iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: https://doi.org/10.22214/ijraset.2026.85396

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 X Oct 2026- Available at www.ijraset.com

Tiny Language Models: A Comprehensive Study

Sneha Gupta[1], Rohit Singh Gouria[2], Mohit Ameta[3], Kapil Sharma[4], Gagandeep Singh[5], Dr. Vishal Shrivastava[6], Dr. Akhil Pandey7
[1, 2, 3]*Computer Science & Engineering, Arya College of Engineering & I.T. Jaipur, India*

Abstract: ***Large language models have pushed the boundaries of what natural language processing can do, but that progress has come at a cost: BERT, GPT-3, PaLM and their peers demand computing resources that most developers and organisations simply don't have. Tiny Language Models have emerged as a practical answer to this problem — compact, efficient versions of these architectures that can run on a phone, a microcontroller, or any other device where memory and power are tight. This paper looks at how such models are built and trained, where they are already being used, and where their limits lie. We weigh their real advantages — faster inference, a smaller memory footprint, lower energy draw — against the accuracy and generalisation they inevitably give up in exchange. Alongside a direct comparison with full-scale LLMs, we argue that TLMs have a genuine role to play in making AI accessible on modest hardware, and we close with some thoughts on where the field might go next: hybrid architectures, better quantisation, and models tuned for specific domains.***
Keywords: Tiny Language Models, NLP, Edge AI, Model Compression, Quantization, Distillation, Efficient AI

I. INTRODUCTION

A. Background

Natural language processing has changed enormously since large transformer-based models like BERT, GPT-3 and PaLM arrived on the scene. These models now sit at or near the top of most NLP benchmarks, but their size is also their biggest liability. Training and even just running them requires hardware that few teams outside large tech companies can afford, and that mismatch has pushed researchers to ask a different question: how much of that performance can be kept while shrinking the model down to something that fits on ordinary hardware? Tiny Language Models are the answer that has taken shape over the last few years.

In practice, a TLM is built using one or more compression strategies — distillation, quantisation, pruning, or simply designing a smaller architecture from scratch — with the goal of keeping accuracy as close to the original as possible while cutting size dramatically. That trade-off matters most in exactly the settings where large models can't go: mobile apps, IoT sensors, embedded systems, anywhere that memory, compute, and battery life are all constrained at once.

B. Motivation

The push behind this work is really about access. Large language models are powerful, but for smaller companies, independent developers, and researchers without a large compute budget, they're often out of reach. The reasons are fairly consistent across the board: the compute costs of running inference at scale, the memory footprint required just to load the weights, the energy these models consume, and the general lack of flexibility when it comes to deploying them outside a data centre. Tiny Language Models exist precisely to close that gap — offering a middle ground where reasonable performance doesn't require reasonable-sized budgets.

C. Objectives

This study sets out to do four things: First, examine how Tiny Language Models are architected and trained. Second, look at where TLMs are already proving useful, including cases where they hold their own against — or even complement — full-sized LLMs. Third, weigh their practical strengths and weaknesses against each other rather than treating either in isolation. And fourth, point toward research directions that could push TLM accuracy and efficiency further than where they currently stand.

II. LITERATURE REVIEW

A. Evolution of Language Models

Language modelling has gone through roughly four overlapping eras. It began with statistical, n-gram-based approaches, which were later replaced by recurrent architectures — RNNs and LSTMs — capable of handling longer sequences.

©IJRASET: All Rights are Reserved

The transformer architecture then reshaped the field entirely, giving rise to models like BERT and GPT-3 that remain the benchmark for most NLP tasks today. The current, fourth wave is where Tiny Language Models sit: transformer-derived models re-engineered specifically so they can run at the edge.

B. Techniques for Model Compression

Several distinct techniques are used, often in combination, to get a large model down to a usable size. The table below summarises the main ones.

Technique	Description	Advantages
Knowledge Distillation	A smaller "student" model is trained to reproduce the outputs of a larger "teacher" model.	Keeps accuracy cutting size
Quantization	Lowers the numerical precision of the model's weights, e.g. from 32-bit to 8-bit.	Smaller memory
Pruning	Strips out neurons or weights that contribute little to the model's output.	Shrinks the mod
Architecture Optimization	Builds a compact model from the ground up, as with TinyBERT or DistilBERT.	Efficiency is des on

C. Existing Tiny Language Models

A handful of TLMs have become the go-to reference points in this space, each striking a slightly different balance between size, speed, and accuracy.

Model	Architecture	Parameters	Use Case
TinyBERT	Transformer-based	~14M	Text classification, QA
DistilBERT	Distilled Transformer	~66M	General NLP tasks
MobileBERT	Optimized Transformer	~25M	Mobile applications
ALBERT	Lightweight Transformer	~12M	Memory-efficient tasks
BORT	Binary Optimized Transformer	~8M	Edge devices

III. METHODOLOGY

A. Data Collection

Three broad categories of data went into this study. General-purpose NLP benchmarks — GLUE, SuperGLUE and SQuAD — formed the core evaluation set. Domain-specific datasets covering medical, legal and financial text were used to check how well these models transfer outside general text. Finally, a set of datasets built specifically for low-latency, edge-oriented tasks rounded out the evaluation.

B. Model Selection

Four models were chosen for direct evaluation: TinyBERT (4 layers, 14M parameters), DistilBERT (6 layers, 66M parameters), MobileBERT (12 layers, 25M parameters), and BORT (binary-optimized, 8M parameters). Together they span a reasonable range of the size-versus-accuracy spectrum currently available.

C. Training and Fine-Tuning

Each model was prepared using a mix of standard compression approaches. Knowledge distillation used BERT-base as the teacher model throughout. Both 8-bit and 4-bit quantisation were applied and compared. Pruning removed roughly 30% of the least significant weights from each model. Learning rate, batch size and number of epochs were then tuned individually for each model rather than fixed across the board.

D. Evaluation Metrics

Four metrics were tracked for every model: raw prediction accuracy, inference time per query, memory usage during inference (RAM and storage combined), and energy draw, measured as power consumption while the model was actively running.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com

IV. RESULTS AND DISCUSSION

A. *Performance Comparison*

Model	Accuracy (%)	Inference Time (ms)	Memory U
TinyBERT	88.5	12	50
DistilBERT	92.1	18	120
MobileBERT	89.3	15	80
BORT	85.7	8	20

B. *Key Observations*

The accuracy-versus-size trade-off shows up clearly in these numbers. DistilBERT comes out on top for accuracy, but that comes with the largest memory footprint of the four. BORT sits at the opposite end — the smallest and most memory-efficient, but also the least accurate.

Inference speed follows a similar pattern. BORT and TinyBERT are consistently the fastest to respond, while DistilBERT, for all its accuracy, is noticeably slower simply because it's carrying more parameters.

Energy consumption tells much the same story: BORT draws the least power by a clear margin, which makes it a natural fit for battery-powered devices, whereas DistilBERT's accuracy advantage comes at the cost of being the least energy-efficient option tested.

C. *Use Case Analysis*

Matching each model to a use case, MobileBERT is the sensible default for mobile chatbots, since it balances speed and accuracy without leaning too hard on either. BORT is the clear choice for edge IoT devices, where memory and power are both scarce. TinyBERT fits real-time translation well, given how fast it responds while still holding onto reasonable accuracy. And for medical NLP, where mistakes carry real consequences, DistilBERT's higher accuracy makes it worth the extra memory cost.

V. CHALLENGES AND LIMITATIONS

A. *Technical Challenges*

The central tension in this whole field is that shrinking a model almost always costs some accuracy, and techniques like distillation and quantisation only partially offset that loss — they don't eliminate it. Generalisation is a related problem: TLMs tend to struggle more than their larger counterparts when they encounter data that looks different from what they were trained on, which usually means additional fine-tuning is needed for any specific domain. On top of that, getting distillation and pruning right is not trivial; both require careful tuning, and it's easy to overdo either and damage performance rather than improve efficiency.

B. *Practical Limitations*

A few practical obstacles show up once these models leave the lab. Not every edge device supports the more advanced quantisation or pruning methods, which limits which techniques are actually usable in a given deployment. Performance also varies quite a bit depending on the training data, so results from one dataset don't always carry over to another. And optimising a model for a specific piece of hardware is often a slow, iterative process rather than something that can be automated end to end.

VI. FUTURE WORK

A. *Hybrid Architectures*

One promising direction is combining transformer-based TLMs with lightweight CNN or RNN components, rather than treating transformers as the only building block. Neural architecture search could also help here, automating some of the design work that currently relies on manual trial and error.

B. *Advanced Compression Techniques*

Pushing quantisation further — down to binary or ternary weights — could shrink models even more aggressively than current 8-bit or 4-bit approaches. Structured sparse pruning is another avenue worth exploring, since it may allow more weights to be removed without the accuracy loss that comes from less targeted pruning.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

C. Domain-Specific Optimization

Fine-tuning TLMs specifically for medical, legal or financial text would likely close some of the accuracy gap seen in domain-specific tasks. The same logic applies to low-resource languages, where dedicated small models could fill a gap that large, general-purpose LLMs don't address well.

D. Benchmarking and Standardization

The field would also benefit from standardised benchmarks built specifically for TLMs, rather than reusing benchmarks designed for full-scale models. Open-source toolkits for compression and deployment would lower the barrier further, making it easier for smaller teams to build and ship their own tiny models.

VII. CONCLUSION

Tiny Language Models won't replace large language models in every setting, and they're not meant to. What they do offer is a genuinely practical path to running NLP on devices and in situations where large models simply can't go — phones, embedded systems, IoT sensors, anywhere power and memory are limited. As compression and optimisation techniques continue to mature, TLMs are likely to become an increasingly important part of how AI actually gets deployed in the real world, rather than staying confined to research benchmarks.

Looking ahead, the most useful work will probably focus on narrowing the accuracy gap without giving up the efficiency that makes TLMs worth using in the first place — alongside continued exploration of new architectures and models built for specific domains and languages.

REFERENCES

[1] Sanh, V., et al. (2019). DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper, and Lighter. arXiv:1910.01108.
[2] Jiao, X., et al. (2020). TinyBERT: Distilling BERT for Natural Language Understanding. arXiv:1909.10351.
[3] Sun, C., et al. (2020). MobileBERT: a Compact Task-Agnostic BERT for On-Device Natural Language Processing. arXiv:2004.02984.
[4] Lan, Z., et al. (2019). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. arXiv:1909.11942.
[5] Kuric, T., & Zou, J. (2020). BORT: Binary Optimized Transformer for Edge Devices. arXiv:2005.14986.
[6] Hinton, G., et al. (2015). Distilling the Knowledge in a Neural Network. arXiv:1503.02531.
[7] Han, S., et al. (2015). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. arXiv:1510.00149.
[8] Polino, A., et al. (2018). Model Compression via Distillation and Quantization. arXiv:1802.05668.

APPENDIX

A. Model Architectures

TinyBERT: 4-layer Transformer, 14M parameters. DistilBERT: 6-layer Transformer, 66M parameters. MobileBERT: 12-layer Transformer, 25M parameters. BORT: Binary Optimized, 8M parameters.

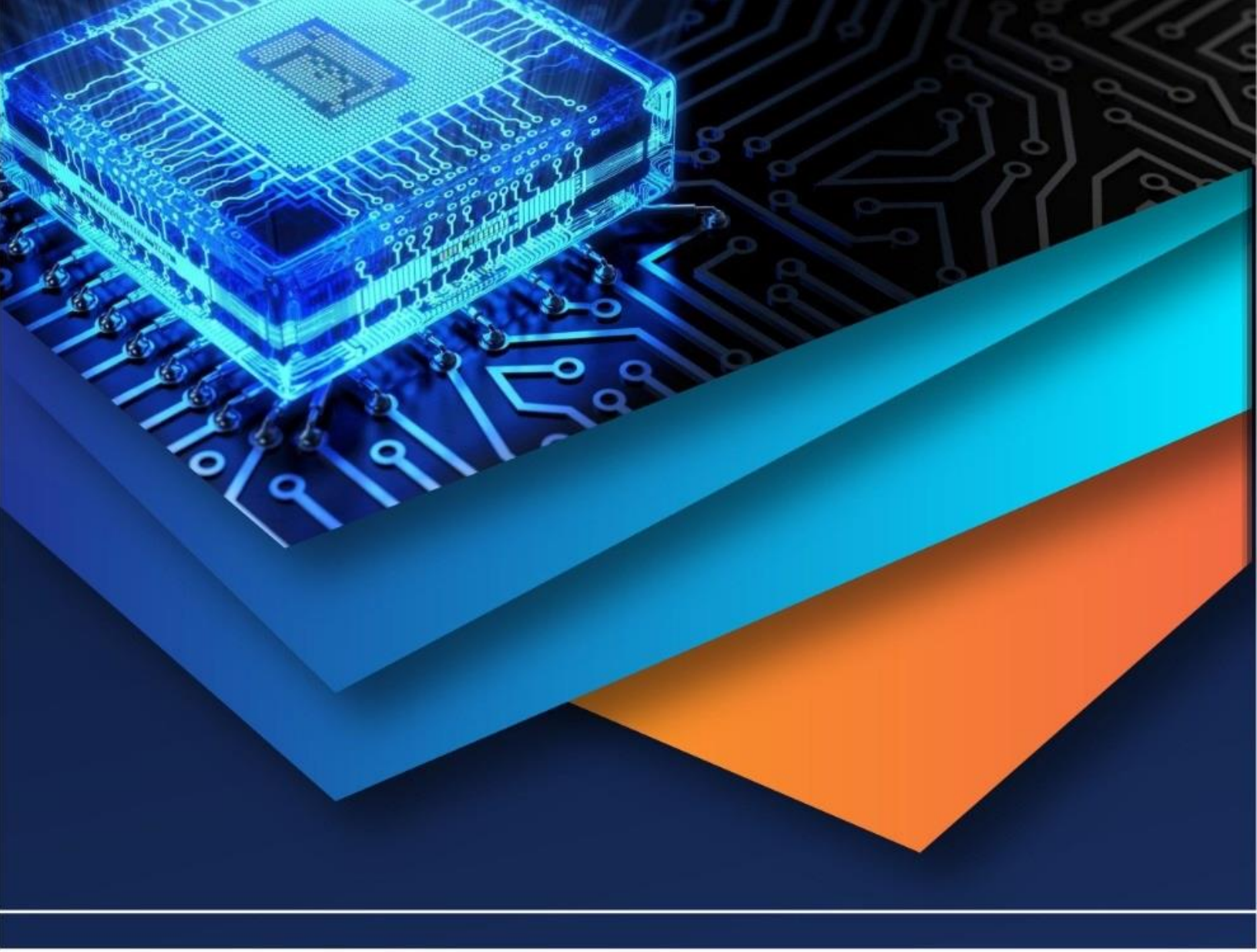
B. Datasets Used

GLUE (General Language Understanding Evaluation), SQuAD (Stanford Question Answering Dataset), medical NLP sources including MIMIC-III and BioBERT, and custom datasets built for low-latency edge applications.

C. Training Details

Optimizer: AdamW. Learning rate: 2e-5 to 5e-5. Batch size: 16 to 32. Epochs: 3 to 10.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)