MIM Research Group Research & Design
www.jrede.org

Review Article

A comparative review of hallucination mitigation and performance improvement techniques in Small Language Models

Fatih Ahmet Senel[1a], Hayri Baytan Ozmen*[1b]

[1]*Faculty of Engineering and Natural Sciences, Süleyman Demirel University, Isparta, Türkiye.*
[2]*Faculty of Engineering and Natural Sciences, Uşak University, Uşak, Türkiye.*

Article Info

Article History:

Received:
23 May 2025
Accepted:
15 June 2025

Keywords:

Knowledge Distillation;

Natural Language Processing (NLP);

Parameter-Efficient Fine-Tuning;

Performance optimization;

Retrieval-Augmented Generation (RAG);

Text generation

Abstract

Small Language Models (SLMs) offer a computationally efficient alternative to Large Language Models (LLMs), enabling natural language processing (NLP) capabilities in resource-constrained and/or private environments such as personal computers, mobile devices, embedded systems, and real-time applications. However, SLMs face significant challenges related to factual hallucination, limited generalization, and degraded task performance due to their reduced parameter capacity. This review provides a comparative analysis of current methods to mitigate hallucination and enhance performance in SLMs. We consider hallucination prevention techniques into five primary strategies: retrieval-augmented generation (RAG), instruction tuning and prompt engineering, fact-checking and verification layers, calibration mechanisms, and fine-tuning with human feedback. In parallel, we explore performance enhancement methods including quantization, pruning, parameter-efficient tuning, knowledge distillation, mixture-of-experts architectures, and domain-adaptive training. A comparative evaluation highlights trade-offs between accuracy, compute efficiency, and deployment feasibility. We identify best-fit combinations of techniques for diverse real-world scenarios—ranging from mobile applications to safety-critical systems—and discuss integration challenges, sustainability concerns, and ethical implications. This work is aimed to introduce and compare methods for developing robust and efficient SLMs capable of reliable deployment across varied NLP contexts.

1. Introduction

Natural Language Processing (NLP) is a subfield of artificial intelligence concerned with the interaction between computers and human (natural) languages. Its development has spanned several decades, from rule-based symbolic approaches in the 1950s and 1960s to statistical models in the 1990s and, more recently, neural models that dominate the field today [1, 2]. Early systems relied heavily on handcrafted grammars and lexicons, which limited their scalability and adaptability. With the advent of machine learning, particularly supervised learning, NLP began leveraging large annotated corpora to learn statistical patterns for tasks such as part-of-speech tagging, named entity recognition, and machine translation.

The transition to deep learning accompanied in the era of neural language models. Word embeddings such as Word2Vec [3] and GloVe [4] provided dense vector representations of

*Corresponding author: baytan.ozmen@usak.edu.tr
[a]orcid.org/0000-0003-1918-7277; [b]orcid.org/0000-0001-6750-8632;
DOI: http://dx.doi.org/10.17515/rede2025-004en0523rs
Res. Des. Vol. 2 Iss. 1 (2025) 45-65

words, capturing semantic similarities. These were later surpassed by contextual embeddings produced by deep neural networks such as ELMo [5], BERT [6], and GPT [7, 8]. These models, especially the Transformer-based architectures, revolutionized NLP by enabling pretraining on massive text corpora followed by fine-tuning on specific downstream tasks.

Large Language Models (LLMs) such as GPT-3 [9], PaLM [10] and GPT-4 [11] possess billions of parameters and exhibit emergent capabilities such as few-shot learning, reasoning, and dialogue generation. These models have become foundational in a variety of applications, including virtual assistants, summarization tools, and generative text engines. However, LLMs also face significant challenges. Main among them are computational inefficiency, high energy costs, and difficulties in deployment on edge devices [12]. Additionally, LLMs are known to hallucinate—i.e., generate fluent but factually incorrect or misleading outputs—which poses serious risks in real-world applications [13].

In response to the operational challenges posed by LLMs, the NLP community has seen renewed interest in Small Language Models (SLMs)—models with significantly fewer parameters that are tailored for efficient, cost-effective deployment. SLMs are often used in scenarios requiring low-latency inference, privacy preservation, or deployment on devices with limited computational resources such as personal computers, mobile phones or embedded systems [14]. Although smaller in size, SLMs can still achieve competitive performance on specific tasks when equipped with architectural and training optimizations [15].

However, SLMs are more prone to issues such as hallucination, reduced reasoning capacity, and brittleness across domains due to their limited parameter space. Addressing these issues is critical for expanding their utility and reliability in real-world applications.

This review aims to provide a comprehensive overview of existing methods to prevent hallucination and enhance the performance of SLMs in NLP. Specifically, we:

- Introduce and explain key hallucination mitigation techniques such as retrieval-augmented generation (RAG), instruction tuning, and post-hoc verification;
- Examine performance enhancement strategies including parameter-efficient tuning (e.g., LoRA), quantization, pruning, and knowledge distillation;
- Offer a detailed comparative analysis of these techniques based on their efficacy, computational cost, and suitability for different use cases;
- Discuss when and how combinations of methods may be employed synergistically;
- Address the question of whether the performance gains justify the additional resource overhead, particularly in low-resource environments.

By focusing on SLMs, this review tries to fills a gap in the literature where most reviews tend to prioritize LLMs. We aim to equip researchers and practitioners with the insights needed to select and implement optimal strategies for building accurate, efficient, and responsible small language models.

2. Understanding Hallucination in Language Models

In the context of natural language generation (NLG), hallucination refers to the generation of output that is linguistically plausible but factually inaccurate, unverifiable, or misleading. Although hallucination affects models of all scales, its manifestations and impacts are often amplified in Small Language Models (SLMs) due to their reduced capacity and generalization ability [13].

The taxonomy of hallucination can be divided broadly into two categories [16, 17]:

- Intrinsic Hallucinations: These are errors introduced due to model misinterpretation or lack of understanding of the input context. The generated content contradicts or deviates from the input or ground truth. For example, in abstractive summarization, an intrinsic hallucination might involve the model fabricating an event that is not present in the source document.
- Extrinsic Hallucinations: In these cases, the model generates information that is not explicitly supported or contradicted by the input. These statements may seem plausible but cannot be grounded in the provided data. This form of hallucination is particularly common when the model is prompted with underspecified queries or lacks external factual support.

More recent studies have extended this classification to include fabricated references, nonsensical reasoning, and domain transfer hallucinations, especially in complex tasks such as dialogue generation, open-domain question answering, and summarization of long-form documents [18].

2.1 Causes of Hallucination in SLMs

While hallucination is a known issue even in large models like GPT-3 and PaLM, it is exacerbated in SLMs due to their constrained architecture, limited training data, and restricted contextual awareness[13]. Below are key contributing factors to hallucinations in SLMs:

2.1.1 Limited model capacity

SLMs typically operate with tens to hundreds of millions of parameters, compared to the tens of billions in LLMs. This limitation reduces their ability to represent complex semantic relationships, long-range dependencies, and nuanced contextual signals [14]. As a result, SLMs often generalize poorly, especially in tasks requiring factual precision or cross-sentence coherence.

