IBM AI Research Whitepaper, 2024.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY