International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Evaluating the Efficiency of Small Language Models for Applications

Vikram Vilasrao Kadam

Student, Department of MCA, PES Modern College of Engineering, Savitribai Phule Pune university, India

Abstract

This study explores the integration of Large Language Models (LLMs) in smart homes, comparing large models like ChatGPT with smaller ones such as DistilBERT and TinyBERT. While large models excel in accuracy and contextual understanding, smaller models offer advantages in efficiency, privacy, and on-device deployment. The research highlights trade-offs between size and performance, identifying scenarios where compact models outperform larger ones in edge-computing environments.

Purpose: To assess the viability of small LLMs for real-time smart home tasks by evaluating their performance, latency, and resource use in edge-based systems.

Research Question: Can small LLMs effectively perform smart home tasks in terms of speed, accuracy, and efficiency compared to larger models?

Methods: A mixed-methods approach was used, involving a literature review and empirical benchmarking of models (e.g., TinyBERT, Electra-small) on standardized datasets (SuperGLUE RTE, NLU). Optimization techniques like distillation, pruning, and quantization were applied, and performance was measured through accuracy, latency, and loss.

Applications: Small models support real-time voice control, security, energy management, and accessibility in smart homes. They are embedded in devices for tasks like lighting control, environmental adjustments, and emergency alerts.

Challenges: Limitations include weaker contextual understanding, difficulty with complex queries, and reduced accuracy due to compression. Hardware variability and adaptability issues also hinder performance.

Optimization Techniques

- Knowledge Distillation: Transfers knowledge from large to small models.
- Chain-of-Thought: Enhances reasoning via step-by-step training.
- Compression: Pruning and quantization for smaller, faster models.
- UL2R & Flan: Google's fine-tuning methods for generalization.

Results: Smaller models like TinyBERT and DistilBERT showed competitive accuracy with lower latency and training time, making them suitable for smart homes. While RoBERTa and ALBERT performed better overall, the trade-off in resource efficiency favored smaller models in embedded settings.

Future Scope: Future work includes task-specific compact models, hybrid edge-cloud systems, multilingual support, and improved training via synthetic and self-supervised data.

Keywords: Smart Homes, Small Language Models, Edge AI, Knowledge Distillation, Model Compression

1. Introduction

Recent advances in language models like GPT-4 and LLaMA have achieved impressive results in tasks like summarization, translation, and question answering. However, their high computational demands, reliance on cloud infrastructure, and latency issues make them less suitable for everyday use in smart homes. Devices such as smart speakers and sensors often have limited processing power and strict performance constraints.

To address this gap, researchers are developing compact language models that offer efficient performance with lower resource needs. Techniques like knowledge distillation and quantization help reduce model size while retaining effectiveness. This research evaluates the performance of such models in smart home environments, aiming to identify the most practical and efficient solutions for edge deployment.