2.1.2 Shallow contextualization and shorter attention windows

Many SLMs are trained or fine-tuned with limited input context (e.g., 512 tokens or fewer), which constrains their ability to understand extended discourse or refer to multiple evidential sources. In multi-turn conversations or document-level summarization, this results in hallucinations due to the loss of grounding signals across segments [19].

2.1.3 Training data biases and incompleteness

Hallucinations often stem from gaps in training corpora or exposure to unreliable text [20]. Since SLMs are trained on a subset of what LLMs typically ingest, they may lack exposure to edge cases, rare facts, or formal domain-specific knowledge, increasing the risk of error in critical use cases such as biomedical NLP or legal summarization.

2.1.4 Over-reliance on statistical priors

SLMs, like their larger counterparts, optimize the next-token prediction probability, leading them to generate text that is statistically probable rather than factually correct. This is particularly problematic when the model is asked for specific, non-frequent information that diverges from corpus norms, such as citing an uncommon journal article or describing niche technical concepts [21].

2.1.5 Lack of external grounding

Most SLMs do not include mechanisms for retrieval-augmented generation (RAG) or fact-checking modules that can verify their outputs against real-time knowledge sources. Without such grounding, these models are forced to rely solely on internal representations, increasing the likelihood of hallucinating when asked for factual information or summarizing ambiguous input [22].

2.1.6 Catastrophic forgetting in fine-tuning

When SLMs are fine-tuned for downstream tasks without care, they may "forget" previously learned knowledge due to the phenomenon of catastrophic forgetting [23]. This often results in hallucinations, especially when fine-tuning for domains with conflicting or newly introduced knowledge representations.

2.2 Impacts of Hallucination on Downstream Applications

The presence of hallucinations in SLM-generated content poses significant technical, ethical, and operational challenges, especially in applications where trust and accuracy are non-negotiable.

In fields like medicine and law, where AI tools are increasingly deployed to generate summaries, case reports, or recommendations, hallucinated content can result in diagnostic errors, misinterpretation of legal precedents, and regulatory non-compliance [24, 25]. SLMs deployed in low-resource clinical settings, if not robustly trained and evaluated, can propagate dangerous misinformation.

When used in education, hallucinations can mislead learners with inaccurate explanations or fabricated references, undermining trust in automated tutoring systems or language learning assistants. These errors are often subtle and hard to detect, especially in SLMs trained on general-purpose data.

For chatbots and digital assistants, hallucinations degrade user experience and erode trust. Users may take plausible-sounding but false statements at face value, particularly when the hallucinations are delivered fluently and confidently. In domains such as finance or insurance, this can have legal and reputational consequences [19].

Hallucinations contribute to content pollution when SLMs are used to generate or amplify content at scale, especially in social media or low-cost content farms. The ability of even small models to produce vast volumes of text means that low-quality or fabricated content can easily outpace human fact-checking and moderation efforts.

2.3 Hallucination Severity in Low-Resource vs. High-Resource Settings

Hallucinations manifest with greater severity and frequency in low-resource environments, which include both data-scarce tasks and deployment scenarios with limited compute, memory, or connectivity. In such cases, the model's ability to access external grounding sources or undergo thorough fine-tuning is severely constrained. For example, in healthcare delivery in rural areas, SLMs may be deployed on mobile devices with minimal local data and no internet access. Without retrieval augmentation or updated knowledge bases, these models rely entirely on internal parameters trained on outdated or general-purpose corpora, thereby increasing their tendency to hallucinate [19, 26].

In contrast, high-resource settings may integrate SLMs with supporting infrastructure like retrievers, validators, or human-in-the-loop supervision, allowing for multi-stage generation pipelines that mitigate hallucination. However, even in these settings, when computational priorities shift toward real-time response or user privacy, grounding mechanisms may be disabled, again increasing hallucination risk. This contrast suggests

that deployment context significantly shapes the impact of hallucinations, and that a one-size-fits-all solution is unlikely to suffice. Fine-grained trade-off analyses between computational cost, hallucination risk, and application criticality are necessary.

2.4 Implications for Trust, Safety, and Model Evaluation

As hallucinations become a central concern in language model deployment, especially in SLMs intended for real-world use, the following implications must be considered:

Even when hallucinations are rare, the inability to predict when a hallucination will occur erodes user trust. Trust calibration requires that users can distinguish between high- and low-confidence model responses, ideally through probabilistic output scores, uncertainty estimations, or fact-checking signals [27]. Unfortunately, most SLMs do not natively provide such mechanisms and must rely on auxiliary components.

Current evaluation metrics like BLEU, ROUGE, or even BERTScore focus on fluency and lexical similarity, but fail to capture factual correctness [28]. Recent proposals such as FactCC [29], SummaC [30] and TruthfulQA [31] attempt to measure hallucination explicitly, but are primarily calibrated for large models. The lack of SLM-specific hallucination benchmarks limits both the diagnosis and mitigation of hallucination in these smaller models.

For regulated industries or safety-critical applications, hallucinations may not only be unacceptable but also legally actionable. As such, regulatory bodies and enterprises are increasingly requiring explainability, traceability, and auditable decision-making in language models, including SLMs. This necessitates hallucination prevention not just as a technical fix but as part of a broader model governance strategy [32].

3. Performance Challenges in Small Language Models

3.1 Memory and Computational Constraints

A defining characteristic of Small Language Models (SLMs) is their architectural minimalism—typically comprising under a few billion to 500 million parameters, sometimes fewer than 100 million. While this parameter reduction provides benefits in terms of deployment efficiency, energy consumption, and latency, it inherently limits the model's ability to represent, store, and generalize over complex language patterns [14].

The compact size of SLMs reduces the breadth of linguistic and world knowledge that can be encoded in model weights. This becomes particularly evident in complex tasks requiring multiple reasoning steps, deep factual recall, or cross-document synthesis. Moreover, many SLMs must operate under stringent inference constraints in environments such as smartphones, microcontrollers, and edge devices. These settings often restrict the available RAM, storage bandwidth, and energy budget, forcing additional compromises like model quantization or the use of limited-context token windows [33].

Crucially, these limitations also restrict the use of advanced processing techniques such as ensemble methods, retrieval-augmented generation, or large-scale context chaining—all of which are feasible with LLMs but impractical or infeasible in SLM regimes [34]. As a result, SLMs often suffer from underfitting, especially in long-form text generation and tasks requiring structured reasoning.

3.2 Limitations in Representation and Generalization

SLMs must learn to compress vast linguistic structures into compact parameter spaces, often resulting in weaker generalization performance on unseen or rare inputs. Unlike large models that possess significant capacity—allowing them to memorize and

interpolate from large corpora—SLMs operate near the threshold of their learning ability, making them more susceptible to data sparsity, task shift, and domain drift [35].

