Review | Received 29 November 2025; Revised 4 May 2026; Accepted 5 May 2026; Published 22 June 2026
https://doi.org/10.55092/aiplus20260003

Small Language Models: opportunities and obstacles

Christos Kotrotsios and Manolis Vavalis*

Department of Electrical and Computer Engineering, University of Thessaly, Volos, Greece

* Correspondence author; E-mail: mav@uth.gr.

Highlights:

- SLMs as Enablers of Accessible Generative AI.
- Strong Results through Fine-Tuning and RAG.
- Efficiency without Compromising Usability.
- Mobile Deployment as a Promising Path.
- Privacy and Local Execution.
- Robustness of the Implemented Pipeline.

Abstract: This paper investigates the challenges and opportunities of Small Language Models as efficient, privacy-aware alternatives to Large Language Models in resource-constrained and real-time environments. It elucidates the related basic concepts and combines an up-to-date comprehensive, yet compact, literature review of architectural and optimization techniques for Small Language Models with a systematic experimental evaluation of selected prototype models that integrate fine-tuning, Retrieval-Augmented Generation, and model quantization for multi-platform deployment. It essentially aims to pave the way towards practical implementations that offer measurable improvements over existing methods and can be readily adopted in applied settings.

Keywords: Small Language Models; fine-tuning; LoRA; quantization; Retrieval-Augmented Generation; on-device deployment; Gemma

1. Introduction

Small Language Models (SLMs) have recently emerged as a significant advancement in artificial intelligence, offering efficient and adaptable alternatives to Large Language Models (LLMs). Although LLMs demonstrate impressive potential, high computational costs, resource requirements, and limitations in real-time application development create barriers to their widespread adoption [1]. However, SLMs, with parameters ranging from a few million to billions, incorporate efficiency and flexibility, making them ideal for resource-constrained devices, specialized tasks, and edge computing platforms.

We refer to Small Language Models as language models with substantially fewer parameters than contemporary large-scale models, designed to operate efficiently in resource-constrained or latency-sensitive environments. Some SLMs are designed natively as compact architectures, while others are obtained from larger models through efficiency-oriented optimization techniques. Common model optimization techniques include pruning, which removes less important parameters, and quantization, which reduces numerical precision to lower memory and computational requirements. A related but distinct approach is knowledge distillation, in which a smaller student model is trained to approximate the behavior of a larger teacher model. A notable example is Distilled BERT (DistilBERT), which retains approximately 97% of Bidirectional Encoder Representations from Transformers (BERT)'s performance on the General Language Understanding Evaluation (GLUE) benchmark while using 40% fewer parameters [2].

Beyond compressed versions of existing LLMs, SLMs are also developed as native small architectures designed from the ground up to be efficient, often relying on optimized parameterization and training schemes. Examples include TinyLlama (1.1 B parameters) and Phi-2, which target edge devices and latency-sensitive tasks.

Thus, SLMs represent both a practical adaptation of large-scale models and an independent line of research aiming to balance efficiency with competitive performance in real-world applications.

Our study aims to provide both theoretical understanding and practical implementation by evaluating the performance, efficiency, and suitability of SLMs in realistic usage scenarios.

Although a plethora of review articles on SLMs already exist [3–5], our work addresses critical gaps by providing: (i) a systematic empirical benchmark of quantization strategies applied to SLMs of comparable size, showing that conservative quantization preserves accuracy whereas aggressive compression leads to severe degradation—challenging the common assumption that LLM-optimized quantization methods transfer directly to small models [6]; (ii) a comprehensive evaluation of SLM behavior in a low-resource language (Greek) within a specialized legal domain, a dimension largely absent from existing surveys; and (iii) a fully integrated prototype system that combines fine-tuning, Retrieval-Augmented Generation, and multi-format quantization with validated deployment across desktop and mobile environments. Together, these contributions bridge the gap between theoretical survey work and practical, reproducible SLM deployment in resource-constrained, domain-specific, and privacy-sensitive settings.

Our objectives are to capture the current state of the art in SLMs, to design and implement a prototype system based on a selected SLM, and to empirically evaluate it through indicators such as accuracy, latency, and resource usage. The results are analyzed in terms of model performance and practical deployment trade-offs, and are further discussed in relation to larger language models in domain-specific settings.

We should acknowledge that, primarily due to space constraints, we were unable to discuss certain SLM-related issues, such as the very recently highlighted connection between SLMs and Agentic Artificial Intelligence (AI), for which we refer to [7].

Our paper aspires to draw conclusions and make recommendations for future research, highlighting key improvement points of SLMs and the potential for their integration into broader Generative AI applications.

The rest of this paper is organized as follows. Section 2 presents the background of the study,

outlining the fundamental AI paradigms, the neural architectures that underpin them—particularly attention mechanisms—as well as the progression from LLMs to smaller, optimized SLMs. Section 3 reviews the current state of the art in SLM development and deployment, including open-source libraries, cloud platforms, optimization tools, and hardware-specific frameworks. Section 4 surveys representative peer-reviewed research on SLMs, with emphasis on landmark papers, methodological trends, experimental setups, and current research gaps. Section 5 briefly discusses relevant grey literature sources and their contribution to the broader SLM ecosystem. Section 6 presents the prototype implementation and experimental evaluation, from system design and model selection to fine-tuning, quantization, and RAG-based testing. Finally, Section 7 summarizes the main findings, discusses the limitations of the present study, and outlines directions for future research.

2. Background

Modern AI comprises two major paradigms: Predictive AI and Generative AI. The transition from predictive to generative paradigm marks a fundamental shift: from systems that merely analyze data to those that can create it, introducing new opportunities and challenges in reliability, ethics, and computational efficiency.

Both paradigms utilize Artificial Neural Networks, consisting of interconnected neurons whose outputs are transformed by activation functions. Training employs backpropagation and gradient descent to minimize error.

Early models relied on Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks to model sequences, but these struggled with long-range dependencies and sequential inefficiency. The breakthrough came with the Transformer architecture [8], which processes tokens in parallel and replaces recurrence with the attention mechanism.

At its core, self-attention computes context-aware token representations by assigning dynamic weights to all tokens in a sequence, enabling the model to focus on the most relevant context regardless of position. Multi-head attention extends this by performing multiple attention operations in parallel, capturing diverse dependency types within a sequence. This innovation paved the way for large-scale models, and subsequently for their efficient descendants, the SLMs.

LLMs, grounded in the Transformer architecture, typically comprise billions of parameters and are trained auto-regressively on massive corpora—web data, books, and code—to learn token probability distributions. Models like GPT-4 [9] exhibit remarkable generative and reasoning capabilities, while BERT [10] achieves state-of-the-art results in classification and semantic analysis through bidirectional masked-language modeling. Despite their success, LLMs require enormous computational resources, making their training and deployment costly and environmentally intensive.

2.1. Generative AI and language models

Generative AI constitutes a rapidly evolving branch of AI focused on the automatic generation of original content, including text, images, audio, and video, by learning statistical regularities from large-scale training data [11]. Unlike predictive systems that primarily classify or estimate outcomes, generative models synthesize new outputs that aim to be coherent, contextually relevant, and often difficult to

distinguish from human-produced content [11]. This shift from analysis to generation has significantly broadened the practical scope of AI across multiple domains.

The principal methodological families of Generative AI include Transformers, Generative Adversarial Networks (GANs), and Diffusion Models. Transformers rely on self-attention mechanisms that dynamically model contextual dependencies within sequences and have become the dominant foundation of modern natural language processing [8]. GANs, in contrast, are based on adversarial training between a generator and a discriminator, and have been widely used in synthetic image generation [12]. Diffusion models operate through progressive noise addition followed by learned denoising and have emerged as a particularly effective paradigm for high-fidelity image and multimodal synthesis [12]. Although these paradigms serve different modalities and use cases, together they define the broader landscape of contemporary Generative AI.

Generative AI has already transformed a wide range of applications. In the textual domain, it powers systems for dialogue, summarization, translation, and content generation, while in visual and auditory domains it supports image synthesis, voice generation, and multimodal interaction. At the same time, the rapid proliferation of these systems has raised important ethical, legal, and societal concerns. Deepfakes threaten trust in digital media, the use of large-scale training corpora raises questions of intellectual property and authorship, and latent biases in training data may propagate harmful stereotypes or misinformation. These challenges reinforce the need for careful evaluation, transparency, and responsible deployment.

In the context of the present study, our interest lies primarily in the transformer-based branch of Generative AI, since it underpins modern language models and, by extension, Small Language Models. For this reason, the following section focuses specifically on Large Language Models as the immediate conceptual and technical precursor to SLMs.

2.2. Large Language Models

Large Language Models (LLMs) form the backbone of contemporary natural language processing and have achieved state-of-the-art performance across a broad range of linguistic tasks. Their foundational architecture is the Transformer [8], which replaced recurrent processing with self-attention and thereby enabled more effective modeling of long-range dependencies in text. This design significantly improved scalability and made it possible to train models with billions of parameters on massive text corpora.

At the core of the Transformer lies the self-attention mechanism, through which each token can attend to other tokens in the input sequence and build context-sensitive representations [8]. Together with positional encoding, which injects token-order information into the model, this mechanism provides the structural basis for the fluency, contextual understanding, and generalization ability of modern LLMs [13]. Owing to their scale and pretraining regime, LLMs can support a wide spectrum of tasks, including translation, summarization, classification, reasoning-oriented prompting, and open-ended text generation.

A major strength of LLMs lies in their versatility. Through zero-shot and few-shot prompting, a single pretrained model can adapt to many tasks without explicit retraining, while fine-tuning enables stronger specialization when domain-specific behavior is required. However, this generality comes at a significant computational cost. Training and deploying such models typically require specialized Graphics Processing Unit (GPU)/Tensor Processing Unit (TPU) infrastructure, substantial energy consumption, and large-scale data pipelines, which limit accessibility and increase operational expense [14,15].

Despite their remarkable capabilities, LLMs face several important limitations. Their autoregressive decoding process and large parameter counts can lead to high latency, particularly in cloud-hosted or interactive applications [14]. They are also susceptible to hallucinations, namely the generation of fluent but factually incorrect outputs, which poses serious challenges in high-stakes domains such as law and healthcare [16]. Additional concerns include bias propagation from training data [17], privacy and data security risks [18], adversarial prompting and model manipulation [19], as well as unresolved legal issues related to copyright and AI-generated content [20]. These issues complicate the safe and reliable use of LLMs in real-world environments.

Taken together, the strengths and weaknesses of LLMs motivate the search for more efficient, controllable, and deployable alternatives. In particular, the need for lower latency, reduced cost, stronger privacy guarantees, and operation on constrained hardware has accelerated interest in Small Language Models, which aim to retain useful language capabilities while substantially reducing computational overhead.

2.3. Emergence of SLMs

2.3.1. Motivations behind the rise of SLMs

The rise of SLMs stems directly from the constraints of LLMs, particularly with regard to cost, scalability, energy efficiency, and privacy.

Economic Sustainability: Training large-scale models like GPT-3, with costs ranging between $4.6–$12 million [21], restricts participation to major corporations. SLMs democratize AI by lowering computational and financial barriers for smaller organizations.

Energy Efficiency: The environmental burden of LLMs—requiring up to 1287 MWh for training [15]—has spurred demand for energy-conscious AI solutions. SLMs are optimized for low-power operation, aligning with global sustainability goals.

Privacy and Compliance: As regulations like the General Data Protection Regulation (GDPR) restrict cloud-based data processing, SLMs offer a privacy-preserving alternative by performing inference locally on user devices.

Task-Specific Intelligence: The “bigger is better” paradigm is being replaced by “smaller is smarter.” Domain-optimized SLMs often outperform general-purpose LLMs in specialized applications.

Infrastructure Accessibility: Unlike LLMs requiring super-computing clusters, SLMs run efficiently on local hardware such as smartphones, laptops, or single-board devices (e.g., Raspberry Pi) [22], enabling real-time, offline use.

2.3.2. Architectures and optimization techniques

To maintain high utility while minimizing computational demands, SLMs employ architectural simplifications and advanced optimization strategies as follows and as summarized in Figure 1.

Parameter Reduction: DistilBERT (66 M parameters) reduces size by 40% (relative to BERT) through layer truncation and parameter sharing, and TinyLLaMA (1.1 B parameters) further compresses memory and activations, maintaining strong performance in constrained environments [23].

Specialized Layers: MobileBERT utilizes inverted bottleneck layers [24] to accelerate inference, while MobileLLaMA (0.5 B parameters) employs grouped-query attention to lower latency without

sacrificing accuracy.

Efficient Transformer Designs: Tay *et al.* [25] identifies multiple efficiency strategies, including:

- Sparse attention (e.g. Longformer, BigBird) to reduce quadratic complexity by attending to subsets of tokens.
- Linearized attention (e.g. Performer, Linformer)—using kernel approximations to achieve linear scaling.
- Low-rank factorization—compression of large weight matrices.
- Memory compression—retention of compressed historical states instead of full ones.
- Early exiting—dynamically stopping computation for simpler inputs.

These innovations strike a balance between computational efficiency and model expressivity, enabling practical deployment on mobile and edge devices. For example, Longformer employs sliding window attention with global tokens for document-level tasks, while Performer leverages random feature mappings for scalable, linear attention.

Modular and Hardware-Aware Design: Modern SLMs are often built with hardware-awareness in mind. Architectures like Open Efficient Language Models (OpenELM) and EdgeGPT adopt a modular design, where components such as attention heads or feedforward networks can be selectively activated or pruned based on hardware capabilities and real-time resource availability.

