Survey

What is the Role of Small Models in the LLM Era: A Survey

Lihu Chen*[1], Gaël Varoquaux[2]

[1] Imperial College London, UK
lihu.chen@imperial.ac.uk
[2] Soda, Inria Saclay, France
gael.varoquaux@inria.fr

arXiv:2409.06857v9 [cs.CL] 21 Sep 2026

Large Language Models (LLMs) have demonstrated remarkable capabilities in various reasoning tasks, which leads to the development of increasingly large models. However, scaling up model sizes results in significantly higher computational costs and energy consumption, which makes these models impractical for academic researchers and businesses with limited resources. At the same time, Small Models (SMs) are frequently used in practical settings, although their significance is currently underestimated. This raises important questions about the role of small models in the era of LLMs, a topic that has received limited attention in prior surveys. In this work, we systematically examine the relationship between LLMs and SMs from two key perspectives: Collaboration *and* Competition (*or Complementarity*)*. We hope this survey provides valuable insights for practitioners, fostering a deeper understanding of the contribution of small models and promoting more efficient use of computational resources. Code and additional resources for this survey are available at:* https://github.com/tigerchen52/awesome_role_of_small_models

1. Introduction

The fast progress of Large Language Models (LLMs) has transformed the field of natural language processing (NLP). LLMs have demonstrated exceptional performance across a range of tasks, including language generation (Dong et al. 2024), language understanding (Wang et al. 2019), and domain-specific applications such as coding (Jiang et al. 2026), medicine (He et al. 2023), and law (Sun 2023). Notably, certain capabilities are enhanced by increasing the size of models, with some abilities only visible in larger models (Wei et al. 2022). This has led to a surge of research and development

* Corresponding authors

Action editor: {action editor name}. Submission received: DD Month YYYY; revised version received: DD Month YYYY; accepted for publication: DD Month YYYY.

© 2025

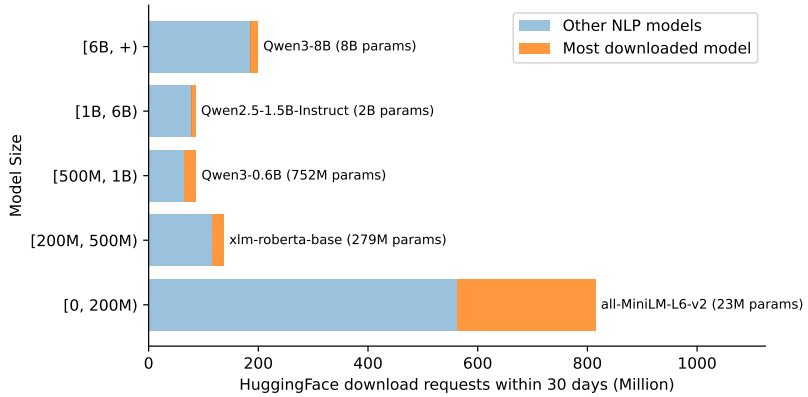


Figure 1: The relationship between model size and monthly downloads. This analysis considers open-source NLP models hosted on HuggingFace and categorizes them into five size groups based on the number of parameters: `[200M, 500M, 1B, 6B]`. For each group, the full bar represents the total monthly download requests of all models, while the highlighted segment corresponds to the most downloaded model. The data was collected on 6 September 2026.

focused on building ever-larger models, such as GPT-4 (OpenAI 2023) and DeepSeek-R1 (DeepSeek-AI et al. 2025).

However, these gains come at a substantial cost. Scaling up model sizes leads to significant increases in computational costs and energy consumption (Wan et al. 2023; Varoquaux, Luccioni, and Whittaker 2025). Additionally, training and deploying LLMs is often impractical in resource-constrained settings, such as for academic researchers, businesses without a strong revenue stream, or deployment on edge devices. As a result, these limitations have motivated a growing interest in the development of small models (SMs) (Lu et al. 2025; Wang et al. 2024b), which can achieve competitive performance with reduced data requirements and fewer parameters.

Recognizing the rise of SMs, we now ask **what qualifies as "small" and what model scope we consider in this survey?** In this work, we use the term *small models* in a broader sense, to include a wider variety of architectures and use-cases.