This leads to several concrete limitations:

- Poor transfer learning: When pre-trained SLMs are adapted to new tasks or domains, they frequently exhibit catastrophic forgetting, where prior knowledge is overwritten by new information [23]. This is especially problematic in multi-domain deployments, where the same model must generalize across different styles, topics, or objectives.
- Reduced contextual sensitivity: Due to smaller attention heads and fewer layers, SLMs often fail to model long-range dependencies in text, which are critical for tasks such as summarization, multi-hop question answering, or document-level sentiment analysis [36].
- Increased reliance on statistical priors: Because of insufficient learning capacity, SLMs tend to default to high-frequency or "safe" outputs from the training distribution, limiting creativity and adaptability in dynamic or evolving environments [37].

In practical applications, these limitations surface as bland, repetitive, or overly generic outputs, particularly in tasks requiring open-ended generation or abstraction beyond surface-level inputs.

3.3 Trade-offs Between Efficiency and Accuracy

A central tension in the design of SLMs is the trade-off between computational efficiency and model accuracy. While SLMs are designed to be lightweight and fast, these benefits often come at the cost of reduced performance on standard NLP benchmarks—especially on tasks requiring semantic nuance, factual grounding, or task-specific adaptation [38].

Trade-off 1: Latency vs. Expressivity

Models designed for fast response—such as those used in interactive systems or edge inference—are optimized for minimal latency, which often necessitates shallow networks, aggressive quantization, or pruning. These operations reduce the model's expressive power and tend to impair output richness and contextual fidelity [39].

Trade-off 2: Generalization vs. Specialization

Highly compact SLMs trained for general use frequently underperform in domain-specific settings, such as legal, biomedical, or financial NLP [40]. Conversely, models fine-tuned for domain specificity may perform poorly on general language tasks [41]. Unlike LLMs, which can afford to retain multipurpose capacities, SLMs often face sharp trade-offs between domain specialization and task versatility [42, 43].

Trade-off 3: Interpretability vs. Optimization

Certain SLM configurations—such as those using low-rank approximations or distilled architectures—offer better interpretability due to their simpler structure, but may not perform optimally on complex tasks [14, 44]. This makes it challenging to strike a balance between transparency and task efficacy, particularly when explainability is a regulatory or business requirement.

Overall, while SLMs hold promise for scalable and democratized NLP, their design is fundamentally constrained by these trade-offs. The implications are profound: achieving performance gains in SLMs requires clever engineering, tailored training regimens, and

often, external augmentation strategies—topics that will be discussed in subsequent sections.

4. Hallucination Prevention Techniques

The challenge of hallucination in Small Language Models (SLMs) has prompted a range of mitigation techniques. Unlike Large Language Models (LLMs), SLMs cannot rely on sheer scale to implicitly encode knowledge or reasoning paths. Instead, hallucination prevention in SLMs must be strategic, modular, and efficient, leveraging methods that compensate for their limited capacity. This section explores five key approaches used to address hallucination in SLMs.

4.1 Knowledge Grounding and Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) is a prominent method that integrates SLMs with external knowledge sources to enhance factual consistency during generation. In this architecture, the model retrieves relevant documents or passages from a knowledge base in real time and tailors its generation on this external context [22]. This decouples the need to store factual knowledge within model parameters—an especially critical adaptation for SLMs with constrained capacity.

For SLMs, retrieval modules are typically lightweight and may rely on sparse retrievers (e.g., BM25) or dense retrievers (e.g., DPR or ColBERT) [45–47]. When combined with efficient attention mechanisms (e.g., late fusion or shallow concatenation), RAG allows SLMs to produce more factually grounded outputs without expanding model size [22].

One implementation strategy involves embedding the retriever inside the inference pipeline, so each query dynamically retrieves the most relevant contexts. Studies have shown that even SLMs under 100M parameters can benefit from plug-and-play grounding, reducing hallucination rates in QA and summarization by 20–30% [19, 48].

However, RAG introduces dependencies on retrieval accuracy and latency, which may limit real-time applications or those operating in offline environments.

4.2 Instruction Tuning and Prompt Engineering

Instruction tuning is a method where models are trained to follow task-specific prompts formulated as natural language instructions [49]. For SLMs, this technique enhances alignment with user intent and improves model behavior across tasks where factual accuracy is paramount.

By carefully designing prompts—either manually or through automated prompt optimization—practitioners can steer the SLM toward safer, less hallucination-prone outputs. For example, explicitly instructing the model to “only answer if certain” or “cite supporting facts” can trigger internal constraints that limit speculative generation.

Instruction tuning can be implemented through multi-task fine-tuning on datasets like FLAN, which include thousands of labeled instruction-following examples [50, 51]. For SLMs, this tuning process often yields disproportionate gains, effectively compressing task knowledge into fewer parameters and reducing hallucination without architectural changes.

Prompt engineering, on the other hand, is a zero-shot or few-shot technique where hallucination is mitigated at inference time by controlling the prompt format, context framing, and output constraints [52]. Techniques such as prompt chaining, structured templates, and declarative framing have shown to reduce hallucination frequency, especially in summarization and QA tasks.

Nevertheless, instruction tuning requires access to curated datasets, and prompt engineering may lack robustness across diverse domains.

4.3 Fact-Checking and Post-Generation Verification

Another viable strategy for hallucination prevention is to evaluate and filter model outputs through automated fact-checking systems. These systems can operate as post-processing layers that assess whether a generated statement is supported by available evidence [53].

For SLMs, lightweight classifiers or entailment models—fine-tuned on verification datasets like FEVER or SciFact—can be used to score the factual consistency of generated text [54, 55]. This two-pass architecture ensures that hallucinated outputs are flagged, re-ranked, or rejected before delivery to users.

Some frameworks also incorporate self-checking mechanisms, where the SLM itself is queried post-generation to verify its prior output. While promising, such methods require careful calibration to avoid recursive error propagation.

In deployment settings where SLMs serve real users, incorporating feedback-based rejection sampling—where only outputs passing factuality filters are accepted—can significantly reduce the incidence of hallucination. However, this may increase latency or reduce generation fluency if over-applied.

4.4 Calibration Techniques (e.g., Temperature Scaling, Confidence Regularization)

Calibration techniques adjust the model's output distribution to better reflect its confidence in generation, thereby reducing speculative or hallucinated completions. One common technique is temperature scaling, which adjusts the softmax temperature during decoding to control the diversity and randomness of generated outputs [56].

For SLMs, using lower temperatures typically results in more conservative and factually stable outputs, though sometimes at the cost of creativity or informativeness. Top-k or nucleus sampling (top-p) can further constrain the token space to probable completions, reducing hallucination risk.

Another promising technique is confidence regularization during training, where the model is penalized for generating high-confidence outputs that are later found to be incorrect. These approaches aim to align confidence with correctness, allowing downstream components to use confidence scores as proxies for factual reliability [27].

For instance, calibrating confidence thresholds to trigger rejection or fallback queries can empower SLMs with error-aware control logic without altering the base architecture.