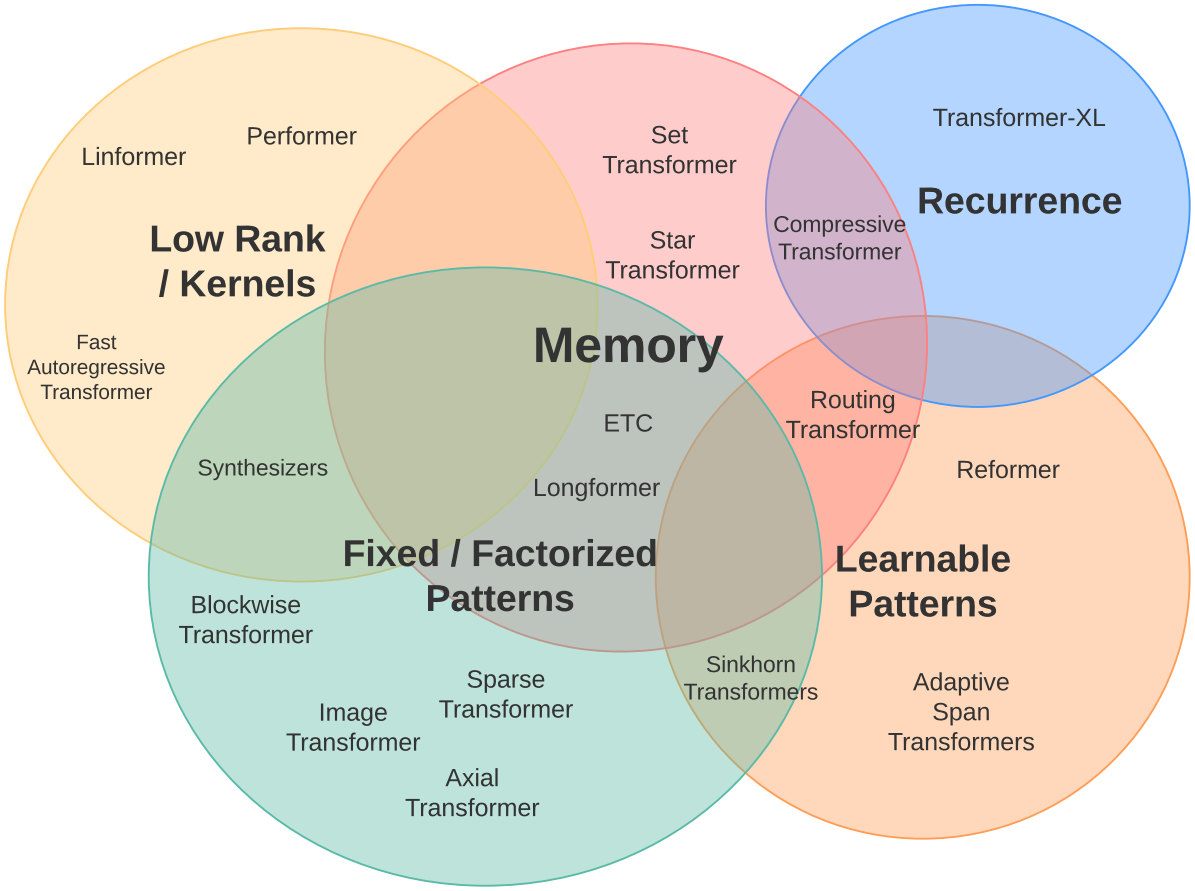

Figure 1. Taxonomy of efficient transformer architectures, illustrating approaches for reducing computational and memory complexity, including sparse and linear attention, low-rank approximations, and hybrid optimizations [25].

2.3.3. Compression techniques

Language model compression relies on several complementary approaches, primarily aimed at reducing memory footprint, inference cost, and deployment requirements without severely degrading performance.

Quantization reduces the numerical precision of model weights—for example, from 32-bit floating-point to 8-bit integers—thereby shrinking model size by up to 4× with limited accuracy loss. Popular frameworks such as TensorFlow Lite and Open Neural Network Exchange (ONNX) Runtime support quantized deployment on mobile and embedded platforms [4]. Within the LLM ecosystem, tools such as Hugging Face's BitsAndBytesConfig further enable 8-bit, 4-bit, and lower-precision quantization, allowing models to run on memory-constrained GPUs without modifying pretrained weights.

Modern quantization techniques [26] seek to balance model size, inference speed, and hardware compatibility. The main representative methods, summarized in Figure 2, include:

- GGML Universal Format (GGUF): a Central Processing Unit (CPU)-oriented quantization format supporting approximately 1.5- to 8-bit precision, widely used with llama.cpp for efficient offline inference and optional hybrid CPU/GPU execution.
- Generalized Post-Training Quantization (GPTQ): a GPU-oriented post-training method, typically operating at 3–4 bits, that minimizes quantization error through approximate second-order optimization.
- Activation-Aware Weight Quantization (AWQ): a 4-bit method that preserves important weights using activation information and is particularly effective for instruction-tuned models on NVIDIA GPUs.
- NF4 and FP4: low-precision formats available through BitsAndBytes, commonly used in CUDA-based workflows and especially relevant to Quantized Low-Rank Adaptation (QLoRA)-style adaptation.
- EXL2: a mixed-precision quantization scheme supporting 2- to 8-bit weights for GPU-focused inference.
- Half-Quadratic Quantization (HQQ): a lightweight method that does not require calibration data and supports both CPU and GPU inference.

Together, these methods illustrate the diversity of quantization strategies available for different deployment environments.

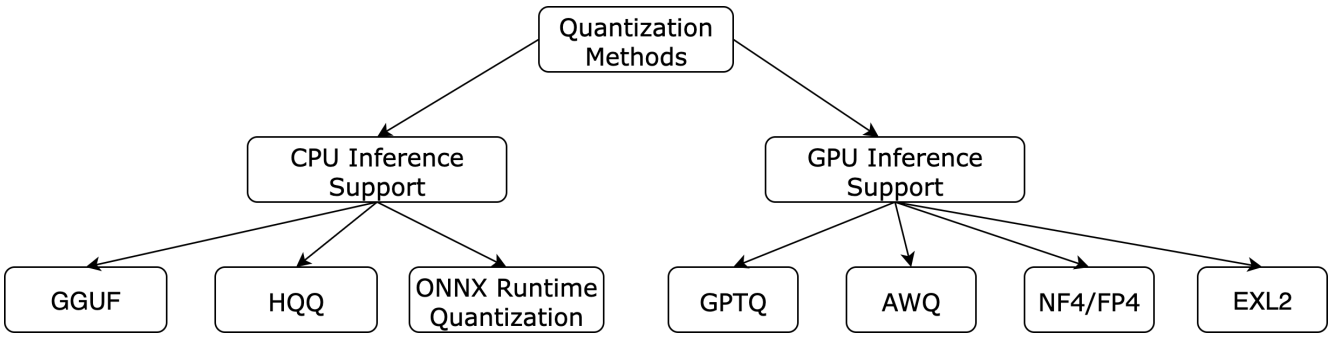

Figure 2. Summary of the quantization methods discussed in the text above.

Pruning removes non-critical neurons, attention heads, or entire layers according to importance criteria. Although pruning may reduce model performance, carefully designed pruning strategies combined with fine-tuning can substantially decrease model size and computational cost while preserving acceptable downstream accuracy. Studies on models such as Large Language Model Meta AI (LLaMA) indicate that significant compression can be achieved without severe degradation when pruning is applied appropriately [27,28].

Knowledge Distillation trains a compact student model to mimic a larger teacher model. Supervision typically combines hard labels from ground truth with soft targets derived from teacher logits, which capture richer inter-class information and often improve generalization [29]. Temperature scaling controls the softness of the teacher distribution:

$$p_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}, \tag{1}$$

where z_i denotes the logit for class i and $T > 1$ produces softer output distributions. Adaptive temperature scaling and Z-score normalization have been proposed to further improve teacher-student alignment [30,31].

The distillation objective is commonly formulated as a weighted combination of cross-entropy with ground-truth labels and Kullback–Leibler divergence between temperature-scaled teacher and student outputs:

$$\mathscr{L}_{\text{total}} = \alpha \cdot \mathscr{L}_{\text{CE}}(y_s, y_{\text{true}}) + (1-\alpha) \cdot T^2 \cdot \mathscr{L}_{\text{KL}}(\sigma(z_s/T), \sigma(z_t/T)). \tag{2}$$

Here, α balances the two terms, while T^2 ensures appropriate gradient scaling. Distillation may be performed in offline, online, or self-distillation settings [32,33]. More recent variants also include ensemble distillation, where multiple teachers are used, as in BabyLlama-2, and feature-based or relation-based transfer schemes that extend beyond output-level imitation [4,32].

Weight Sharing reduces parameters by forcing multiple weights or layers to reuse the same values or representations. In transformer-based models, where redundancy is often substantial, this approach can provide meaningful compression with limited architectural disruption.

2.3.4. Application domains and advantages of SLMs

The practical benefits of SLMs become evident across a variety of domains, where the trade-off between performance and efficiency is not only acceptable but advantageous. Next, we briefly comment on several of these domains.

SLMs tailored to clinical decision support or summarization of electronic health records, are gaining popularity. Compact models trained on medical corpora offer real-time diagnosis suggestions, literature summarization, and patient monitoring, often operating locally to preserve data privacy.

SLMs enable real-time data processing directly on edge devices without the latency or privacy risks associated with cloud transmission. Use cases include air quality monitoring, smart wearables, and home automation systems. By performing inference locally, SLMs avoid transmitting sensitive data to remote servers [22], thus enhancing security and regulatory compliance—especially critical in GDPR-regulated contexts or privacy-sensitive industries.

SLMs have been successfully adopted in microservice architectures and lightweight chatbots. These "Micro LLMs" are capable of delivering fast, cost-effective, and contextually relevant responses to customer queries, reducing the dependency on human agents, and cutting operational costs.

SLMs are increasingly used in personalized tutoring systems, language learning apps, and content simplification tools. Their ability to be fine-tuned in domain-specific educational material allows curriculum-aligned explanations, adaptive feedback generation, and even low-resource language support—all without the infrastructure burden of large-scale models.

SLMs facilitate text simplification, speech-to-text transcription, and real-time translation on devices with limited resources. This is particularly beneficial for individuals with disabilities or users in low-connectivity regions, allowing more inclusive access to AI.

In low-connectivity environments—rural areas, SLMs are used to provide language capabilities locally. Educational tools, for example, can support personalized tutoring offline, adapting to the student's reading level and pacing.

Table 1 outlines the key distinctions between SLMs and LLMs [34] and highlights the complementary nature of SLMs and LLMs. While LLMs are ideal for high-stakes, complex reasoning across domains, SLMs excel in lightweight, targeted deployments with real-time and privacy constraints.

Table 1. Comparison of characteristics between Large and Small Language Models.

	LLMs	**SLMs**
Parameters	10 B–500 B+	100 M– 5 B
Hardware Requirements	High-end GPUs/Cloud clusters	Commodity hardware (laptops, smartphones, edge nodes)
Inference Speed	Slow (due to size)	Fast; suitable for real-time
Cost (Training+Inference)	Very high	Significantly lower
Energy Consumption	High; significant carbon footprint	Low; suitable for embedded systems
Accuracy & Capability	High; excels in multi-domain and complex tasks	Adequate for domain-specific or simpler tasks
Generalization Ability	High; broad applicability	Moderate; task-specific
Privacy	Cloud-based; potential risks	On-device execution; better data control

2.4. SLMs developed and their applications

Early efforts toward lightweight language modeling were largely based on model compression. Typical examples are: DistilBERT [2] represents one of the first systematic attempts to reduce the size of the model through knowledge distillation, achieving near-BERT performance with 40% fewer parameters.

TinyBERT [35] further optimizes for edge devices and IoT environments, balancing accuracy with low latency.

A Lite BERT (ALBERT) [36] achieves parameter efficiency through weight sharing and factorized embeddings, making it ideal for constrained conversational or classification systems.

Recent years have witnessed the rise of standalone SLMs, designed from scratch rather than derived solely through compression. These models incorporate architectural innovations and curated training pipelines that enable competitive performance at a fraction of the computational cost. Representative examples include:

Phi-4 Family (Microsoft, 2024–2025): Including Phi-4-mini (3.8 B) and Phi-4-multimodal (5.6 B), this family exemplifies a new generation of compact models trained on high-quality, reasoning-oriented corpora. These models achieve strong performance relative to much larger systems while remaining deployable on standard hardware, with the multimodal variant leveraging Mixture-of-LoRAs for efficient specialization [37,38].

Gemma 3 (Google DeepMind, 2025): Available in compact variants such as 1 B and 4 B parameters, Gemma 3 combines memory-efficient attention, hardware-aware scaling, and native support for low-bit inference, facilitating deployment on commodity devices [39].

Qwen2.5 (Alibaba, 2024–2025): Qwen2.5 introduces efficient multilingual decoder-only architectures with strong reasoning and tool-use capabilities, and has emerged as one of the most competitive open-access families in the small-to-medium parameter range [40].

OpenELM (Apple, 2024): With model sizes ranging from 270 M to 3 B parameters, OpenELM is explicitly optimized for edge and mobile execution through layer-wise scaling and a transparent training and inference design [41].