Definition of Small Models. As counterparts to LLMs, *Small Models* (SMs) generally refer to models with a relatively lower number of parameters. Rather than focusing on a particular model architecture, this survey emphasizes the relationship between small and large models, where the notion of *small* is inherently relative to the reference model. Consequently, SMs may include both encoder-only and decoder-only Transformer language models, as well as, more broadly, other neural architectures or even statistical models. Although the examples discussed in this survey are mainly Transformer-based language models, the underlying principles are not tied to a specific architecture. At the same time, there is no universally accepted definition or parameter threshold separating *large* from *small*. Therefore, we adopt a *relative definition* of model size. For instance, BERT (110M parameters) (Devlin et al. 2019) is considered small relative to LLaMA-8B (Dubey et al. 2024), while LLaMA-8B is small relative to GPT-4 (OpenAI 2023). This

relative perspective ensures that the concepts discussed in this survey remain applicable as model architectures and scales continue to evolve[1].

One may assume that the wide adoption of LLMs makes small models no longer prominent. However, our findings suggest that the usage of small models is significantly underestimated in practical settings. Figure 1, analyzing the downloads of open-source models of various sizes from HuggingFace, shows that smaller models remain highly popular. This raises fundamental questions about the role of small models in the era of LLMs and their *ecological niche* within the broader AI ecosystems, which is a topic seldom discussed in prior surveys.

To assess the role of SMs, it is essential to compare their strengths and weaknesses relative to LLMs. Table 1 highlights four key dimensions to consider:

Accuracy. LLMs have demonstrated superior performance across a wide range of tasks due to their large number of parameters and extensive training on diverse datasets (Raffel et al. 2020; Kaplan et al. 2020). Although SMs generally lag behind in overall performance, they can achieve competitive results when enhanced by techniques such as knowledge distillation (Xu et al. 2024b).

Generality. LLMs are highly generalizable and capable of handling a broad spectrum of tasks with minimal training examples (Dong et al. 2024; Liu et al. 2023). In contrast, SMs are often more specialized, and fine-tuning SMs on domain-specific datasets can outperform general LLMs for specific tasks (Lehman et al. 2023; Bucher and Martini 2024; Zhang et al. 2024b).

Efficiency. LLMs require substantial computational resources for both training and inference (Wan et al. 2023), leading to high costs and latency, making them less practical for real-time applications (e.g. information retrieval (Reimers and Gurevych 2019)) or resource-constrained environments (e.g. edge devices (Dhar et al. 2024)). In contrast, SMs require less training data and computational power, offering competitive performance while significantly reducing resource demands.

Interpretability. Smaller, shallower models tend to be more transparent and interpretable than their larger, deeper counterparts (Gilpin et al. 2018; Barceló et al. 2020). In fields such as healthcare (Caruana et al. 2015), finance (Kurshan et al. 2021), and law (Eliot 2021), smaller models are often preferred because their decisions must be easily understandable by non-experts (e.g. doctors, financial analysts).

In this work, we systematically examine the role of small models in the era of LLMs from two key perspectives: (1) ***Collaboration*** (§ 3). LLMs offer superior accuracy and can handle a wide range of tasks, while SMs are more specialized and cost-effective. In practice, the collaboration between LLMs and SMs can strike a balance between performance and efficiency. (2) ***Competition and Complementarity*** (§ 4). SMs possess distinct advantages, such as simplicity, lower cost, and greater interpretability, and their capabilities can be further enhanced with the help of LLM distillations. It is crucial to carefully assess the trade-offs between LLMs and SMs based on the specific requirements of the task or application. We aim for this study to provide actionable insights for

1 Although we adopt a relative definition of model size, most small models discussed in this survey contain fewer than 1B parameters.

Dimension	LLMs	SMs
Accuracy	state-of-the-art ✓	decent ✗
Generality	general-purpose ✓	task-specific ✗
Efficiency	resource-intensive ✗	resource-efficient ✓
Interpretability	low interpretable ✗	high interpretable ✓

Table 1: Comparisons of different dimensions between LLMs and SMs.

practitioners, especially for researchers and businesses with constrained resources, who are interested in leveraging SMs alongside LLMs. Furthermore, we hope our analysis clarifies the ecological niche of small models in society, which offers design principles and ideas that remain relevant even as model architectures and sizes continue to evolve over time.

2. Related Work

Recently, there has been a noticeable shift in the research community from large language models to small and efficient language models, as small models can achieve competitive performance while effectively reducing computational and monetary overhead. This trend is now supported by several systematic surveys, which we situate and contrast with our work. For example, Lu et al. (2025) provide an overview of 70 state-of-the-art open-source SMs, with a focus on transformer-based, decoder-only language models (100M-5B parameters). Wang et al. (2024b) conduct a comprehensive review of SMs, covering their technical approaches, application scenarios, enhancement methods, collaboration strategies with LLMs, and challenges related to trustworthiness. Van Nguyen et al. (2024) analyze the architectures, training techniques, and compression techniques of SMs. Similarly, Subramanian, Elango, and Gungor (2025) examine ~160 papers about SMs in the 1–8 B parameter range, and discuss how SMs can be used to balance performance, efficiency, and cost.