4.5 Fine-Tuning with Human Feedback

Fine-tuning with human feedback represents a powerful technique for aligning SLM behavior with human expectations, especially for controlling hallucinations. The most widely studied method in this category is Reinforcement Learning from Human Feedback (RLHF), which trains the model not only to produce coherent responses, but also to prioritize outputs rated as helpful, harmless, and truthful by human annotators [57].

While RLHF has been predominantly applied to LLMs, its principles are transferable to SLMs with appropriate efficiency adaptations. Instead of full-scale reinforcement learning pipelines, SLMs can leverage simplified forms such as Direct Preference Optimization (DPO), which skips the reinforcement learning step and fine-tunes the model directly on preference pairs [58].

For hallucination prevention, the feedback signal focuses on factuality, coherence, and source fidelity. For example, annotators may rank responses based on factual correctness,

enabling the model to learn implicit representations of truthfulness. Once trained, the model internalizes these preferences and exhibits significantly lower hallucination rates, even under ambiguous or underspecified prompts.

In low-resource settings where human annotation is limited, proxy signals such as automatic entailment scores or factual consistency metrics can be used to simulate feedback. This hybrid approach, though less precise than full human oversight, enables scalable alignment tuning of SLMs with factuality as a core training objective.

However, RLHF and DPO are data-intensive and sensitive to annotation quality. Improper or biased preferences can lead to overfitting or suppression of valid alternative expressions. As such, they are most effective when paired with robust evaluation protocols and diverse human raters.

5. Performance Enhancement Techniques

Due to their constrained capacity, Small Language Models (SLMs) require tailored techniques to maximize their performance across diverse tasks. Unlike Large Language Models (LLMs), which benefit from sheer scale, SLMs must leverage parameter-efficient, resource-aware strategies to reach comparable utility. This section reviews five such methods, focusing on architecture optimization, training efficiency, and knowledge transfer.

5.1 Quantization and Pruning

Quantization refers to the process of converting a model's weights and activations from high-precision (e.g., 32-bit floating point) to lower precision formats such as 8-bit, 4-bit, or even binary representations [15]. This reduces the model size and memory footprint, allowing for faster inference and deployment on edge devices without necessarily retraining the model from scratch.

SLMs are particularly suited to quantization because their smaller architectures make quantization-aware training and fine-tuning more tractable. Empirical studies have shown that well-quantized SLMs can preserve over 95% of their original performance, even in dense generative tasks [59].

Pruning, on the other hand, involves removing redundant weights, attention heads, or entire layers based on importance scores. Methods such as magnitude pruning, structured pruning, and movement pruning can reduce the computational load and inference latency. For SLMs, pruning must be applied judiciously to avoid over-pruning, which can lead to severe degradation due to their limited redundancy [60].

Together, quantization and pruning serve as key enablers for real-time, low-power applications, including chatbots, mobile summarizers, and translation systems.

5.2 Low-Rank Adaptation and Parameter-Efficient Fine-Tuning

Low-Rank Adaptation (LoRA) is a fine-tuning technique that freezes the original model weights and injects small trainable matrices into specific layers [38]. This significantly reduces the number of trainable parameters—often by over 90%—while maintaining competitive performance across a wide range of tasks.

LoRA is particularly impactful for SLMs because it allows for task-specific fine-tuning with minimal memory and compute cost, enabling a single base model to serve multiple use cases. For instance, fine-tuning a 100M parameter model with LoRA may only require adjusting a few million parameters, making it feasible even in low-resource or on-device settings.

Other parameter-efficient tuning (PET) methods include adapters, prefix-tuning, and $(IA)^3$, each of which introduces small-scale modifications to model internals while preserving generalization [61–63]. These approaches not only improve modularity and task specialization, but also reduce the risk of catastrophic forgetting, making them ideal for multi-domain deployments [64]. PET techniques have been shown to outperform full fine-tuning in data-limited scenarios and exhibit strong transferability across domains, making them foundational to modern SLM pipelines [15, 19].

5.3 Knowledge Distillation from LLMs

Knowledge distillation involves training a compact student model (e.g., an SLM) to imitate the behaviors, logits, or hidden representations of a larger, pretrained teacher model (e.g., an LLM). The goal is to transfer knowledge from high-capacity models into smaller, deployable ones [65].

In SLMs, distillation can take several forms:

- Logit-based distillation, where the student learns to match the soft target distributions of the teacher.
- Intermediate representation distillation, aligning hidden states or attention maps.
- Response-based distillation, where the student mimics the outputs of the teacher in generative or classification tasks.

Modern techniques such as TinyBERT and DistilBERT have shown that carefully distilled SLMs can achieve up to 97% of the teacher's accuracy on GLUE benchmarks while being 40–60% faster [60, 66].

Distillation is especially useful when pretraining from scratch is infeasible, and when deploying models in latency-sensitive environments. However, distillation quality is highly dependent on teacher diversity, task alignment, and training signal richness.

5.4 Mixture of Experts (MoE) for SLMs

Although traditionally associated with LLMs, Mixture of Experts (MoE) architectures have recently been adapted for use in SLMs. MoE models consist of multiple expert subnetworks, with a gating mechanism that activates only a subset during inference, thereby increasing capacity without linearly increasing compute [67].

In the context of SLMs, sparse MoE architectures can be implemented by using small-scale expert modules specialized for tasks, domains, or languages. When only a few experts are active at each step, the effective model size grows, but the runtime cost remains constant. This enables SLMs to achieve LLM-level performance on specific subtasks while remaining efficient [68].

Some lightweight MoE models—such as Switch Transformers Lite—have demonstrated that even models with under 200M parameters can benefit from expert sparsity, especially in multitask learning and dialogue generation [68]. Key challenges remain in balancing expert usage and avoiding overfitting to particular experts, but the approach is promising for scalable SLM enhancement.

5.5 Specialized Training Regimes (Task-specific and Domain-adaptive)

Another impactful strategy for improving SLM performance is the use of specialized training regimes, tailored to either a specific task or a target domain. Since SLMs lack the overparameterization of LLMs, their ability to generalize broadly is constrained—making focused and deliberate training essential to maximize utility.

Task-specific fine-tuning

SLMs can be fine-tuned on carefully curated task datasets (e.g., sentiment analysis, summarization, NER) using supervised learning. When the dataset is well-matched and of sufficient quality, even a low size model can approach or exceed LLM performance on narrow tasks [69]. Fine-tuning with task-specific loss functions, such as factuality-aware or span-based objectives, further optimizes model behavior toward the desired output format.

Domain-adaptive pretraining (DAPT)

In cases where the target domain differs significantly from general corpora (e.g., legal, medical, scientific text), Domain-Adaptive Pretraining helps by continuing pretraining on unlabeled in-domain data before fine-tuning. This enables the SLM to learn domain-specific vocabulary, syntax, and concepts that would otherwise be underrepresented in general training data [70].