TinyLLaMA and BabyLLaMA: Building on the LLaMA family of decoder-only transformers, these compact LLaMA-derived models illustrate the effectiveness of targeted distillation, pruning, and efficient pretraining for resource-constrained deployment scenarios [4,5].

Mistral 7 B and Mixtral (2024): Although not strictly small in parameter count, these efficiency-oriented models are relevant reference points because grouped-query attention, sliding-window attention, and sparse expert activation (in Mixtral 8×7B) demonstrate how architectural design can substantially reduce deployment costs in practice.

Modern SLMs rely on multiple, often complementary, advanced optimization paradigms that extend beyond mere pruning or distillation:

Quantization-Aware Training (QAT): Recent architectures, such as Gemma 3 and related small reasoning models, increasingly incorporate quantization during training or provide dedicated QAT checkpoints, enabling efficient 4–8-bit inference with limited accuracy degradation. This facilitates native deployment on edge accelerators and embedded GPUs.

Sparse Mixture-of-Experts (MoE): Sparse activation via MoE significantly reduces inference costs by enabling only a subset of experts during each forward pass. Models like Mixtral demonstrate that using 2 out of 8 experts retains strong performance while operating at SLM efficiency levels.

Low-Rank Adaptation (LoRA) and QLoRA: LoRA introduces lightweight, low-rank trainable matrices within attention layers to enable parameter-efficient fine-tuning. QLoRA further combines this with quantization, supporting effective training on 4-bit weights and enabling on-device personalization.

Efficient Attention Mechanisms: Techniques such as FlashAttention v2 and Grouped-Query Attention (GQA) improve memory locality and computational throughput without sacrificing attention quality, now standard in most recent SLM architectures.

Weight Sharing and Tied Embeddings: First applied in ALBERT and later adopted by Phi-3/4 models, layer-wise weight sharing and embedding tying minimize parameter count with minimal accuracy loss, contributing significantly to architectural compactness.

A comparative summary of recent SLMs is given in Table 2, highlighting their parameter size, key optimization strategies, and application scope:

Table 2. Indicative comparison of representative recent SLM families.

Model	Size	Core Techniques	Target Use Cases
Phi-4 Mini	3.8 B	High-quality data, efficient decoder, GQA	On-device agents, reasoning tasks
Gemma 3-1B	1 B	Native quantization support, efficient attention, instruction tuning	Edge NLP tasks, summarization
Qwen2.5-3B	3 B	Efficient multilingual decoder, tool use, quantization support	Multilingual assistants, reasoning, chat
OpenELM 1.1B	1.1 B	Layer-wise scaling, transparent training, hardware-aware design	iOS apps, on-device inference
TinyLLaMA 1.1B	1.1 B	Efficient pretraining, LLaMA-derived design	General lightweight NLP, local deployment
BabyLLaMA	0.345 B	Ensemble distillation from two teachers, limited-data pretraining	Resource-constrained mobile devices

It is worth pointing out that recent developments in SLMs mark a paradigm shift in generative AI—from maximizing scale towards optimizing efficiency, modularity, and deployability. Through innovations such as quantization-aware training, sparse activation, and parameter-efficient fine-tuning, SLMs achieve a favorable trade-off between performance and computational cost. This evolution not only enables broader accessibility and sustainability, but also unlocks new applications across privacy-sensitive, real-time, and edge-oriented environments. Specifically,

- SLMs are increasingly integrated into federated learning frameworks, allowing decentralized training with strong privacy guaranties and minimal data transmission [42].
- Initiatives such as SMLBench and EdgeEval aim to standardize evaluation metrics—latency, throughput, energy use, and few-shot learning—establishing transparent baselines for comparing efficiency-oriented models.
- Specialized SLMs (e.g., LegalLLaMA, BioMedTiny, TutorBot-mini) are fine-tuned for vertical domains, prioritizing interpretability and domain compliance over raw generalization power.
- The open release of models like Phi-3-mini, TinyLLaMA, and Gemma fosters reproducibility, accessibility, and collaborative experimentation, substantially broadening participation in SLM research.

3. Small Language Models—state of the art

SLMs have rapidly progressed from experimental prototypes to practical tools integrated into production systems. This section reviews the current landscape of SLM research and deployment, with an emphasis on open-source libraries, cloud infrastructure, and optimization tool-chains that enable efficient inference under resource constraints. The analysis highlights strengths, limitations, and emerging trends that inform the design and evaluation choices presented in later sections.

The SLM ecosystem has expanded considerably, supported by open-source frameworks, hardware accelerators, and cloud-based services. These components collectively enable training, optimization, and inference across diverse environments—from embedded devices to large-scale clusters. The following subsections outline the main software and hardware foundations currently supporting SLM development and deployment.

3.1. Open-source libraries

Open-source software has played a central role in the rapid development and practical deployment of SLMs. Hugging Face Transformers remains the dominant framework for model loading, fine-tuning, and inference across PyTorch and TensorFlow, while also supporting export to runtimes such as ONNX and TensorFlow Lite and providing utilities for knowledge distillation and parameter-efficient adaptation [43]. Owing to its extensive model hub and unified Application Programming Interface (API), it has become a common entry point for both research experimentation and applied SLM development.

For efficient inference, several complementary runtimes have emerged. ONNX Runtime provides hardware-agnostic deployment across CPU, GPU, Field-Programmable Gate Array (FPGA), and edge devices, whereas NVIDIA TensorRT (TensorRT)-LLM targets NVIDIA hardware through fused kernels, mixed-precision execution, and dynamic-shape optimization, making it particularly suitable for latency-sensitive serving. In low-resource and privacy-preserving settings, a tensor library by Georgi Gerganov (GGML)/its successor file format (used in llama.cpp) (GGUF), llama.cpp, and Ollama have become especially important, enabling lightweight local execution of quantized models on desktops, laptops, and embedded systems. Machine Learning Compilation (MLC)-LLM further extends this landscape by using the ATensor Virtual Machine (TVM) compiler stack to generate optimized inference code for Android, iPhone Operating System (iOS), WebAssembly, and related edge targets, thus supporting mobile and cross-platform deployment without heavyweight dependencies.

Additional tools further broaden the ecosystem. LangChain supports modular pipeline orchestration for retrieval, tool use, and application-level integration, while TensorFlow Lite and Core Machine Learning (ML) facilitate deployment on mobile and Apple hardware, respectively, with support for quantized and hardware-accelerated inference. For fine-tuning and scalable adapter serving, Axolotl streamlines LoRA- and QLoRA-based adaptation, whereas LoRA eXtended/LoRA Server (LoRAX) enables efficient multi-LoRA inference on top of shared base models. Taken together, these libraries form a mature tooling stack that supports the full SLM lifecycle, from experimentation and compression to local deployment and production serving.

3.2. Cloud platforms

Deploying SLMs in cloud and edge environments is a key enabler for scalable and latency-aware AI systems. Major providers offer specialized hardware, software stacks, and orchestration frameworks that support efficient inference under diverse deployment constraints. This subsection highlights representative platforms and their relevance for SLM-oriented workloads.

AWS Inferentia is AWS 's custom accelerator for high-throughput and cost-efficient inference, supporting frameworks such as TensorFlow, PyTorch, and ONNX. Through the Neuron SDK and SageMaker ecosystem, it can accelerate quantized and pruned SLMs in cloud production settings, although it is oriented more toward large-scale managed inference than highly constrained edge deployment.

Google provides two TPU-based options targeting different operational scenarios. Cloud TPUs support large-scale transformer workloads through native TensorFlow and JAX integration, as well as deployment within Vertex AI and Kubernetes-based environments. By contrast, Edge TPU (Coral) targets low-power, on-device inference through TensorFlow Lite, making it suitable for compact and

privacy-preserving SLM deployment on embedded systems.

Azure Machine Learning Edge functions primarily as an orchestration and deployment layer rather than a dedicated accelerator. Through ONNX Runtime, automated optimization, and containerized delivery via Azure IoT Hub, it enables SLM execution across heterogeneous edge devices. Features such as Continuous Integration (CI)/Continuous Deployment (CD) integration, monitoring, and drift detection make it attractive for production-grade real-time systems.

The NVIDIA Jetson family (e.g., Nano, Xavier NX, and Orin) combines compact GPU hardware with TensorRT-based optimization, including FP16 and INT8 inference, and supports ONNX-, PyTorch-, and transformer-based SLM deployment. Owing to its balance between performance, power efficiency, and software maturity, Jetson has become a widely adopted platform for robotics, edge analytics, and smart-device applications.

Finally, IBM Watson Machine Learning supports ONNX- and Predictive Model Markup Language (PMML)-based inference in hybrid cloud–edge settings, with emphasis on optimization, auditability, and secure deployment. Although less prominent than the largest cloud ecosystems, it remains relevant in specialized and regulated environments where governance and model traceability are especially important.

3.3. Model optimization tools

Efficient SLM deployment depends not only on model compression, but also on runtimes and compilers that reduce latency, memory usage, and energy consumption during inference. This subsection outlines representative optimization toolchains commonly used for transformer-based execution.

vLLM is a high-throughput inference engine designed for efficient serving of transformer models. Its core PagedAttention mechanism manages key–value caches more effectively, reducing memory overhead in multi-user settings, while continuous batching further improves throughput. Native LoRA support also makes vLLM attractive for serving fine-tuned SLMs.

SGLang combines a domain-specific language with an optimized runtime for LLM applications. It supports dynamic batching, parallel execution, and speculative decoding, while its "prompt-as-code" abstraction simplifies the implementation of more complex inference pipelines.

Open Visual Inference and Neural Network Optimization (OpenVINO) targets efficient inference on Intel CPUs, Vision Processing Units (VPUs), and integrated GPUs, and increasingly supports transformer-based models such as DistilBERT and TinyBERT. Its optimization pipeline includes techniques such as 8-bit Integer (INT8) post-training quantization, layer fusion, and vectorized execution, making it relevant for low-power edge and industrial settings.

TensorRT is NVIDIA's inference optimization framework for GPU deployment. By combining kernel auto-tuning, layer fusion, and mixed-precision execution (e.g., FP16 and INT8), it is widely used for latency-sensitive transformer inference on NVIDIA hardware.

ONNX Runtime provides hardware-agnostic execution across CPUs and multiple accelerator backends, including Compute Unified Device Architecture (CUDA), TensorRT, and DirectML. Through graph optimization, operator fusion, and backend-specific acceleration, it offers a portable solution for deploying lightweight transformer models across heterogeneous environments.

Table 3 provides an indicative example of the latency improvements that optimized runtimes can offer across hardware platforms. Although the benchmark is based on a classical machine learning workload rather than a transformer model, it still illustrates the broader benefit of efficient execution backends for resource-constrained deployment.

Table 3. Indicative inference performance across execution backends and platforms.

Model	Platform	Inference Time (ms)	Speedup
Scikit-Learn	Desktop	60	1 (baseline)
ONNX Runtime	Desktop	4.2	14
ONNX Runtime	Raspberry Pi 4	7.5	8

Overall, these frameworks show that runtime-level optimization is a critical complement to architectural compression, especially when SLMs are deployed under strict hardware and latency constraints.

3.4. Framework and hardware-specific optimizations

Beyond general-purpose runtimes, major deep learning frameworks and specialized accelerators offer built-in mechanisms that further enhance SLM efficiency. These optimizations are typically applied during export or deployment and often serve as a preparatory step before targeting lightweight runtimes.

PyTorch integrates graph-level optimizations (torch.compile), operator fusion, and quantization-aware training (QAT), enabling reduced-precision execution (INT8/FP16/BF16). In combination with pruning and sparsity, these mechanisms lower latency and model size while providing a strong foundation for exporting optimized SLMs to ONNX, TensorRT, or other backends.

Hugging Face Optimum offers a unified interface for exporting and tuning models for ONNX, OpenVINO, and TensorRT. It streamlines quantization, pruning, and hardware-specific adjustments, simplifying deployment across heterogeneous devices even though it does not automatically apply all optimizations.

Apple's Neural Engine (ANE), accessed via Core ML, delivers low-latency on-device inference for quantized and pruned transformers on Apple Silicon. It provides efficient execution for mobile applications, although its proprietary nature limits portability beyond the Apple ecosystem.

Google EdgeTPU is a low-power accelerator optimized for INT8 workloads. Although originally designed for vision models, it supports compact transformer architectures converted to TensorFlow Lite, enabling energy-efficient SLM inference in IoT, robotics, and embedded deployments.

3.5. Summary

Software-level tools such as vLLM, SGLang, OpenVINO, ONNX Runtime, and TensorRT, together with accelerators like EdgeTPU and ANE, form the core ecosystem for efficient SLM deployment. Framework-embedded optimizations in PyTorch and Hugging Face Optimum bridge model development and hardware execution by integrating quantization, pruning, and fusion techniques directly into the workflow.

vLLM excels in memory-efficient, high-throughput inference via PagedAttention, while SGLang offers flexible, latency-aware execution. OpenVINO targets Intel hardware with INT8 quantization and

vectorized CPU optimization, and the ONNX Runtime provides broad cross-platform compatibility. TensorRT and EdgeTPU highlight the role of specialized accelerators, whereas PyTorch and Optimum simplify the export of optimized models. Core ML supports efficient transformer execution on Apple Silicon through the ANE.

Together, these toolchains enable practical and scalable SLM deployment in cloud, edge and mobile environments, underscoring the importance of aligning lightweight architectures with advanced optimization frameworks.