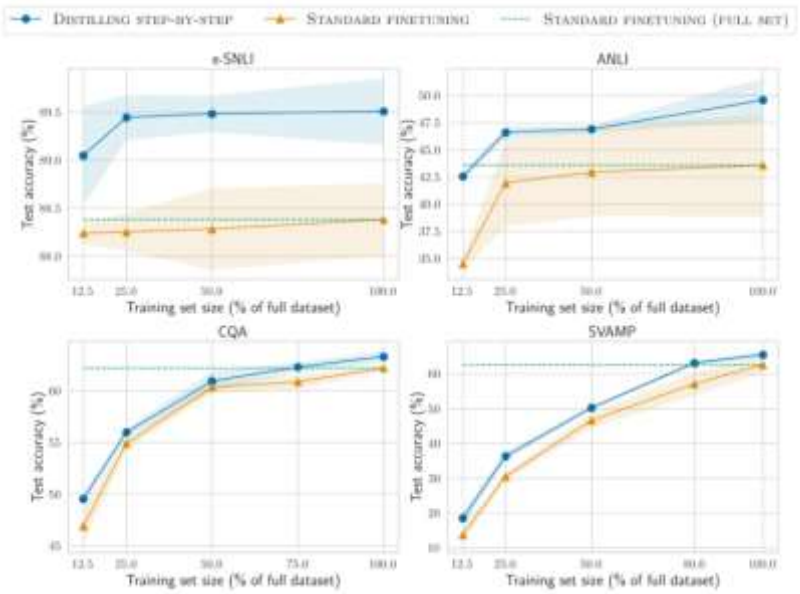
2. literature Survey

The improvement of language models has historically followed a principle that "bigger is better." With models like GPT-3, GPT-4, and LLaMA, the focus was on increasing the number of parameters to improve generalization and performance across multiple NLP tasks. However, the challenge lies in deploying these massive models in real-world, constrained environments such as smart homes.

Recent advancements have demonstrated that smaller models, such as DistilBERT, TinyBERT, and Electra-Small, can maintain near state-of-the-art performance while using a fraction of the computational resources. DistilBERT, for example, is 40% smaller than BERT and runs 60% faster while retaining 97% of its language understanding capabilities. Similarly, TinyBERT is tailored for low-latency environments with high-speed inference and minimal memory consumption.

Optimization techniques play a crucial role in making these small models viable. Knowledge distillation helps transfer the capabilities of large models to smaller ones. Other approaches like model pruning, quantization, and advanced fine-tuning such as UL2R and Flan from Google improve performance without increasing size.

Figure:



Distilling step-by-step achieves better performance than fine-tuning with fewer examples.

Figure:

-	Random	25.0
-	Average human rater	34.5
May 2020	GPT-3 5-shot	43.9
Mar. 2022	Chinchilla 5-shot	67.6
Apr. 2022	PaLM 5-shot	69.3
Oct. 2022	**Flan-PaLM 5-shot**	**72.2**
	Flan-PaLM 5-shot: CoT + SC	**75.2**
-	Average human expert	89.8
	Jun. 2023 forecast (Hypermind)	73.2
	Jun. 2024 forecast (Hypermind)	75.0
	Jun. 2023 forecast (Metaculus)	82.7
	Jun. 2024 forecast (Metaculus)	87.6

Flan-PaLM outperforms ChatGPT and Chinchilla despite being smaller, showing the impact of smart training strategies.

These findings suggest that through effective compression and training, small LLMs are becoming increasingly practical for integration into edge devices in smart homes.

3. Methodology

This research was conducted in two major phases: a literature review and practical experiments involving small LLMs. The goal was to evaluate how efficiently these models perform in smart home-relevant NLP tasks.

Datasets:

- **SuperGLUE RTE**: Recognizing Textual Entailment, testing the model's ability to identify logical relationships between sentence pairs.
- **NLU Evaluation**: Used to simulate real-life smart home dialogues for tasks like intent recognition and sentence classification.

Models Evaluated:

- DistilBERT
- TinyBERT (as Dolphin 2.9.2)
- ALBERT (as Phi 3 Medium)
- RoBERTa-base
- Electra-small

```
model = AutoModelForSequenceClassification.from_pretrained(model_info['
```

Optimization Techniques:

- Knowledge Distillation (KL divergence loss)
- Pruning (removing low-weight parameters using torch.nn.utils.prune)

```
for module in model.modules():
    if isinstance(module, nn.Linear):
        prune.l1_unstructured(module, name="weight", amount=0.2)
```

- Quantization (converting weights to lower precision for efficiency)

```
quantized_model = torch.quantization.quantize_dynamic(
```

```
model, {torch.nn.Linear}, dtype=torch.qint8
```

Experimental Setup: Training and inference were executed using PyTorch and Hugging Face Transformers on the Stanage A100 GPU system. Batch sizes, learning rates, and epochs were adjusted per dataset (e.g., 20 epochs for SuperGLUE, 3 for NLU). Evaluation metrics included training/inference time, accuracy, loss, and model size.

```
start_time = time.time()
trainer.train()                    #exe training process
train_duration = time.time() - start_time #measure time before and after
```