DAPT is particularly effective for SLMs because it allows for efficient specialization without complete retraining. In biomedical applications, for example, BioBERT and SciBERT-style adaptations have produced significant gains using only modestly sized models [40, 71].

Multi-task and Curriculum Learning

Advanced regimes also include multi-task learning, where the model is trained simultaneously on related tasks (e.g., QA + summarization), and curriculum learning, where training progresses from simple to complex examples. These techniques help mitigate overfitting and promote better generalization—critical properties for underparameterized SLMs [72].

In all cases, the success of these regimes depends on the alignment between training data and deployment scenarios. SLMs are more brittle than LLMs under distribution shift, and thus require more careful data selection and evaluation.

6. Comparative Analysis

This section synthesizes the techniques introduced in previous discussions by conducting a comparative analysis in terms of their effectiveness, computational efficiency, scalability, and applicability to real-world deployment of Small Language Models. Given the diversity of methods available, understanding which to use, when, and how to combine them is essential for researchers and engineers seeking to maximize both factual accuracy and functional performance under limited resources. Table 1 compares considered hallucination prevention and performance enhancement methods

6.1 Resource Requirements vs. Performance Gains

The utility of a method often hinges on the performance gain it offers relative to its cost in resources—a central concern in SLM development. Techniques like LoRA, quantization, and instruction tuning strike a particularly favorable balance. LoRA enables up to 90% reduction in trainable parameters with minimal performance loss [38], while 8-bit quantization can reduce memory footprint by 4× with <2% degradation in accuracy [59].

By contrast, methods such as RLHF and RAG yield high returns in hallucination reduction but incur substantial compute and human annotation overheads. RLHF, for instance, requires thousands of preference-labeled instances and multiple optimization stages [57]. RAG's dependence on real-time retrieval and memory access may disqualify it from edge-device deployments.

Table 1. Comparison of considered hallucination prevention and performance enhancement methods

Method	Category	Advantages	Disadvantages	Typical Use Case
Retrieval-Augmented Generation (RAG)	Hallucination Prevention	Strong grounding; reduces factual errors; domain adaptable	Requires external retriever and memory; slower inference	QA, document summarization, open-domain tasks
Instruction Tuning	Hallucination Prevention	Improves alignment and intent-following; reusable across tasks	Needs curated datasets; sensitive to prompt phrasing	Multi-task systems, APIs, general assistants
Fact-checking/ Verification	Hallucination Prevention	Adds post-hoc quality control; modular	Increases latency; may miss subtle hallucinations	Critical domains (e.g., medical, legal)
Calibration Techniques	Hallucination Prevention	Controls randomness; enhances output reliability	Reduces diversity; requires tuning	Controlled generation; high-stakes applications
RLHF / DPO	Hallucination Prevention	Aligns outputs to human preferences; adaptable across models	High annotation cost; risk of reward hacking	Safety-critical dialogue, personal assistants
Quantization & Pruning	Performance Enhancement	Reduces memory and compute; enables edge deployment	Can degrade accuracy if aggressive	Real-time applications, mobile devices
LoRA / Parameter-Efficient Tuning	Performance Enhancement	Minimal resource use; enables fast adaptation	Adds complexity to training pipeline	Domain-specific or multi-task settings
Knowledge Distillation	Performance Enhancement	Transfers LLM knowledge into compact models; scalable	Dependent on teacher quality; less adaptable post-distillation	Model compression, enterprise deployments
Mixture of Experts (MoE)	Performance Enhancement	Expands capacity without linear compute growth	Complex routing; hard to train and balance	Specialized agents, multitask systems
Domain-Adaptive & Task-Specific Tuning	Performance Enhancement	Highly effective in matched domains; easy to implement	Narrow generalization; requires curated data	Biomedical NLP, legal analysis, customer service

In low-resource or latency-sensitive environments, quantization, fact-checking with lightweight classifiers, and prompt engineering are often the only viable options. Meanwhile, infrastructure-rich settings (e.g., cloud services, research labs) can afford to implement more intensive techniques like RLHF or dynamic MoE routing for SLM orchestration.

6.2 When to Use What: Use-Case-Driven Combinations

Selecting and combining hallucination prevention and performance enhancement techniques depends heavily on the deployment context, available resources, and task criticality. While individual methods offer isolated benefits, their synergistic use often yields optimal performance. The following scenarios illustrate combinations of methods best suited for typical SLM applications:

- Mobile or On-Device NLP Applications: For environments with severe compute and memory constraints, a combination of 8-bit quantization, structured pruning, and parameter-efficient fine-tuning (e.g., LoRA) allows for real-time inference with minimal loss in performance [15, 38]. Temperature tuning can be applied to suppress hallucinated content while preserving fluency.
- Enterprise Question Answering Systems: Applications such as legal or technical support benefit from retrieval-augmented generation (RAG) combined with domain-adaptive pretraining [70] and instruction tuning [73]. For high-fidelity tasks, integrating post-hoc verification improves factual reliability, as shown in medical summarization and scientific QA contexts [53].
- Multi-Task Digital Assistants: Assistant models designed for varied conversational goals should employ LoRA or adapters for modular task specialization [64], augmented by multi-task instruction tuning and direct preference optimization (DPO) to reduce hallucination and align outputs with user expectations [58].
- Safety-Critical Applications (e.g., Healthcare, Finance): In these domains, hallucination poses legal and ethical risks. Here, low-temperature decoding, confidence regularization, and factuality-based filtering pipelines are essential. Fine-tuning with human-labeled preference datasets [57] further ensures output safety.
- Content Creation and Education Tools: These applications may prioritize creativity and fluency but still require factual grounding. Distillation from high-performing LLMs and structured prompt engineering are effective here [14, 52, 66]. When available, RAG modules improve informativeness with minimal hallucination.

Each of these combinations showcases the adaptability of SLMs when engineering constraints and task-specific needs are jointly considered.

6.3 Scalability and Portability Across Platforms and Domains

The scalability and portability of hallucination prevention and performance enhancement methods vary significantly depending on their algorithmic structure and infrastructure dependency.

- Highly Portable Methods: Techniques such as quantization, LoRA, and instruction tuning are architecturally non-invasive and computationally efficient, making them suitable for cross-platform deployment, including mobile, edge, and embedded devices [15, 38, 49]. Their lightweight implementation also facilitates deployment in environments with intermittent connectivity or offline constraints.
- Limited Portability Techniques: Strategies like retrieval-augmented generation, reinforcement learning from human feedback (RLHF), and mixture-of-experts

(MoE) are resource-intensive and require supporting infrastructure (e.g., real-time retrieval databases, expert gating mechanisms, or human annotation loops). While effective in high-resource settings, they are challenging to scale to decentralized or embedded systems [57, 68].

- Cross-Domain Generalizability: Methods like instruction tuning, distillation, and multi-task fine-tuning have shown high generalizability across NLP benchmarks and languages [66, 74]. However, domain-adaptive pretraining remains critical for performance in specialized fields such as medicine or law, where general-domain pretraining is insufficient [70].