4. Peer-reviewed articles on Small Language Models

Next we provide a comprehensive overview of peer-reviewed research that focuses on SLMs efficiency, optimization techniques, and evaluation. Although SLMs are designed to offer lightweight and resource-efficient alternatives, their development is intrinsically linked to the evolution of large-scale models such as BERT [10], GPT-3 [9], and the Transformer architecture itself [8]. These foundational models have demonstrated exceptional performance across natural language processing tasks, but their computational and memory demands pose significant challenges for deployment in constrained environments.

SLMs aim to retain much of this performance while drastically reducing model size, inference latency, and energy consumption. The literature reviewed here covers key efficiency-oriented approaches, including compression techniques such as quantization and pruning, as well as teacher-student training paradigms such as knowledge distillation. It also examines the experimental setups and benchmark results reported in leading conferences and journals.

4.1. Selected landmark papers

A number of landmark contributions have shaped the design space of Small Language Models by addressing complementary aspects such as compression, sparsity, reasoning, and deployment. Rather than reviewing each paper in isolation, we highlight here the main thematic directions—distillation-based compact models, sparsity and pruning, survey and measurement work, application-specific studies, and reasoning-oriented architectures—and refer to representative works for each.

A first line of work focuses on knowledge distillation and compact BERT-style models. DistilBERT [2] shows that a student model can retain most of BERT's performance while reducing parameter count and inference cost, and TinyBERT [35] extends this paradigm with layer-wise distillation, matching intermediate representations between teacher and student. MobileBERT [24] further adapts these ideas to resource-constrained devices by introducing bottlenecked architectures and quantization-aware optimization. Together, these works establish distillation and architectural simplification as effective tools for deriving smaller yet competitive models from large-scale transformers.

A second strand of research investigates sparsity and pruning as primary mechanisms for efficiency. Solution Path Pruning (SPP) [44] formulates pruning via differential equations and enables arbitrary sparsity levels in a single pass, while TranSpa [45] explores structured sparse training that is explicitly aligned with sparse hardware. In federated settings, FedSpaLLM [46] combines adaptive mask expansion and layer sampling to achieve global sparsity targets across heterogeneous clients. Taken together, these

methods show that transformer models can often be pruned to high sparsity levels while maintaining accuracy close to dense baselines on a range of NLP benchmarks, and that sparsity can be treated as a design principle rather than a purely post-hoc compression step.

Beyond individual techniques, several surveys and measurement studies provide higher-level perspectives on SLMs. Subramanian *et al.* [4] and Nguyen *et al.* [47] offer comprehensive taxonomies of architectures, training regimes, application domains, and hardware deployment, emphasizing the trade-offs between size, performance, and computational cost. Tay *et al.* [25] survey efficient transformer variants more broadly, covering sparse attention, low-rank factorization, kernel-based methods, and quantization, and thus contextualize SLMs within the wider landscape of efficient sequence models. Lu *et al.* [48] complement these works with extensive empirical measurements, including latency and memory profiles, and tools for analyzing performance trade-offs.

Another important set of contributions examines application- and domain-specific behavior. Li *et al.* [49] study SLMs in interactive environments such as productivity tools and dialog systems, showing that instruction-tuned, context-grounded small models can deliver high utility with significantly lower latency and cost. Lepagnol *et al.* [50] evaluate SLMs in zero-shot classification across multiple domains, finding that, with appropriate training and prompt design, small models can approach the accuracy of much larger systems. Hillier *et al.* [51] demonstrate the feasibility of ultra-small models on microcontrollers, relying on extreme quantization and aggressive pruning, and thereby pushing SLMs into severely resource-limited settings. Although not strictly focused on SLMs, studies such as Martin *et al.* [52] on legal reasoning highlight the potential impact of language models in high-stakes domains—a dimension we revisit in our legal case study in Section 4.

A growing body of work targets the reasoning capabilities of compact models. Srivastava *et al.* [53] systematically evaluate deductive, abductive, and common-sense reasoning under constrained parameter budgets, showing that curriculum learning and chain-of-thought prompting can significantly narrow the gap to larger models. In particular, their experiments indicate that, when trained or prompted with explicit reasoning traces and carefully designed task curricula, small models can substantially improve their performance on mathematical and logical reasoning benchmarks relative to naïvely trained baselines. Recent technical reports on small, modern architectures such as Phi-4 [37], Phi-4-Mini [38], Qwen2.5 [40], Gemma 3 [39], and OpenELM [41] further reinforce this trend. They show that careful data curation, instruction tuning, and alignment (often via Reinforcement Learning from Human Feedback (RLHF)) can yield small models with strong reasoning performance, multilingual capability, and real-time inference on commodity or edge hardware.

Overall, these landmark works collectively illustrate how SLMs have evolved from distilled versions of large encoders to a diverse ecosystem of architectures and training pipelines that explicitly target sparsity, reasoning, deployment constraints, and real-world utility.

4.1.1. Comparative overview of selected SLM landmark papers

Table 4 summarizes the key characteristics of selected SLM publications discussed in this section, organized by focus area. For each work, we highlight its main strength, a representative limitation, and one notable result.

This comparative view facilitates a deeper understanding of how different research initiatives approach the challenge of building capable small-scale language models, revealing both common strategies and divergent design choices. Importantly, the field is now moving beyond compressing existing large models: emerging work such as OpenELM and TinyLLaMA illustrates a shift towards architectures and training pipelines that are natively designed for efficiency. This convergence of compression methods with purpose-built small models highlights a new phase in the democratization of language intelligence, where practical deployment is prioritized alongside state-of-the-art performance.

Table 4. Key characteristics of selected SLM publications grouped by focus area.

Focus Area	Publication	Strength	Limitation	Notable Result
Pruning	**Ding *et al.* (SPP)**	Single-pass structured pruning framework that can target arbitrary sparsity levels using a differential-inclusion formulation.	Evaluated mainly on language modeling and standard NLP benchmarks; long-context and highly compositional reasoning workloads are not explored in depth.	Reports that LLaMA2-7B can be pruned to very high sparsity (around 90%) with limited accuracy degradation relative to the dense model on common NLP tasks.
Structured sparse training	**Xiao *et al.* (TranSpa)**	Hardware-aware structured sparse training that produces highly compressed transformers with efficiency gains in both training and inference.	Focuses on efficiency and sparsity; the impact on downstream reasoning-centric tasks is not extensively analyzed.	Achieves substantial speedups in training and inference while roughly halving model size for GPT-2- and LLaMA-style models, with minimal loss in perplexity.
Pretraining from scratch	**Nguyen *et al.* (TinyLLaMA)**	Compute-efficient full pretraining of a 1.1B-parameter transformer tailored to small-scale budgets.	Underperforms larger contemporary LLMs on challenging reasoning and multilingual benchmarks.	Shows that a 1.1B model can achieve competitive performance on general NLP benchmarks under significantly reduced training compute compared to larger models.
Application- and domain-specific behavior	**Li *et al.*; Lepagnol *et al.*; Martin *et al.***	Show that small models can be tailored to interactive tools and multi-domain classification, while related work on legal reasoning illustrates the potential of language models in high-stakes domains.	Often restricted to specific domains or controlled evaluation setups; robustness to noisy, real-world deployments remains an open question.	Indicate that instruction-tuned and context-grounded SLMs can deliver high utility at lower latency and cost in practical applications, and that language models more broadly can support preliminary legal analysis.
Reasoning in compact models	**Srivastava *et al.***	Systematically evaluate deductive, abductive, and common-sense reasoning under tight parameter budgets, comparing training and prompting strategies.	Focused on benchmark suites; long-term robustness and transfer to real-world reasoning tasks are less explored.	Show that curriculum learning and chain-of-thought prompting can significantly improve reasoning performance of small models relative to naïvely trained baselines.
Reasoning with synthetic data	**Phi-4**	Strong reasoning performance at relatively small parameter scales via curated and synthetic training corpora.	Multilingual coverage remains limited; early releases have constrained or staged open-weight availability.	Technical reports indicate that Phi-4 variants outperform GPT-3.5 on MMLU and related reasoning benchmarks at markedly lower parameter counts.
Layer-wise scaling and open infrastructure	**OpenELM**	Modular, layer-wise-scaled architecture with fully open training pipeline, weights, and data curation, enabling high reproducibility.	Reported gains appear to plateau beyond ≈ 1.1 B parameters within the family, suggesting diminishing returns at larger sizes.	OpenELM-1.1B is reported to outperform OLMo-1B on MMLU while retaining an edge- and mobile-oriented design and transparent training setup.
Mixture of Experts (MoE)	**Qwen2.5-MoE**	Sparse MoE design that achieves high task performance with effective parameter utilization by activating only a subset of experts per token.	Requires specialized routing and serving infrastructure, making it less suitable for highly resource-constrained or purely offline edge devices.	Results show that Qwen2.5-MoE variants can match or approach larger dense LLMs on several reasoning and chat benchmarks while maintaining parameter efficiency.

4.1.2. Critical insights and gaps

Although the reviewed works collectively highlight various directions in SLM research, from architectural pruning [45] to application-focused case studies [49,52], certain limitations persist. For example, although many models achieve strong performance on standard benchmarks, few examine robustness under distribution shifts or complex long-context reasoning. Moreover, only a subset (e.g., Phi-4, OpenELM) prioritizes transparency and reproducibility, which are crucial for community validation. In particular, surveys such as [4,47] present comprehensive taxonomies, but the consensus on evaluation standards remains fragmented. In general, the landscape reflects a shift towards smaller, specialized, application-sensitive, or domain-specific models [54], although the trade-off space between size, generalization, and safety is still under active exploration.

4.2. Methodologies and experimental setups

The landmark studies in the domain of SLMs showcase a rich variety of methodologies aimed at improving efficiency while preserving useful task performance. This section summarizes the main methodological trends and experimental paradigms reflected in the selected works.

4.2.1. Knowledge distillation and layer alignment

A recurring theme in the development of compact language models is knowledge distillation, where a smaller student model learns to approximate the behavior of a larger teacher. DistilBERT [2] and TinyBERT [35] are canonical examples, the latter incorporating layer-wise distillation by aligning hidden states and attention maps for improved fidelity. These models are trained on downstream tasks using soft-label supervision from the teacher, often enhanced by task-specific fine-tuning datasets.

Knowledge distillation and layer alignment remain dominant strategies, but their effectiveness depends strongly on the quality and domain match of the teacher model. While methods such as TinyBERT produce compact and efficient models, their generalization beyond the teacher's domain may deteriorate, raising concerns for open-ended deployment.

4.2.2. Architectural optimizations for efficiency

Several models integrate architectural innovations to reduce computational footprint. MobileBERT [24] employs bottleneck structures, inverted residuals, and quantization-aware training to suit mobile devices. Qwen2.5 and Gemma 3 introduce decoder-only efficient transformer architectures optimized for inference-time performance and multilingualism, while Phi-4-Mini incorporates Mixture-of-LoRAs, enabling low-rank adaptation layers for parameter-efficient tuning.

These architectures often prioritize compatibility with edge hardware, with some models explicitly evaluated on mobile chipsets or GPUs supporting sparsity-aware execution. Overall, such optimizations provide promising deployment gains, although many remain architecture-specific and only partially transferable across model families.

4.2.3. Pruning and sparsity-aware training

Another dominant methodology is structured pruning and sparsity. SPP [44] formulates pruning as an optimization problem via differential inclusions, achieving arbitrary sparsity levels in a single pass. TranSpa [45] explores structured sparse training, integrating sparsity constraints during training to produce hardware-friendly models. FedSpaLLM [46] introduces federated pruning schemes in which sparsity masks are learned collaboratively across devices, with local sampling and global coordination.

These methods can substantially reduce memory and computation requirements, often with only moderate degradation in downstream performance. However, their practical benefits still depend heavily on hardware support and on the stability of sparse optimization during training.

4.2.4. Experimental benchmarks and evaluation strategies

Experimental evaluation across studies relies heavily on standardized NLP benchmarks such as GLUE, Stanford Question Answering Dataset (SQuAD), Multi-Genre Natural Language Inference (MNLI), and oolean Questions (BoolQ) for general language understanding. Domain-specific benchmarks such as LegalBench [49,52] test LM performance in niche applications such as legal reasoning and interactive environments. Authors in [48,53] assess reasoning capabilities, including deductive, abductive, and common-sense reasoning, using structured evaluations and chain-of-thought prompting.

Models are typically evaluated using metrics such as accuracy, F1 score, latency, inference throughput, and memory usage, often under constrained environments such as Raspberry Pi, mobile GPUs, or federated client devices. Recent work also increasingly adopts LLM-as-a-judge protocols, where a stronger reference model assigns quality or preference scores to SLM outputs, providing scalable evaluation for open-ended generation and complex reasoning tasks [48,53]. Human-in-the-loop evaluation nevertheless remains the gold standard in safety-critical or domain-intensive settings, where expert annotators assess faithfulness, correctness, and usability.

4.2.5. Training setups and optimization

Several works emphasize transparent and reproducible training setups. OpenELM [41] publishes complete training pipelines, weights, and datasets, allowing community validation and reuse. Phi-4 and Gemma 3 reports provide scaling laws, pretraining regimes, including synthetic corpora, and detailed alignment procedures such as RLHF. Nguyen *et al.* [47] and Subramanian *et al.* [4] analyze various training regimes, including curriculum learning, instruction tuning, and domain adaptation, as strategies to improve generalization in compact models.