Figure A.1:

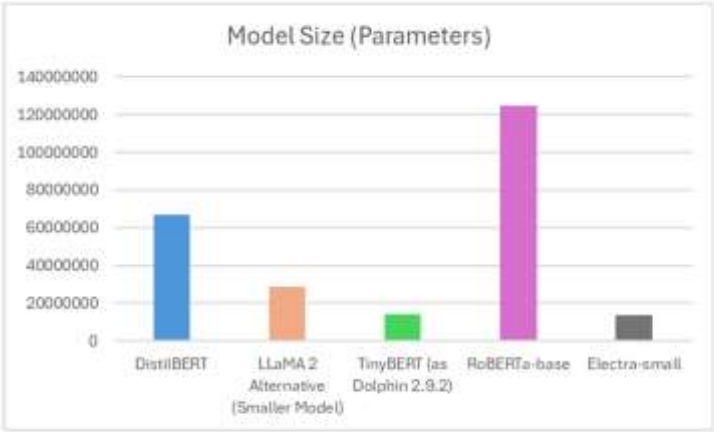


Figure A.1: Model Size.

Model Size Comparison

Figure A.2:

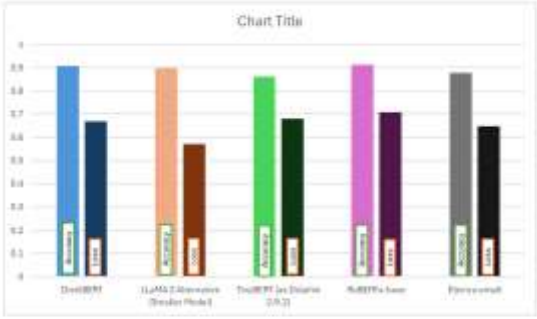


Figure A.2: Model Accuracy and loss

Accuracy vs. Loss Chart

4. Results

SuperGLUE RTE Results:

Model	Training Time (s)	Accuracy	Inference Time (s)	Loss	Parameters
DistilBERT	142.42	0.5964	2.73	2.7286	66M
TinyBERT	42.24	0.6948	1.85	1.4223	14.3M

ALBERT	317.65	0.7912	4.89	1.6491	11.6M
RoBERTa-base	276.65	0.7992	4.92	1.7724	124M
LLaMA 2 Alt	43.36	0.6707	1.73	1.9840	28.7M

NLU Dataset Results:

Model	Training Time (s)	Accuracy	Inference Time (s)	Loss	Parameters
DistilBERT	949.26	0.908	27.5	0.6676	67M
TinyBERT	417.88	0.8604	18.65	0.6796	14.3M
RoBERTa-base	1802.41	0.9123	50.49	0.7079	124M
Electra-small	1053.85	0.8783	48.24	0.6478	13.5M
LLaMA 2 Alt	438.75	0.8979	16.91	0.5708	28.7M

These results validate that smaller models are competitive with larger ones in terms of accuracy while significantly outperforming them in speed and resource efficiency—key criteria for real-time smart home applications.

Figure B.2:

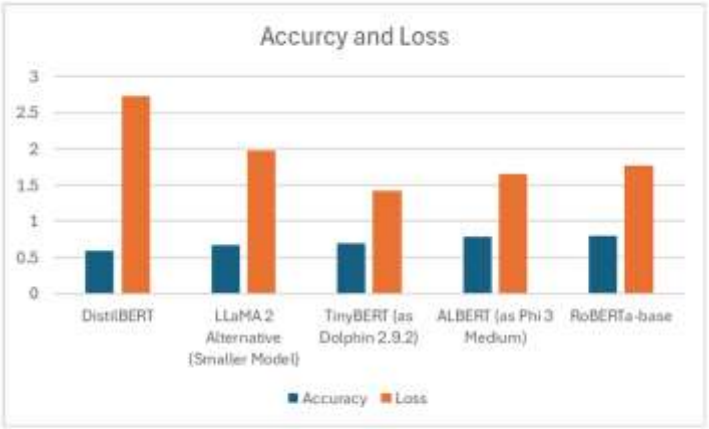


Figure B.2: SuperGLUE: Accuracy and Loss.