In summary, method selection should account for not only model size and accuracy, but also operational constraints such as target hardware, latency tolerance, and domain specificity. A modular, composable approach offers the best path toward scalable, efficient, and reliable SLM deployment.

7. Discussion

One of the most pressing challenges in optimizing Small Language Models (SLMs) is achieving compatibility among multiple performance and hallucination mitigation strategies. As shown in the prior sections, many techniques target different dimensions of model behavior—some modify the base architecture (e.g., pruning, LoRA), while others affect training procedures (e.g., instruction tuning, RLHF), and still others operate externally (e.g., fact-checkers, RAG modules).

Successfully integrating these approaches requires attention to their interdependencies and interference potential. For example, deploying quantization alongside RAG may lead to mismatches between token embeddings and retrieval vectors unless carefully aligned [15]. Similarly, combining LoRA with RLHF necessitates tuning not only reward signals but also how and where LoRA adapters are applied—especially in transformer attention layers [38, 58].

There is also the question of interoperability across tools and pipelines. For instance, many fine-tuning and alignment strategies assume access to open-weight models and fine-grained control over layers. This poses a challenge when using proprietary or API-constrained SLMs, where only prompt-level tuning is possible. In such cases, prompt engineering and calibration become the only feasible levers, despite their lower ceiling on performance.

Thus, the future of effective SLM deployment likely lies in modular, plug-and-play systems that allow for composable combinations of retrieval, tuning, and filtering techniques, each calibrated to the resource and risk profile of the application.

7.1 Sustainability and Cost-Efficiency Considerations

While much of the focus in NLP research has been on maximizing performance, the rising emphasis on sustainability and cost-efficiency compels a re-evaluation of model design priorities. Large Language Models have been criticized for their carbon footprint, hardware demands, and data inefficiency [46]. In contrast, SLMs offer a promising alternative—but only if their enhancement techniques are themselves sustainable.

Many of the most effective hallucination prevention and performance boosting methods—such as RLHF, RAG, or MoE—require significant training or infrastructure overhead, which can offset the lightweight benefits of the base model. A key direction for future work involves developing low-cost analogues of these methods that retain most of the

performance gains. For example, pseudo-labeling for alignment or factuality-aware pretraining might provide lightweight alternatives to human-in-the-loop approaches [21].

Moreover, SLM-centric design needs to shift toward lifecycle-aware modeling, where training, deployment, and continual learning are planned with compute budgets in mind. Techniques such as progressive freezing, dynamic fine-tuning, and on-device continual learning could allow for long-term use with minimal retraining [75].

7.2 Ethical Considerations and Transparency

The deployment of SLMs—especially in high-stakes and consumer-facing contexts—raises important ethical questions, particularly around trust, transparency, and safety. Even though SLMs are less capable than LLMs, they can still produce hallucinated content that users mistake for truth, particularly due to their fluent style and confident tone [13]. This risk is magnified in applications involving children, vulnerable populations, or marginalized groups.

One promising mitigation strategy is to embed factual confidence scores or disclaimers into model outputs—an approach shown to improve user calibration and reduce overreliance on model suggestions [27]. Moreover, transparency about model capabilities and limitations, including accuracy rates and hallucination tendencies, should be a standard component of model documentation [76].

From a governance perspective, organizations deploying SLMs should consider auditable logs of model decisions, especially when using post-generation filtering or human feedback integration. This is critical for ensuring accountability in regulated domains such as healthcare, finance, and education.

Finally, the development of benchmark suites tailored for SLMs, particularly around factuality and task alignment, is needed. Current evaluation standards disproportionately favor LLMs and do not adequately reflect the real-world constraints or behaviors of smaller models.

7.3 Future Directions in SLM Optimization

Several open research directions emerge from this review:

- SLM-centric pretraining paradigms: Rather than distilling from LLMs or truncating larger architectures, future models could be trained from scratch using SLM-optimized curricula that prioritize factual grounding, compression efficiency, and domain relevance.
- Neural-symbolic hybrids: Combining compact language models with symbolic reasoning engines or structured databases may offer improved reliability, especially in logic-heavy or tabular domains [77].
- Benchmarking under constraints: New evaluation datasets and leaderboards should be created that explicitly account for compute, memory, and latency limits, promoting fair and transparent comparisons of SLM capabilities.
- Factuality-aware decoding strategies: Advances in controllable generation could empower SLMs to internally regulate hallucinations during output generation, obviating the need for post-processing in many contexts.

Overall, the path forward involves rethinking NLP not only as a scale game, but as a systems optimization problem—balancing factuality, efficiency, and trust in constrained environments.

8. Conclusion

This review has examined the hallucination prevention and performance enhancement techniques applicable to Small Language Models (SLMs), with an emphasis on their practical deployment in resource-constrained and real-world settings. In contrast to the dominant narrative centered on Large Language Models (LLMs), this paper highlights that SLMs—despite their limitations in capacity—can be made competent, trustworthy, and efficient through the careful application of modular and synergistic strategies.

For hallucination mitigation, five major approaches were surveyed: retrieval-augmented generation (RAG), instruction tuning and prompt design, post-hoc fact-checking, calibration methods, and preference-based alignment such as RLHF or DPO. Each offers unique strengths and trade-offs. For performance enhancement, methods such as quantization, pruning, parameter-efficient tuning (e.g., LoRA), knowledge distillation, MoE architectures, and domain-adaptive training were reviewed. These strategies have demonstrated substantial gains in both efficiency and task performance, even in low-resource or real-time inference environments.

In light of the comparative analysis no single method seems universally optimal. Rather, method selection should be driven by deployment requirements, such as latency, memory constraints, and domain specificity. Effective SLM pipelines often involve hybrid configurations, balancing accuracy with computational feasibility.

Based on the findings, we offer the following recommendations for practitioners and researchers working with SLMs:

- For On-Device Deployment: Prioritize quantization, pruning, and parameter-efficient fine-tuning to balance speed and accuracy. Use calibration techniques like temperature scaling to manage hallucination risk without incurring additional compute.
- For Domain-Specific Applications: Apply domain-adaptive pretraining and instruction tuning. Where feasible, integrate lightweight retrieval modules or task-specific factuality filters.
- For Critical Systems (e.g., medical, legal): Combine instruction tuning with output verification modules. Prefer conservative decoding strategies and, when possible, introduce RLHF or DPO to enforce user-aligned safety.
- For Scalable Multi-Task Systems: Use LoRA with shared base models and task-specific adapters. Consider prompt standardization or instruction chaining to reduce hallucination and improve generalization.
- For Model Development and Research: Invest in benchmarks and metrics tailored to SLMs—especially ones that emphasize factuality, robustness, and cost-performance balance.

One of the important gaps in the current research landscape is the lack of standardized, publicly available benchmarks for evaluating SLMs on both factuality and efficiency metrics. Existing evaluation protocols tend to overfit the needs of LLMs, emphasizing scale-driven generalization and zero-shot capabilities. These metrics fail to capture the pragmatic trade-offs that dominate SLM usage—such as inference speed, memory use, and factual error rates in low-capacity regimes.