4.2.6. Cross-model and domain-specific insights

Several studies offer comparative analyses or case studies showing that performance is not dictated solely by model size, but by the interplay between architectural choices, training paradigms, and deployment context. Lepagnol *et al.* [50] evaluate zero-shot performance and demonstrate that, with appropriate prompt design, small models can approach or match larger counterparts on multi-domain classification. Li *et al.* [49] conduct a real-world deployment case study in productivity tools, measuring interaction

latency and user-perceived quality. Hillier *et al.* [51] push the boundaries of resource constraints by deploying ultra-small models on microcontrollers using aggressive quantization and weight sharing.

Table 5 summarizes the main efficiency-oriented techniques appearing across the landmark studies, indicating where they are used, whether they have been shown to outperform strong baselines, and to what extent they have become "must-have" features in modern SLM design.

Table 5. Cross-model and domain-specific insights on key efficiency techniques: where they are used, when they outperform, and how essential they are in current SLM design.

Technique	Used in	Evidence of Outperformance	Current Role in SLM Design	When It Falls Short
Knowledge distillation and layer-wise alignment	DistilBERT, TinyBERT, MobileBERT, various encoder-based SLMs	Smaller students retain a large fraction (often $>95\%$) of teacher performance on GLUE-style benchmarks while using fewer parameters and lower latency.	De facto standard for compressing encoder models and a common ingredient in task-specific SLMs, especially for classification and QA.	Gains are tightly coupled to teacher quality and domain; generalization beyond the teacher's coverage degrades, and for decoder-only LLMs distillation is less universally adopted.
Architectural bottlenecks and mobile-oriented redesign	MobileBERT, OpenELM, compact variants of Phi-4 and Gemma 3	Latency and memory footprint are significantly reduced on mobile/edge hardware, with accuracy comparable to larger baselines on standard benchmarks.	Widely used in deployment-oriented SLM families targeting phones, laptops, and lightweight servers; increasingly seen as good practice for on-device models.	Designs are architecture- and hardware-specific; bottleneck patterns that work for one family may not transfer cleanly to others without re-optimization.
Pruning and sparsity-aware training	SPP, TranSpa, FedSpaLLM, sparse LLaMA and GPT-style models	High sparsity (often $>80\%$–90%) can be achieved with modest drops in perplexity and downstream accuracy, especially when sparsity is introduced during training.	Promising tool for reducing memory and Floating Point Operations (per second) (FLOPs) when hardware and software stacks support sparse execution; increasingly explored for federated and edge settings.	Benefits depend strongly on hardware support and routing overhead; many production stacks still treat sparsity as experimental, so it has not yet become a universal must-have.
Pretraining native small models	TinyLLaMA, compact Phi-4/Gemma/Qwen variants	Well-trained 1–4B models can match or exceed older large models on several benchmarks and offer better latency/memory trade-offs than compressed legacy models.	Emerging paradigm for next-generation SLMs, especially when training resources are available and deployment constraints are known in advance.	Requires substantial pretraining compute and careful data curation; may not be practical for all organizations compared to fine-tuning or compressing existing LLMs.
Prompting and zero-shot task design	Lepagnol *et al.*, Li *et al.*, many application-focused studies	With carefully engineered prompts, small models can achieve zero-shot accuracy close to larger LLMs on classification and retrieval-style tasks.	Commonly used in practice to extend the reach of SLMs without expensive fine-tuning, particularly in multi-domain and interactive applications.	Highly sensitive to prompt design and domain shift; brittle prompts can cause large performance drops, so prompting alone is not a reliable replacement for good training.
Extreme compression for microcontrollers	Hillier *et al.* (Super Tiny Language Models/μModels)	Enables basic NLP tasks (e.g., intent classification, simple generation) on devices with a few MB of memory and no external accelerator.	Niche but important for ultra-constrained IoT and embedded scenarios; illustrates the outer limits of SLM deployment.	Models are task-specific and far from general-purpose LMs; accuracy and expressiveness are limited, and such techniques are unnecessary for most server or mobile deployments.

4.3. Synthesis and insights

The body of work surveyed reveals a multifaceted landscape of strategies for designing, compressing, and deploying SLMs, each tailored to optimize performance under resource constraints. Several key insights emerge from the comparative analysis of the methodologies and experimental results.

Foundational efforts such as DistilBERT and TinyBERT demonstrate the enduring effectiveness of knowledge distillation in transferring performance from large-scale teacher models to compact students. However, more recent models-such as Phi-4 and Gemma 3, reported to leverage instruction tuning, RLHF, and domain adaptation, indicating that performance retention is increasingly coupled with application-specific optimization rather than general-purpose compression alone.

Approaches like SPP and TranSpa reflect a growing focus on structured sparsity during training and inference, aligning well with hardware-aware optimizations. These methods suggest that sparsity is not merely a post hoc compression technique but is increasingly treated as a design principle. FedSpaLLM, in particular, extends this paradigm to federated and decentralized environments, underscoring the role of adaptive sparsity for on-device learning.

Efforts such as MobileBERT [51], and OpenELM exhibit significant architectural deviations from canonical Transformer designs, often introducing inverted residuals, quantization-aware modules, or modular scaling. These innovations are tightly coupled with deployment constraints, while the work in [51] targets micro-controllers with limited memory budgets, OpenELM scales across open platforms and larger inference contexts. This suggests that the deployment context increasingly shapes architectural decisions and that one-size-fits-all solutions are becoming less relevant.

A noteworthy trend is the broadening of evaluation metrics beyond classical benchmarks such as GLUE or SQuAD. Articles such as [47] and models such Phi-4 incorporate latency, energy consumption, and alignment quality into their evaluation frameworks. Others [50,52], emphasize domain-specific efficacy, including legal reasoning and zero-shot classification. At the same time, the field suffers from evaluation heterogeneity; studies often rely on different benchmark versions, subsets, hardware setups, and reporting formats, making cross-comparison difficult. This highlights the urgent need for standardized multidimensional evaluation protocols that jointly capture accuracy, efficiency, and robustness.

The emergence of openly released models and frameworks—such as OpenELM, Gemma 3, and Qwen2.5 reflect a broader movement toward transparent, reproducible, and democratized research. These projects not only set new performance baselines but also enable downstream researchers and practitioners to adapt SLMs for bespoke use cases. The collaboration between academia, industry, and open-source communities (e.g., Hugging Face, Meta, Google, Microsoft) further accelerates this trend, promoting transparency, reproducibility, and wide adoption.

Despite substantial progress, gaps remain. Few studies rigorously examine robustness under distribution shifts (e.g., across languages or domains), or the long-term stability of SLMs under continual updates and federated learning. In addition, the challenges of fairness, bias mitigation, and safe use are still underexplored. However, real-world deployments demonstrate the tangible utility of SLMs: from medical decision support [55,56] (where privacy and low latency are critical) to financial analysis [57], legal text processing [58], and edge-based productivity assistants [59]. These applications highlight how the

compact nature of SLMs uniquely positions them for environments with constrained compute and strict efficiency requirements.

It is apparent to us that the current wave of SLM research reveals an ecosystem rich with architectural diversity, methodological rigor, and contextual awareness. The future of SLMs appears to hinge not only on minimizing parameters but also on integrating model design with deployment objectives, promoting accessibility, and advancing evaluation standards that better reflect real-world utility.

4.4. Summary

This subsection has reviewed a representative selection of peer-reviewed and preprint research that collectively outlines the evolving design space of SLMs. Across the surveyed works, we observe a transition from traditional compression-based approaches (e.g., distillation, pruning) to more holistic strategies that incorporate architecture rethinking, deployment-specific tuning, and efficiency-aware evaluation.

Pioneering models such as DistilBERT and TinyBERT laid the foundation for efficient knowledge transfer, while more recent contribution—including Phi-4, Gemma 3, and Qwen2.5—highlight the increasing role of instruction tuning, alignment, and real-world applicability. Experimental innovations in sparsity (e.g., TranSpa, FedSpaLLM) and model compression (e.g., Solution Path Pruning, MobileBERT) have been shown to significantly reduce inference costs without severely degrading performance.

Importantly, the reviewed literature reflects a clear trend towards the integration of model design with application constraints, whether these involve memory limits, compute budgets, or latency requirements. Open-source releases and transparent benchmarks further enable broader accessibility and reproducibility, while highlighting the importance of cross-community collaboration.

Nevertheless, challenges remain in terms of evaluation heterogeneity, robustness, and responsible deployment. Future research will need to address these open questions while taking advantage of the practical successes of real-world applications to guide the sustainable development of SLMs.

5. Grey literature material

Grey literature constitutes an important, though often under-examined, source for understanding the rapid evolution of SLMs. It includes technical reports, white papers, software documentation, and corporate research disclosures disseminated through platforms such as arXiv, GitHub, and institutional research blogs. In a field where industrial innovation often moves faster than formal publication cycles, grey literature provides early access to implementation details, engineering trade-offs, and deployment strategies, often preceding peer-reviewed validation.

5.1. Industry-driven technical reports and white-papers

Major technology companies remain primary contributors to authoritative grey literature on SLMs, releasing technical reports that describe architectures, training strategies, optimization methods, and deployment constraints. Meta's LLaMA documentation and the community-developed TinyLLaMA project provide influential reference points for compact transformer design, supervised fine-tuning, and compute-efficient scaling [5,60]. Similarly, Apple's OpenELM emphasizes transparent training pipelines and hardware-aware optimization for Apple Silicon [41,61].

Other major model families further illustrate the diversity of industrial SLM design. Microsoft's Phi-3 highlights curated-data training and favorable latency–accuracy trade-offs for constrained devices, while Google DeepMind's Gemma and Alibaba's Qwen place particular emphasis on open-weight release, multilingual support, quantization readiness, and edge-oriented benchmarking [39,62]. Taken together, these reports provide early visibility into practical engineering choices, although reproducibility and transparency still vary across organizations.

5.2. Open-source contributions and benchmarking platforms

Open-source ecosystems such as Hugging Face, GitHub, and Papers With Code extend the impact of industrial disclosures by offering model checkpoints, training recipes, and evaluation resources. Earlier compact architectures such as DistilBERT, TinyBERT, and ALBERT laid the groundwork for efficient language modeling, while more recent projects such as TinyLLaMA demonstrate that modern SLM development can also be community-driven and relatively transparent.

Benchmarking platforms remain equally important for empirical comparison. Systems such as LMSYS Chatbot Arena and MMLU-Pro support large-scale model assessment, although current benchmarks still under-represent multilingual behavior, domain-specific performance, and hardware-level deployment constraints. In parallel, lightweight deployment frameworks such as Ollama and llama.cpp have made local experimentation considerably more accessible, while also introducing heterogeneity in evaluation settings that complicates direct cross-platform comparison.

6. Prototype system and experimentation

6.1. System overview and implementation

We next present the design and implementation of a lightweight legal assistant focused on the domain of Greek labor law, developed as an applied testbed for evaluating the capacities of SLMs in a low-resource and legally sensitive environment. The system targets two deployment settings—a desktop application equipped with a Retrieval Augmented Generation (RAG) pipeline, and a fully offline mobile version—both built around fine-tuned variants of Gemma 3 (1B/4B) models. The objective is to provide accurate and accessible legal information while ensuring robustness, efficiency, and privacy.

The system architecture was derived from a structured analysis of functional and non-functional requirements. The desktop application integrates a locally fine-tuned Gemma 3 model, a Facebook AI Similarity Search (FAISS)-based retrieval index constructed from legislative texts, curated question/answer pairs and academic notes, and an optional validation mechanism that forwards the outputs to high-end cloud LLMs when the user requests refinement. In contrast, the mobile application is designed for on-device inference using quantized GGUF models without internet connectivity or retrieval components, prioritizing privacy and accessibility. Both platforms share a modular design with well-defined interfaces for preprocessing, inference, retrieval, and user interaction.

A three-part dataset was created to support training and retrieval. Legislative texts were extracted from the "greek_legal_code" dataset and filtered to the labor law domain, producing a corpus of 1284 entries characterized by linguistic heterogeneity and temporal dispersion. A complementary question/answer dataset

was constructed from official FAQ resources of the Ministry of Labor and SEPE, and expanded through manual paraphrasing and controlled GPT-4-based augmentation to approximately 2500 aligned pairs. The augmentation process consisted primarily of paraphrasing, question rewriting, and answer rewriting, with limited generation of new Question Answering (QA) pairs. All augmented examples were manually reviewed for validity and relevance before inclusion. To avoid contamination, augmentation was applied only after the dataset split, and test instances were fully excluded from this process; specifically, test items were not used either as seeds or as bases for paraphrasing. A smaller corpus of academic explanatory notes (around 250–280 entries) was additionally incorporated to support conceptual clarity during retrieval. Together, these datasets provide authoritative foundational information, greater linguistic coverage, and pedagogical interpretability.

Given the practical focus of the implemented prototype, model and tool selection was guided primarily by relevance to resource-constrained deployment and to low-resource language settings, with particular attention to applicability in Greek.