Accuracy and Loss Graph: Inference & Training Time Chart

5. Discussion

The practical use of small language models in smart homes hinges on their ability to execute tasks like voice command recognition and control with minimal delay. Models like TinyBERT and Electra-small are ideal for these tasks due to their quick inference times and small size, which makes them deployable directly on devices.

Edge vs. Cloud: Cloud-based models introduce latency and privacy concerns. On-device small LLMs eliminate this by processing locally. However, cloud hybrids could be used for tasks requiring heavy reasoning or long-term memory.

Challenges:

- Generalization to multi-step reasoning or ambiguous inputs is weaker.
- Trade-offs between model size and accuracy must be carefully managed.

- Continuous updates are needed to adapt to evolving user preferences.

Opportunities:

- Training domain-specific models for appliances (e.g., security, thermostat)
- Incorporating federated learning to enable personalization without data sharing

6. Conclusion

This study explored how small LLMs like DistilBERT, TinyBERT, and Electra-small can be efficiently used in smart homes. Despite their reduced size, they retain strong performance across inference tasks. The results support a shift toward compact AI models that can be deployed at the edge. These models are faster, cost-effective, and better aligned with user privacy needs. Future research can explore personalized LLMs, advanced optimization techniques, and cross-lingual support to enhance usability across demographics.

7. References

1. Abdin, M., Jacobs, S. A., Awan, A. A., Aneja, J., Awadallah, A., Awadalla, H., et al. *Phi-3 technical report: A highly capable language model locally on your phone*. arXiv preprint arXiv:2404.14219 (2024).
2. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., et al. *GPT-4 technical report*. arXiv preprint arXiv:2303.08774 (2023).
3. Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., et al. *Scaling instruction-finetuned language models*. Journal of Machine Learning Research 25, 70 (2024), 1–53.
4. Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. *Electra: Pre-training text encoders as discriminators rather than generators*. arXiv preprint arXiv:2003.10555 (2020).
5. Hsieh, C.-Y., Li, C.-L., Yeh, C.-K., Nakhost, H., Fujii, Y., Ratner, A., et al. *Distilling step-by-step! Outperforming larger language models with less training data and smaller model sizes*. arXiv preprint arXiv:2305.02301 (2023).
6. Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., et al. *TinyBERT: Distilling BERT for natural language understanding*. arXiv preprint arXiv:1909.10351 (2019).
7. Kuzmin, A., Nagel, M., Van Baalen, M., Behboodi, A., and Blankevoort, T. *Pruning vs quantization: Which is better?* Advances in Neural Information Processing Systems 36 (2024).
8. Sanh, V., Debut, L., Chaumond, J., and Wolf, T. *DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter*. arXiv preprint arXiv:1910.01108 (2019).
9. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., et al. *Alpaca: A strong, replicable instruction-following model*. Stanford Center for Research on Foundation Models. Retrieved from: https://crfm.stanford.edu/2023/03/13/alpaca.html
10. Tay, Y., Wei, J., Chung, H. W., Tran, V. Q., So, D. R., Shakeri, S., et al. *Transcending scaling laws with 0.1% extra compute*. arXiv preprint arXiv:2210.11399 (2022).
11. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., et al. *LLaMA: Open and efficient foundation language models*. arXiv preprint arXiv:2302.13971 (2023).
12. Turc, I., Chang, M.-W., Lee, K., and Toutanova, K. *Well-read students learn better: On the importance of pre-training compact models*. arXiv preprint arXiv:1908.08962 (2019).