SLM-specific evaluation suites that incorporate the following may be beneficial for further advancement of the field:

- Task coverage diversity, including both generative and discriminative benchmarks;

- Multi-domain factuality tests, ideally linked to human-annotated ground truths;
- Efficiency-performance trade-off metrics, measuring energy consumption, latency, and model size against output quality;
- Robustness to prompt variation and domain drift, assessing stability and hallucination under real-world noise.

The development of such benchmarks may be regarded as important to advancing the responsible and effective deployment of SLMs in academic, commercial, and public-interest settings.

References

[1] Liddy ED. Natural language processing. In: Encyclopedia of Library and Information Science. 2nd Ed. New York: Marcel Decker, Inc.; 2001.

[2] Jurafsky D, Martin JH. *Speech and Language Processing*. Pearson; 2023.

[3] Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space. 2013 http://arxiv.org/abs/1301.3781.

[4] Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014. p. 1532–43.

[5] Peters M, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, et al. Deep Contextualized Word Representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics; 2018. p. 2227–37.

[6] Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North. Stroudsburg, PA, USA: Association for Computational Linguistics; 2019. p. 4171–86.

[7] Radford A, Narasimhan K, Salimans T, Sutskever I, others. Improving language understanding by generative pre-training. 2018;

[8] Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I, et al. Language models are unsupervised multitask learners. OpenAI blog. 2019; 1(8):9.

[9] Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language Models are Few-Shot Learners. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, editors. Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2020. p. 1877–901.

[10] Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, et al. PaLM: Scaling Language Modeling with Pathways. 2022 http://arxiv.org/abs/2204.02311.

[11] OpenAI, Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, et al. GPT-4 Technical Report. 2023 http://arxiv.org/abs/2303.08774.

[12] Alizadeh K, Mirzadeh SI, Belenko D, Khatamifard S, Cho M, Del Mundo CC, et al. LLM in a flash: Efficient large language model inference with limited memory. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024. p. 12562–84.

[13] Ji Z, Lee N, Frieske R, Yu T, Su D, Xu Y, et al. Survey of Hallucination in Natural Language Generation. ACM Comput Surv. 2023; 55(12):1–38.

[14] Turc I, Chang M-W, Lee K, Toutanova K. Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. 2019 http://arxiv.org/abs/1908.08962.

[15] Dettmers T, Pagnoni A, Holtzman A, Zettlemoyer L. QLoRA: Efficient Finetuning of Quantized LLMs. 2023 http://arxiv.org/abs/2305.14314.

[16] Maynez J, Narayan S, Bohnet B, McDonald R. On Faithfulness and Factuality in Abstractive Summarization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational

Linguistics; 2020. p. 1906–19.
[17] Naveen B, Ashita S, Pushpak B. NLI to the Rescue: Mapping Entailment Classes to Hallucination Categories in Abstractive Summarization. In: Proceedings of the 20th International Conference on Natural Language Processing (ICON). 2023. p. 120–32.
[18] Honovich O, Aharoni R, Herzig J, Taitelbaum H, Kukliansy D, Cohen V, et al. TRUE: Re-evaluating Factual Consistency Evaluation. In: Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering. Stroudsburg, PA, USA: Association for Computational Linguistics; 2022. p. 161–75.
[19] Shuster K, Poff S, Chen M, Kiela D, Weston J. Retrieval Augmentation Reduces Hallucination in Conversation. 2021 http://arxiv.org/abs/2104.07567.
[20] Chae K, Choi J, Jo Y, Kim T. Mitigating Hallucination in Abstractive Summarization with Domain-Conditional Mutual Information. In: Findings of the Association for Computational Linguistics: NAACL. 2024. p. 1809–20.
[21] Huang S, Dong L, Wang W, Hao Y, Singhal S, Ma S, et al. Language is not all you need: Aligning perception with language models. Adv Neural Inf Process Syst. 2023; 36:72096–109.
[22] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. Adv Neural Inf Process Syst. 2020; 33:9459–74.
[23] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res. 2020; 21(140):1–67.
[24] Olshaker H, Brin D, Kalderon E, Kraus M, Konen E, Klang E. Evaluating the Diagnostic Performance of Large Language Models in Identifying Complex Multisystemic Syndromes: A Comparative Study with Radiology Residents. 2024.
[25] Gaube S, Suresh H, Raue M, Merritt A, Berkowitz SJ, Lermer E, et al. Do as AI say: susceptibility in deployment of clinical decision-aids. npj Digit Med. 2021; 4(1):31.
[26] Ghweeba M, Lindenmeyer A, Shishi S, Waheed A, Kofi M, Amer S. The Attitudes of Egyptian Web-Based Health Information Seekers Toward Health Information Provided Through the Internet: Qualitative Study. JMIR Form Res. 2022; 6(2):e30108.
[27] Kadavath S, Conerly T, Askell A, Henighan T, Drain D, Perez E, et al. Language Models (Mostly) Know What They Know. 2022 http://arxiv.org/abs/2207.05221.
[28] Fabbri AR, Kryściński W, McCann B, Xiong C, Socher R, Radev D. Summeval: Re-evaluating summarization evaluation. Trans Assoc Comput Linguist. 2021; 9:391–409.
[29] Kryściński W, McCann B, Xiong C, Socher R. Evaluating the Factual Consistency of Abstractive Text Summarization. 2019 http://arxiv.org/abs/1910.12840.
[30] Laban P, Schnabel T, Bennett PN, Hearst MA. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. Trans Assoc Comput Linguist. 2022; 10:163–77.
[31] Lin S, Hilton J, Evans O. TruthfulQA: Measuring How Models Mimic Human Falsehoods. 2021 http://arxiv.org/abs/2109.07958.
[32] Weidinger L, Uesato J, Rauh M, Griffin C, Huang P-S, Mellor J, et al. Taxonomy of risks posed by language models. In: Proceedings of the 2022 ACM conference on fairness, accountability, and transparency. 2022. p. 214–29.
[33] Kim G, Cho K. Length-Adaptive Transformer: Train Once with Length Drop, Use Anytime with Search. 2020 http://arxiv.org/abs/2010.07003.
[34] Wang Y, Zhong W, Li L, Mi F, Zeng X, Huang W, et al. Aligning Large Language Models with Human: A Survey. 2023 http://arxiv.org/abs/2307.12966.
[35] Ouyang L, Wu J, Jiang X, Almeida D, Wainwright C, Mishkin P, et al. Training language models to follow instructions with human feedback. Adv Neural Inf Process Syst. 2022; 35:27730–44.
[36] Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. 2019

http://arxiv.org/abs/1909.11942.

[37] Schick T, Schütze H. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, PA, USA: Association for Computational Linguistics; 2021. p. 2339–52.

[38] Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, et al. LoRA: Low-Rank Adaptation of Large Language Models. 2021 http://arxiv.org/abs/2106.09685.