Model selection experiments indicated that Gemma 3 models outperformed comparably sized LLaMA and Mistral variants in Greek fluency and consistency, while also offering high efficiency across desktop, server, and mobile hardware. Preliminary experiments with LoRA-based adaptation were also conducted, but the resulting outputs showed weaker domain alignment and lower practical usefulness in the specialized Greek labour-law setting considered here. For this reason, full fine-tuning was selected for the final experiments, as it yielded more reliable adaptation to the linguistic and terminological demands of the task. Experiments were conducted on an RTX 3050 8GB GPU for the 1B model, and on an A40 server for the 4B model, using a standard training configuration with cosine learning-rate scheduling, warm-up, gradient accumulation, and label smoothing. The evaluation combined perplexity, BLEU-1, BERTScore, F1, and qualitative output inspection.

The RAG pipeline, implemented only in the desktop version, employs multi-source retrieval with priority given to curated QA pairs and academic notes over raw legislation. Dense FAISS embeddings, lightweight keyword heuristics, and a ranking mechanism are used to construct context bundles that the model uses during generation. This design mitigates hallucinations by placing answers in curated high-quality sources. The mobile version instead relies exclusively on quantized local inference, demonstrating that domain-specific legal assistance can run entirely offline on consumer devices.

Overall, the system shows that Small Language Models, when supported by targeted fine-tuning, curated multi-source datasets, retrieval grounding, and hardware-efficient deployment strategies, can function effectively in high-stakes legal contexts. The proposed architecture is extensible to additional legal domains or languages and offers a practical path to AI-assisted, accessible, explainable, and privacy-preserving legal information systems.

6.2. Experimentation

This section presents a preliminary experimental evaluation of our system, focusing on three complementary dimensions: the performance of the fine-tuned models, the impact of quantization and GGUF conversion on mobile deployment, and the effectiveness of the RAG pipeline. The results are analyzed both quantitatively, through standard NLP metrics, and qualitatively, by examining the trade-offs between latency, accuracy, and usability across different deployment scenarios.

6.2.1. Performance evaluation—fine-tuned model evaluation

To assess the performance of the fine-tuned models, we conducted a systematic evaluation on a holdout test set of 400 samples. The evaluation procedure involved generating the model output based on the input prompts and comparing them against the corresponding gold references. Controlled generation settings and a carefully balanced decoding configuration were applied to ensure consistency across experiments. The reported performance values are averaged over five runs with different random seeds.

The evaluation relied on a comprehensive set of automatic metrics that capture different aspects of performance. For surface-level accuracy, we employed token-level F1, Precision, and Recall scores, measuring the exact lexical overlap between generated and reference answers. Beyond this, we adopted widely used text generation metrics such as BLEU-1, ROUGE-L, and METEOR, which reflect fluency, recall of key phrases, and partial matches with linguistic flexibility. To complement these measures with a semantic perspective, BERTScore (F1 variant) was used to capture embedding-level similarity between the model output and the reference text.

To further account for the challenges of evaluating Greek text, the implementation incorporated improved normalization, tokenizer-aligned tokenization, and enhanced computation for BLEU, ROUGE, and METEOR to better reflect content overlap. This ensured that the reported scores are both reliable and sensitive to the linguistic characteristics of the data.

The final evaluation compared four configurations of fine-tuned models: Gemma3-1B trained for 4 and 6 epochs, and Gemma3-4B trained for 5 and 7 epochs. Their performance in various metrics is illustrated in Figure 3.

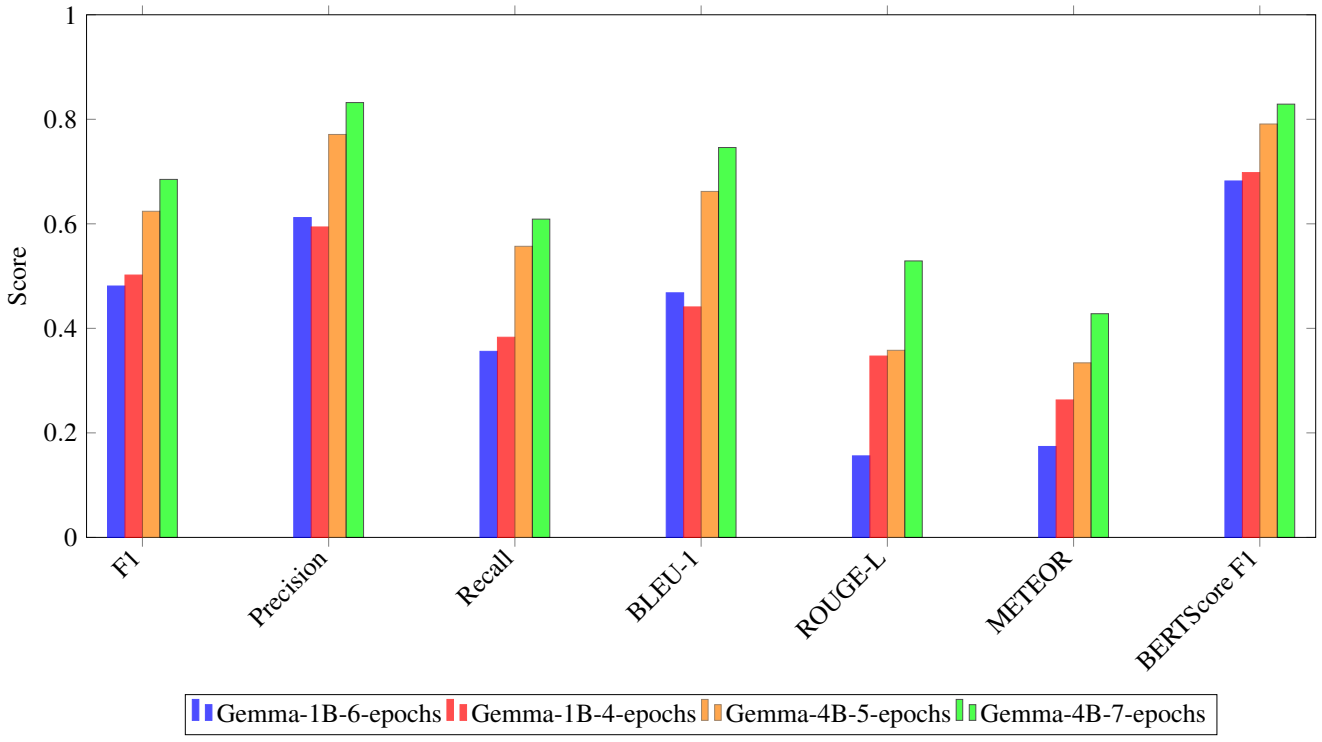

Figure 3. Comparison between fine-tuned models.

6.2.2. Performance evaluation—quantized model evaluation

To complement the evaluation of the fine-tuned models in their standard formats, an additional series of experiments has been conducted on their quantized GGUF variants. The evaluation procedure was identical to the one described above (Section 6.2.1), with the same held-out test set and the same set of metrics, ensuring strict comparability across model configurations.

Higher-precision formats (f16, bf16, q8_0) preserved most of the accuracy while moderately reducing computational burden, while more aggressive schemes (tq1_0, tq2_0, Q4_K_M) introduced noticeable degradation in both lexical overlap and semantic similarity.

The results of these evaluations are summarized in Figure 4, highlighting (i) the robustness of different quantization formats within a single architecture, and (ii) the trade-offs observed when comparing heavily compressed models to smaller models with higher-precision formats. We note that we have selected the Gemma-3-4B model trained for 7 epochs because it achieved the best performance in our evaluations of the fine-tuned models.

To elucidate the trade-offs between heavily Quantized Gemma-4B and smaller high-precision models, we report in Figure 5 results using Gemma-3-1B in both its fp16 and q8 quantization formats, while from the Gemma-3-4B family we retain only the q8 variant.

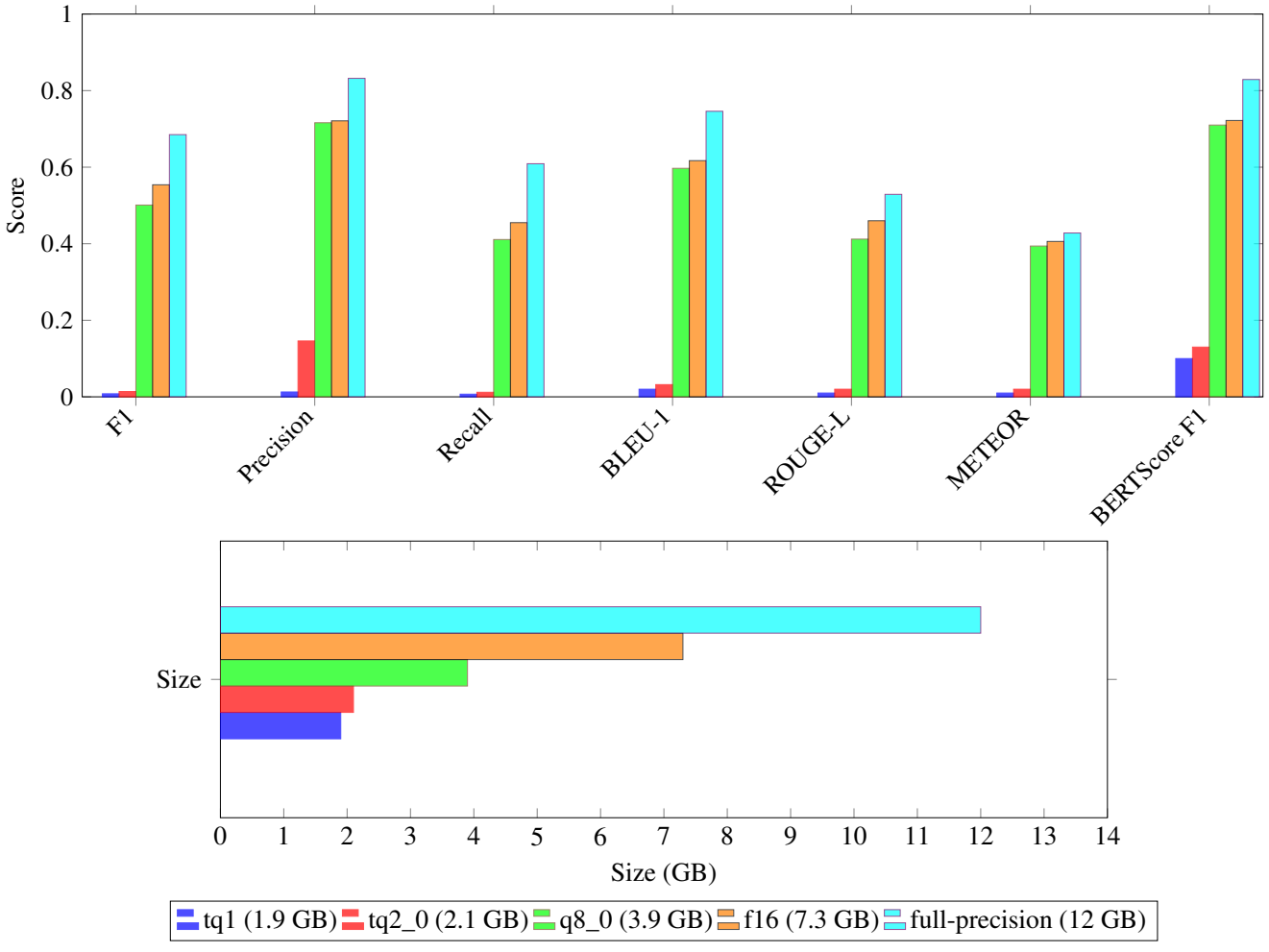

Figure 4. Scores (top) and sizes (bottom) of different quantization formats for Gemma-4B-7-epochs.

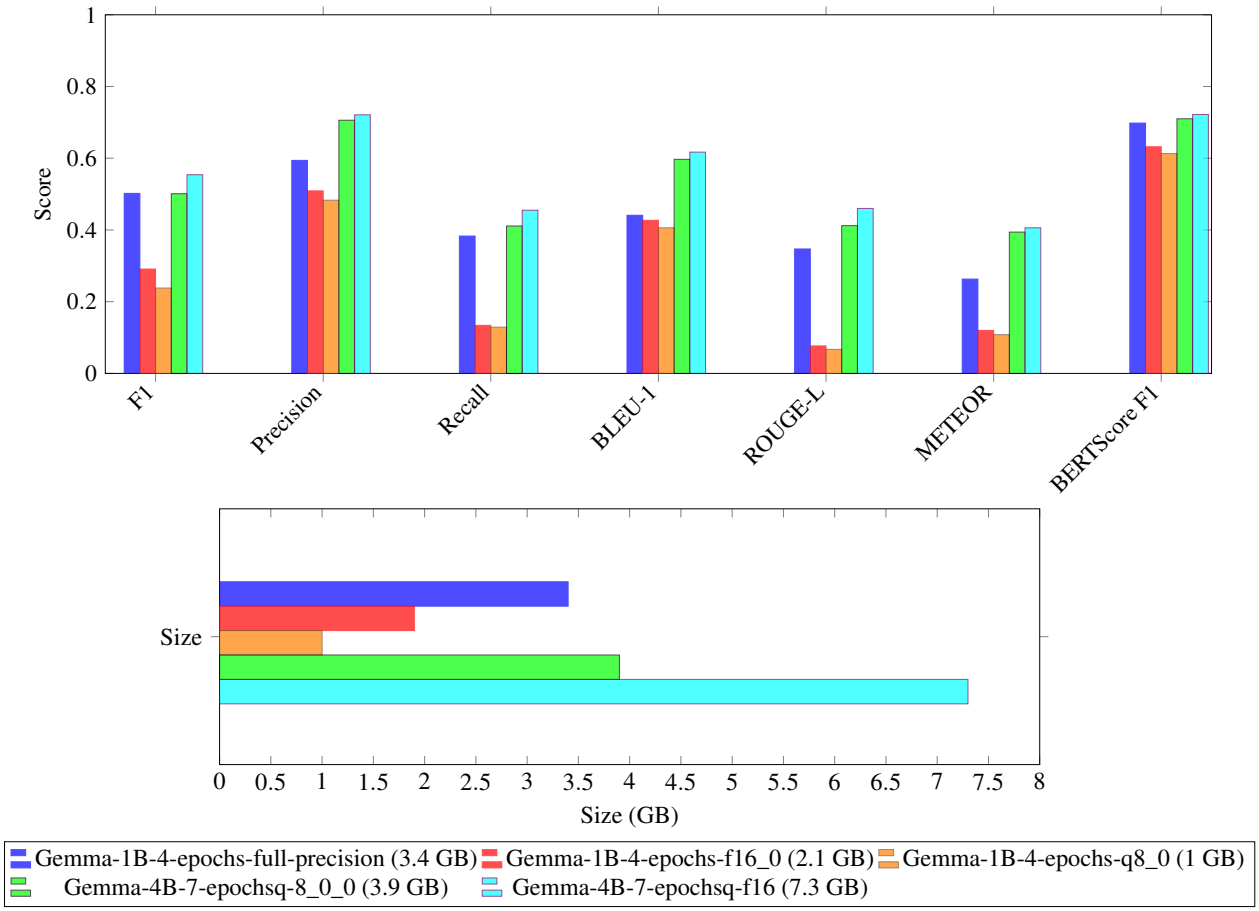

Figure 5. Scores (top) and sizes (bottom) of different quantization formats on Gemma-3-4B-7-epochs.

6.2.3. RAG evaluation

To complement the quantitative metrics presented above, we conducted a set of 50 manual test queries to assess the added value of the RAG pipeline. The queries were carefully selected from realistic Greek labor-law scenarios and evaluated qualitatively by the authors for relevance, factual grounding, and completeness. A formal rubric, inter-annotator agreement, and automated frameworks (e.g., RAG Assessment (RAGAS)) were not included; we instead prioritized in-depth, domain-specific insights into the practical performance of the integrated fine-tuning + RAG pipeline.

The base model, even after fine-tuning, frequently failed to provide complete or factually correct answers, particularly for queries requiring external factual grounding.

When paired with the RAG retriever, the model was able to access relevant context passages, leading to significantly more accurate and grounded answers. In several cases, the RAG-augmented base model achieved performance comparable to (and in some prompts even surpassing) the fine-tuned models, especially when careful prompt engineering was applied.

The strongest results were obtained by combining fine-tuned models with RAG. The fine-tuned models had already internalized task-specific conceptual structures, while the retriever provided additional factual context. This combination produced answers that were both semantically rich and factually grounded, reducing hallucinations and improving robustness across different question types.

6.2.4. Analysis of results

The experimental results highlight several important trends regarding the performance of the fine-tuned models and the impact of quantization on efficiency and accuracy.

Regarding the fine-tuned models, among the four configurations evaluated, the Gemma-3-4B variants consistently outperformed the Gemma-3-1B counterparts across all metrics. The 7-epoch fine-tuned Gemma-3-4B achieved the highest overall performance, reaching an F1 score of 0.685, BLEU-1 of 0.746, and BERTScore F1 of 0.829. This shows that both the scale of the model and the sufficient duration of training contribute to capturing more semantic and lexical nuances of the task.

The comparison between training epochs also revealed a clear pattern: extending training generally led to measurable improvements. For example, Gemma-3-1B at 6 epochs slightly improved precision but failed to surpass the 4-epoch variant in most other metrics, indicating potential overfitting or diminishing returns at smaller model scales. In contrast, Gemma-3-4B benefited strongly from additional training, with the 7-epoch model outperforming the 5-epoch version across the board, particularly in ROUGE-L (+0.171) and METEOR (+0.094). These results suggest that larger models are more robust to longer training regimes and can better leverage extended optimization.

The evaluation of quantized models revealed a sharp contrast between aggressive and conservative quantization strategies. Extremely compressed formats (e.g., tq1_0 and tq2_0) severely degraded the quality of the model, reducing F1 scores close to zero and rendering the models unsuitable for practical use. By contrast, higher-precision schemes, such as q8_0 and f16, preserved much of the full-precision accuracy while substantially reducing memory requirements. For Gemma-3-4B-7 epochs, the q8_0 variant retained an F1 of 0.501 and BERTScore of 0.710 at only 3.9 GB, compared to 12 GB for the full-precision model. This shows a favorable balance between efficiency and quality for deployment scenarios with limited hardware resources.

We next examine the trade-offs between model size and quantization. It is apparent from our experimentation that interesting trade-offs emerged when comparing heavily quantized large models with smaller but high-precision models. For example, Gemma-3-4B-7 epochs in q8_0 format achieved similar or superior performance to Gemma-3-1B full-precision, despite requiring only slightly more memory (3.9 GB *vs.* 3.4 GB). This suggests that moderate quantization of larger architectures can yield better cost–performance ratios than relying on smaller models with full precision.

However, overly aggressive quantization of the smaller Gemma-3-1B (e.g., f16 or q8_0) caused drastic losses in recall and METEOR, highlighting the vulnerability of small models to compression. In practice, this indicates that quantization is most beneficial for larger architectures, where redundancy in parameterization helps absorb precision loss without catastrophic degradation.

At the same time, the Gemma-3-1B model demonstrates surprisingly competitive performance given its size. Its fp16 variant, in particular, offers a favorable compromise between memory efficiency and accuracy, occupying less than 2 GB while still maintaining acceptable results. This makes it especially attractive for deployment on resource-constrained devices, such as smartphones, where storage and compute capacity are limited.

An additional finding from our manual RAG evaluation concerns the comparison between RAG-only and RAG+fine-tuned configurations. In the RAG-only setting, retrieval improves factual grounding by supplying relevant context passages, but the pretrained model does not always make effective use of this information, often producing incomplete or partially misaligned answers. By contrast, the RAG+fine-tuned setup benefits from domain-adaptive training on Greek labor-law data, which helps the model better capture both the linguistic characteristics of Greek as a low-resource language and the specialized legal terminology of the domain. As a result, the retrieved evidence becomes more usable during generation, leading to answers that are more accurate, better structured, and less prone to hallucination. This suggests that, in low-resource and domain-specific settings, retrieval is most effective when combined with a language- and domain-adapted SLM.

As a sanity check beyond SLM-only comparisons, we also carried out a small-scale qualitative experiment in which we posed representative Greek labour-law queries to both our best SLM configuration (fine-tuned Gemma-3-4B with RAG) and GPT-4 with web access, and manually assessed the answers. In these trials, GPT-4 occasionally produced longer, more stylistically polished responses and sometimes exploited up-to-date online sources, but for the narrowly scoped questions in our dataset the SLM+RAG setup frequently produced answers that we judged as comparably useful in terms of factual adequacy and practical guidance. In parallel, we probed other high-capacity LLMs without internet access, including LLaMA-family models, Gemini, and DeepSeek. Although these systems generally exhibited stronger raw fluency in Greek, they often hallucinated legal facts or misinterpreted key provisions, underscoring the difficulty of handling a low-resource language and a highly specialized legal domain without explicit domain grounding. Overall, we attribute the relative robustness of our SLM pipeline to the strong domain conditioning provided by the curated Greek labour-law corpus and the RAG component, which compensates for the lack of web access and yields more reliable answers than generic, non-grounded LLMs in this setting. In addition, GPT-4 incurs higher and more variable end-to-end latency due to remote inference and per-token API billing, whereas our SLM-based assistant responds locally on commodity hardware with predictable latency and no usage-based cost, which is particularly important for repeated interactive use in practice.

Overall, the results support three main conclusions that highlight the importance of balancing the size of the model, the training regime and the quantization strategy to achieve optimal performance under real-world constraints.

(1) The model scale is the strongest quality factor and Gemma-3-4B consistently outperforms the 1B variant.

(2) The duration of training positively impacts larger models, while smaller models plateau or degrade more quickly.

(3) Quantization offers a viable path for efficient deployment, but only when applied conservatively and primarily to larger models, which can retain accuracy while dramatically reducing memory footprint.

7. Summary and conclusions

This paper investigated the challenges and opportunities of SLMs, analyzing their theoretical underpinnings, surveying the state of the art, and implementing a complete prototype that combined fine-tuning, RAG, quantization, and deployment across both desktop and mobile environments. The overarching question was whether SLMs can offer a realistic, sustainable and reliable alternative to LLMs in practical contexts characterized by limited resources. In summary, our study led to a series of significant conclusions that highlight the unique value of SLMs:

One important outcome is that SLMs significantly lower the barrier of entry into the world of generative AI. Much of the experimentation in this paper was conducted on a PC equipped with an entry-level GPU. Despite these modest resources, it was possible to fine-tune, quantize, and deploy models that delivered competitive results. This shows that researchers, small organizations, and even individual enthusiasts can meaningfully engage with cutting-edge AI without relying on expensive cloud infrastructure. The lower cost per token compared to commercial LLM APIs further amplifies this accessibility, reinforcing the role of SLMs as an accessible alternative in AI.

The combination of fine-tuned models with RAG proved to be especially powerful. Fine-tuning allowed the model to capture the conceptual and domain-specific nuances of the dataset, while RAG extended its knowledge horizon by allowing it to access additional contextual information during inference. Together, these techniques yielded responses that were not only more accurate, but also more stable and interpretable. With high-quality data, such an approach can approach, and in some domain-specific settings rival, much larger models when the task is narrowly scoped and well defined [63]. Informal comparisons against GPT-4 and other large models further suggested that, for well-defined Greek labour-law questions, a carefully fine-tuned SLM with RAG can deliver practically comparable answers while offering markedly better latency, cost, and privacy characteristics.

Through quantization and careful model selection, SLMs achieved performance levels that covered most practical needs while dramatically reducing memory requirements. The models were able to respond reliably even when deployed on devices with limited Random Access Memory (RAM). This finding illustrates that efficiency and usability can coexist, provided that the right trade-offs are applied.

The successful deployment of quantized SLMs on mobile devices was one of the most practically relevant results. Despite the challenges posed by limited memory, the models responded with satisfactory accuracy and latency. Equally important was how seamlessly the mobile application loaded and functioned, which underscored the feasibility of on-device AI assistants. Given the growing reliance on smartphones, mobile SLMs constitute a highly promising direction for both consumer and enterprise use cases, offering offline availability, personalization, and independence from cloud services.

In a world where personal data are increasingly at risk, the ability to run SLMs locally—whether on desktops, on edge servers, or smartphones—provides a powerful advantage. By avoiding the need to transmit sensitive queries to remote servers, organizations can protect user privacy while still leveraging advanced AI capabilities. This feature will likely become a decisive factor in the adoption of SLMs in domains such as law, healthcare, and education.

The overall pipeline, which covers fine-tuning, quantization, RAG integration, and deployment, did not reveal major architectural weaknesses. In contrast, it was modular, extensible, and robust in different scenarios. This robustness suggests that similar architectures could be easily adapted to other domains with minimal redesign.

Despite the encouraging results, our study faced certain limitations:

Dataset Constraints: The training corpus was limited in size and scope. Larger, more diverse, and domain-balanced datasets would likely enhance both the accuracy and generalizability of the models.

Hardware Restrictions: Our experiments have been constrained by the available hardware, both in terms of GPU memory and mobile device RAM. More advanced equipment would enable deeper exploration of hyperparameter tuning and larger-scale training.

Evaluation Scope: Although multiple automated metrics were used, a broader evaluation that included expert human feedback—especially in legal reasoning—would provide more nuanced insights.

Building on our study, the following promising avenues for future research and development emerge.

Enhanced Mobile Deployment: The successful PocketPal prototype showed that quantized models (e.g., q8_0) can be loaded and executed directly on smartphones, even under strict RAM constraints. Although performance was below that of full-precision models, the application remained responsive and practically useful. Future work could explore more advanced compression methods and mobile-aware optimization to further improve latency and accuracy.

Server-Based Multi-User Systems: Beyond local deployment, an equally interesting direction is the integration of SLMs into a centralized server environment. Hosting the model on a dedicated server would allow multiple users within an organization to connect simultaneously, benefiting from shared access to an optimized AI system while maintaining lower costs than LLM-based services.

Improved Data Quality and Scale: One of the limiting factors in our study was the availability of curated data. With a higher budget and access to larger, more balanced corpora, especially for domain-specific tasks, model fine-tuning could be pushed much further. Better data would likely yield a more robust conceptual capture, strengthening the model's ability to compete with much larger systems.

Cross-Domain and Multilingual Expansion: Extending SLMs to cover multiple languages and domains beyond labor law (e.g., healthcare, finance, education) would broaden their utility and test their adaptability. Multilingual SLMs could be transformative for underrepresented communities.

Cost and Accessibility: The work demonstrated that meaningful research and deployment can be achieved on consumer-grade hardware. This accessibility is an important factor for democratizing AI. Future studies could quantify the cost-per-token more systematically and explore open-source toolchains that reduce entry barriers even further.

Integration of Advanced Compression Techniques: Exploring structured pruning, knowledge distillation, and on-the-fly adaptive inference could further improve efficiency. Combining these methods with quantization may lead to models that are even smaller, faster, and more capable.