[39] Zhang W, Hou L, Yin Y, Shang L, Chen X, Jiang X, et al. TernaryBERT: Distillation-aware Ultra-low Bit BERT. 2020 http://arxiv.org/abs/2009.12812.

[40] Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Wren J, editor. Bioinformatics. 2020; 36(4):1234–40.

[41] Alsentzer E, Murphy JR, Boag W, Weng W-H, Jin D, Naumann T, et al. Publicly Available Clinical BERT Embeddings. 2019 https://arxiv.org/abs/1904.03323.

[42] Grassi L, Recchiuto CT, Sgorbissa A. Knowledge-grounded dialogue flow management for social robots and conversational agents. Int J Soc Robot. 2022; 14(5):1273–93.

[43] Van Nguyen C, Shen X, Aponte R, Xia Y, Basu S, Hu Z, et al. A Survey of Small Language Models. 2024 http://arxiv.org/abs/2410.20011.

[44] Wang S, Li BZ, Khabsa M, Fang H, Ma H. Linformer: Self-Attention with Linear Complexity. 2020 http://arxiv.org/abs/2006.04768.

[45] Robertson S, Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond. Found Trends® Inf Retr. 2009; 3(4):333–89.

[46] Khattab O, Zaharia M. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY, USA: ACM; 2020. p. 39–48.

[47] Karpukhin V, Oguz B, Min S, Lewis P, Wu L, Edunov S, et al. Dense Passage Retrieval for Open-Domain Question Answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg, PA, USA: Association for Computational Linguistics; 2020. p. 6769–81.

[48] Izacard G, Grave E. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. 2020 http://arxiv.org/abs/2007.01282.

[49] Wei J, Bosma M, Zhao VY, Guu K, Yu AW, Lester B, et al. Finetuned Language Models Are Zero-Shot Learners. 2021 http://arxiv.org/abs/2109.01652.

[50] Chung HW, Hou L, Longpre S, Zoph B, Tay Y, Fedus W, et al. Scaling Instruction-Finetuned Language Models. 2022 http://arxiv.org/abs/2210.11416.

[51] Longpre S, Hou L, Vu T, Webson A, Chung HW, Tay Y, et al. The flan collection: Designing data and methods for effective instruction tuning. In: International Conference on Machine Learning. 2023. p. 22631–48.

[52] Wang Y, Wang M, Manzoor MA, Liu F, Georgiev G, Das RJ, et al. Factuality of Large Language Models: A Survey. 2024 http://arxiv.org/abs/2402.02420.

[53] Atanasova P. Generating fact checking explanations. In: Accountable and Explainable Methods for Complex Reasoning over Text. Springer; 2024. p. 83–103.

[54] Thorne J, Vlachos A, Christodoulopoulos C, Mittal A. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics; 2018. p. 809–19.

[55] Wadden D, Lin S, Lo K, Wang LL, van Zuylen M, Cohan A, et al. Fact or Fiction: Verifying Scientific Claims. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg, PA, USA: Association for Computational Linguistics; 2020. p. 7534–50.

[56] Guo C, Pleiss G, Sun Y, Weinberger KQ. On calibration of modern neural networks. In: International conference on machine learning. 2017. p. 1321–30.

[57] Christiano PF, Leike J, Brown T, Martic M, Legg S, Amodei D. Deep reinforcement learning from human preferences. Adv Neural Inf Process Syst. 2017; 30.

[58] Rafailov R, Sharma A, Mitchell E, Ermon S, Manning CD, Finn C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. 2023 http://arxiv.org/abs/2305.18290.

[59] Frantar E, Ashkboos S, Hoefler T, Alistarh D. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. 2022 http://arxiv.org/abs/2210.17323.

[60] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2020 https://arxiv.org/abs/1910.01108.

[61] Li XL, Liang P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics; 2021. p. 4582–97.

[62] Houlsby N, Giurgiu A, Jastrzebski S, Morrone B, De Laroussilhe Q, Gesmundo A, et al. Parameter-efficient transfer learning for NLP. In: International conference on machine learning. 2019. p. 2790–9.

[63] Liu H, Tam D, Muqeeth M, Mohta J, Huang T, Bansal M, et al. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. Adv Neural Inf Process Syst. 2022; 35:1950–65.

[64] Mahabadi RK, Ruder S, Dehghani M, Henderson J. Parameter-efficient Multi-task Fine-tuning for Transformers via Shared Hypernetworks. 2021 http://arxiv.org/abs/2106.04489.

[65] Hinton G, Vinyals O, Dean J. Distilling the Knowledge in a Neural Network. 2015 http://arxiv.org/abs/1503.02531.

[66] Jiao X, Yin Y, Shang L, Jiang X, Chen X, Li L, et al. TinyBERT: Distilling BERT for Natural Language Understanding. In: Findings of the Association for Computational Linguistics: EMNLP 2020. Stroudsburg, PA, USA: Association for Computational Linguistics; 2020. p. 4163–74.

[67] Shazeer N, Mirhoseini A, Maziarz K, Davis A, Le Q, Hinton G, et al. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. 2017 http://arxiv.org/abs/1701.06538.

[68] Fedus W, Zoph B, Shazeer N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. J Mach Learn Res. 2022; 23(120):1–39.

[69] Howard J, Ruder S. Universal Language Model Fine-tuning for Text Classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics; 2018. p. 328–39.

[70] Gururangan S, Marasović A, Swayamdipta S, Lo K, Beltagy I, Downey D, et al. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. 2020 http://arxiv.org/abs/2004.10964.

[71] Beltagy I, Lo K, Cohan A. SciBERT: A Pretrained Language Model for Scientific Text. 2019 http://arxiv.org/abs/1903.10676.

[72] Xu B, Zhang L, Mao Z, Wang Q, Xie H, Zhang Y. Curriculum learning for natural language understanding. In: Proceedings of the 58th annual meeting of the association for computational linguistics. 2020. p. 6095–104.

[73] Kojima T, Gu SS, Reid M, Matsuo Y, Iwasawa Y. Large Language Models are Zero-Shot Reasoners. 2022 http://arxiv.org/abs/2205.11916.

[74] Sanh V, Webson A, Raffel C, Bach SH, Sutawika L, Alyafeai Z, et al. Multitask Prompted Training Enables Zero-Shot Task Generalization. 2021 http://arxiv.org/abs/2110.08207.

[75] Aghajanyan A, Shrivastava A, Gupta A, Goyal N, Zettlemoyer L, Gupta S. Better Fine-Tuning by Reducing Representational Collapse. 2020 http://arxiv.org/abs/2008.03156.

[76] Mitchell M, Wu S, Zaldivar A, Barnes P, Vasserman L, Hutchinson B, et al. Model cards for model reporting. In: Proceedings of the conference on fairness, accountability, and transparency. 2019. p. 220–9.

[77] Chen X, Liang C, Yu AW, Zhou D, Song D, Le Q V. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. In: International Conference on Learning Representations. 2019.