Privacy-Oriented AI Applications: Since SLMs can run locally, they are naturally aligned with privacy-sensitive use cases. Future work should emphasize developing end-to-end pipelines where sensitive data never leave the device, making such solutions attractive for legal, healthcare, or educational applications.

Domain-Specific RAG Systems: Since the synergy of fine-tuning and RAG proved to be so successful, future efforts should explore more sophisticated retrieval pipelines, curated corpora, and adaptive indexing strategies. This would allow small models to compete head-to-head with much larger systems in specific, high-value tasks.

In conclusion, this paper demonstrated that Small Language Models are far more than trimmed-down versions of Large Language Models. They represent a distinct and valuable paradigm in AI—one that emphasizes efficiency, accessibility, privacy, and adaptability. Their ability to perform reliably on low-cost hardware, deliver competitive results in domain-specific contexts, and function seamlessly on mobile devices makes them a compelling alternative to resource-intensive LLMs.

The findings suggest that the future of AI will not be defined solely by ever-larger models, but also by smaller, smarter, and more accessible ones. With continued innovation in training techniques, deployment strategies, and data curation, SLMs are poised to become a cornerstone of practical, sustainable, and democratized AI in the years ahead.

Data availability statement

The complete prototype implementation—including fine-tuning scripts, RAG pipeline, and quantization experiments—is available at https://github.com/koxrhstos/slm-greek-labor-law-chatbot.git together with installation instructions, usage examples, and pretrained checkpoints where applicable.

Declaration of generative AI and AI-assisted technologies

During the preparation of this manuscript, the authors used ChatGPT only to improve language and readability. The authors take full responsibility for the content of the manuscript.

Acknowledgments

The research of M. Vavalis has been co-financed by the European Union and Greek national funds through the Program COMPETITIVENESS under the call RESEARCH–INNOVATE (project code: ΕΚΠΑΡ01-0069568).

Authors' contribution

Christos Kotrotsios: methodology, writing—original draft, software and writing—review and editing; Manolis Vavalis: conceptualization, methodology, funding acquisition and writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

Manolis Vavalis holds the position of Associate Editor for *AI+* and has not peer reviewed or made any editorial decisions for this paper.

References

[1] Takyar A. Small language models explained: use cases, applications, advantages, technologies, implementation and development. 2024. Available: https://www.leewayhertz.com/small-language-models/ (accessed on 16 April 2025).

[2] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv* 2020, arXiv:1910.01108.

[3] Wang F, Zhang Z, Zhang X, Wu Z, Mo T, *et al.* A comprehensive survey of small language models in the era of large language models: techniques, enhancements, applications, collaboration with LLMs, and trustworthiness. *arXiv* 2024, arXiv:2411.03350.

[4] Subramanian S, Elango V, Gungor M. Small language models (SLMs) can still pack a punch: a survey. *arXiv* 2025, arXiv:2501.05465.

[5] Zhang P, Zeng G, Wang T, Lu W. TinyLlama: an open-source small language model. *arXiv* 2024, arXiv:2401.02385.

[6] Wang J, Zeng Y, Guo J, Ma Y, Liu A, *et al.* SLMQuant: benchmarking small language model quantization for practical deployment. *arXiv* 2025, arXiv 2511.13023.

[7] Belcak P, Heinrich G, Diao S, Fu Y, Dong X, *et al.* Small language models are the future of agentic AI. *arXiv* 2025, arXiv:2506.02153.

[8] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, *et al.* Attention is all you need. *arXiv* 2023, arXiv:1706.03762.

[9] Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, *et al.* Language models are few-shot learners. *arXiv* 2020, arXiv:2005.14165.

[10] Devlin J, Chang M, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv* 2019, arXiv:1810.04805.

[11] Ding L, Lawson C, Shapira P. Rise of generative artificial intelligence in science. *Scientometrics* 2025, 130(9):5093–5114.

[12] Abbasi MA. Foundations of generative AI: GANs vs diffusers vs transformer models. 2024. Available: https://github.com/MuhammadAhsaanAbbasi/generative-ai/tree/main/00_foundations/00_genai_foundations (accessed on 21 April 2026).

[13] Golovneva O, Wang T, Weston J, Sukhbaatar S. Contextual position encoding: learning to count what's important. *arXiv* 2024, arXiv:2405.18719.

[14] Håkansson A, Phillips-Wren G. Generative AI and large language models benefits, drawbacks, future and recommendations. *Procedia Comput. Sci.* 2024, 246:5458–5468.

[15] de Vries A. The growing energy footprint of artificial intelligence. *Joule* 2023, 7(10):2191–2194.

[16] Sun Y, Gai Y, Chen L, Ravichander A, Choi Y, *et al.* Why and how LLMs hallucinate: connecting the dots with subsequence associations. *arXiv* 2025, arXiv:2504.12691.

[17] Cuskley C, Woods R, Flaherty M. The limitations of large language models for understanding human language and cognition. *Open Mind* 2024, 8:1058–1083.

[18] Yao Y, Duan J, Xu K, Cai Y, Sun Z, *et al.* A survey on large language model (LLM) security and privacy: the good, the bad, and the ugly. *High-Confid. Comput.* 2024, 4(2):100211.

[19] Zhang J, Bu H, Wen H, Liu Y, Fei H, *et al.* When LLMs meet cybersecurity: a systematic literature review. *Cybersecurity* 2025, 8:55.

[20] Abdali S, Anarfi R, Barberan CJ, He J. Securing large language models: threats, vulnerabilities and responsible practices. *arXiv* 2024, arXiv:2403.12503.

[21] Bronsdon C. What is the cost of training LLM models? A comprehensive guide for AI professionals. 2025. Available: https://www.galileo.ai/blog/llm-model-training-cost (accessed on 21 April 2026).

[22] Mojthdi B. Revolutionizing IoT: the rise of small language models (SLMs) in edge computing. 2024. Available: https://www.virtvps.com/small-language-models-revolutionizing-iot-the-r/ (accessed on 1 May 2025).

[23] Gülen K. Small language models (SLMs). 2025. Available: https://dataconomy.com/2025/03/10/what-are-small-language-models-slms-2/ (accessed on 21 April 2026).

[24] Sun Z, Yu H, Song X, Liu R, Yang Y, *et al.* MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, June 5–10, 2020, pp. 2158–2170.

[25] Tay Y, Dehghani M, Bahri D, Metzler D. Efficient transformers: a survey. *ACM Comput. Surv.* 2022, 55(6):1–28.

[26] Ionio. LLMs on CPU: the power of quantization with GGUF, AWQ, & GPTQ. 2025. Available: https://www.ionio.ai/blog/llms-on-cpu-the-power-of-quantization-with-gguf-awq-gptq (accessed on 6 May 2025).

[27] Michel P, Levy O, Neubig G. Are sixteen heads really better than one? *arXiv* 2019, arXiv:1905.10650.

[28] Liu Y, Ning J, Xia S, Gao X, Qiang N, *et al.* Pruning large language models by identifying and preserving functional networks. *arXiv* 2025, arXiv:2508.05239.

[29] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. *arXiv* 2015, arXiv:1503.02531.

[30] Matsuyama K, Anjum U, Matsuyama S, Shoda T, Zhan J. Adaptive temperature based on logits correlation in knowledge distillation. *arXiv* 2025, arXiv:2503.09030.

[31] Sun S, Ren W, Li J, Wang R, Cao X. Logit standardization in knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA, June 19–21, 2024, pp. 15731–15740.

[32] Mansourian AM, Ahmadi R, Ghafouri M, Babaei AM, Badali Golezani E, *et al.* A comprehensive survey on knowledge distillation. *arXiv* 2025, arXiv:2503.12067.

[33] Fu Y, Yu Y, Han X, Li R, Long X, *et al.* Dynamic self-distillation via previous mini-batches for fine-tuning small language models. *arXiv* 2024, arXiv:2411.16991.

[34] Guha A, Kashiramka S, Krishnan R. Survey of small language models. *Int. J. Eng. Res. Technol* 2024, 13(10).

[35] Jiao X, Yin Y, Shang L, Jiang X, Chen X, *et al.* TinyBERT: distilling BERT for natural language understanding. *arXiv* 2020, arXiv:1909.10351.

[36] Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, *et al.* ALBERT: a lite BERT for self-supervised learning of language representations. *arXiv* 2019, arXiv:1909.11942.

[37] Abdin MI, Aneja J, Behl HS, Bubeck S, Eldan R, *et al.* Phi-4 technical report. *arXiv* 2024, arXiv:2412.08905.

[38] Abouelenin A, Ashfaq A, Atkinson A, Awadalla H, Bach N, *et al.* Phi-4-mini technical report: compact yet powerful multimodal language models via mixture-of-LoRAs. *arXiv* 2025, arXiv:2503.01743.

[39] Team G, Kamath A, Ferret J, Pathak S, Vieillard N, *et al.* Gemma 3 technical report. *arXiv* 2025, arXiv:2503.19786.

[40] Yang A, Yang B, Zhang B, Hui B, Zheng B, *et al.* Qwen2.5 technical report. *arXiv* 2024, arXiv:2412.15115.

[41] Mehta S, Sekhavat MH, Cao Q, Horton M, Jin Y, *et al.* OpenELM: an efficient language model family with open training and inference framework. *arXiv* 2024, arXiv:2404.14619.

[42] Becking D, Friese I, Müller K, Buchholz T, Galkow-Schneider M, *et al.* Efficient federated learning tiny language models for mobile network feature prediction. *arXiv* 2025, arXiv:2504.01947.

[43] Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, *et al.* Transformers: state-of-the-art natural language processing. In *Proceedings of Findings of the Association for Computational Linguistics: System Demonstrations, EMNLP 2020*, Online, November 16–20, 2020, pp. 38–45.

[44] Ding Y, Fan K, Wang Y, Sun X, Fu Y. Adaptive pruning of pretrained transformer via differential inclusions. *arXiv* 2025, arXiv:2501.03289.

[45] Xiao J, Yin M, Yang C, Sui Y, Phan H, *et al.* TranSpa: towards efficient structured sparse training for transformers. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Singapore, April 24–28, 2025.

[46] Bai G, Li Y, Li Z, Zhao L, Kim K. FedSpaLLM: federated pruning of large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Albuquerque, USA, 2025, pp. 8361–8373.

[47] Nguyen CV, Shen X, Aponte R, Xia Y, Basu S, *et al.* A survey of small language models. In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing—Natural Language Processing in the Generative AI Era*, Varna, Bulgaria, September 8–10, 2025, pp. 807–821.

[48] Lu Z, Li X, Cai D, Yi R, Liu F, *et al.* Small language models: survey, measurements, and insights. *arXiv* 2024, arXiv:2409.15790.

[49] Li B, Zhang Y, Bubeck S, Pathuri J, Menache I. Small language models for application interactions: a case study. *arXiv* 2024, arXiv:2405.20347.

[50] Lepagnol P, Gerald T, Ghannay S, Servan C, Rosset S. Small language models are good too: an empirical study of zero-shot classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Torino, Italia, May 20–25, 2024, pp. 14923–14936.

[51] Hillier D, Guertler L, Tan C, Agrawal P, Ruirui C, *et al.* Super tiny language models. *arXiv* 2024, arXiv:2405.14159.

[52] Martin L, Whitehouse N, Yiu S, Catterson L, Perera R. Better call GPT, comparing large language models against lawyers. *arXiv* 2024, arXiv:2401.16212.

[53] Srivastava G, Cao S, Wang X. Towards reasoning ability of small language models. *arXiv* 2025, arXiv:2502.11569.

[54] Iozzia G. *Domain-Specific Small Language Models*, 1st ed. Shelter Island: Manning Publications, 2026.

[55] Garg M, Raza S, Rayana S, Liu X, Sohn S. The rise of small language models in healthcare: a comprehensive survey. *arXiv* 2025, arXiv:2504.17119.

[56] Kim H, Hwang H, Lee J, Park S, Kim D, *et al.* Small language models learn enhanced reasoning skills from medical textbooks. *npj Digit. Med.* 2025, 8:240.

[57] Infosys. The case for small language models in financial services. 2025. Available: https://www.infosys.com/iki/perspectives/small-language-models-financial-services.html (accessed on 21 April 2026).

[58] UBIAI. SLMs *vs.* LLMs: a definitive guide to small & large language models in 2025. 2025. Available: https://ubiai.tools/slms-vs-llms-a-definitive-guide-to-small-large-language-models-in-2025 (accessed on 21 April 2026).

[59] AI P. Small language models and edge AI: a concise guide for 2025. 2025. Available: https://peaklightai.medium.com/small-language-models-and-edge-ai-a-concise-guide-for-2025-63936989a0d3 (accessed on 21 April 2026).

[60] Touvron H, Martin L, Stone KRW, Albert P, Almahairi A, *et al.* LLaMA 2: open foundation and fine-tuned chat models. *arXiv* 2023, arXiv:2307.09288.

[61] Touvron H, Martin L, Stone K, Albert P, Almahairi A, *et al.* Apple intelligence foundation language models. *arXiv* 2023, arXiv:2307.09288.

[62] Yang A, Li A, Yang B, Zhang B, Hui B, *et al.* Qwen3 technical report. *arXiv* 2025, arXiv:2505.09388.

[63] Maroudas S, Legkas S, Malakasiotis P, Chalkidis I. Legal-tech open diaries: lesson learned on how to develop and deploy light-weight models in the era of humongous language m0odels. *arXiv* 2022, arXiv:2210.13086.