While existing surveys systematically investigate SMs, these works examine these models in isolation (i.e. their architectures, training, and deployment), rather than emphasizing their relationship with LLMs. In contrast to these works, our survey emphasizes the relationship between large and small models, which extends beyond mere differences in parameter size. Specifically, we examine how small models complement, compete with, and collaborate with large models across different tasks and deployment scenarios. Through this perspective, we aim to discuss the broader role and ecological niche of small models in our society.

3. Collaboration

In the following, we describe a dual collaboration paradigm in which SMs and LLMs mutually enhance one another to optimize resource usage and enhance reasoning capabilities. In Section 3.1, we explore how SMs can strengthen LLMs by guiding training, inference, evaluation, and robustness. Conversely, in Section 3.2, we examine how

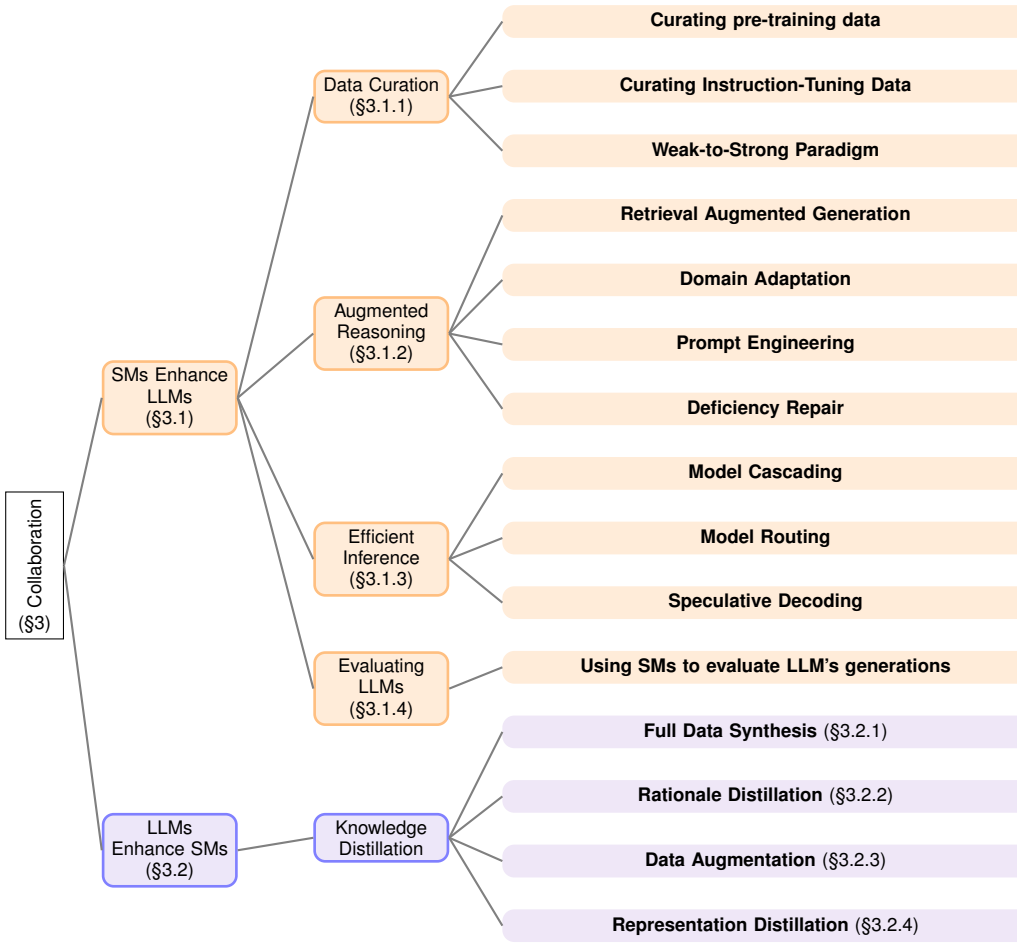


Figure 2: Collaborations between LLMs and SMs

LLMs can, in turn, help SMs by providing richer supervision. The complete interaction framework is illustrated in Figure 2, showing how these two types of models interact and collaborate toward more efficient, powerful systems.

3.1 Small Models Enhance LLMs

To better understand how SMs enhance LLMs, we organize this section according to the life cycle of LLMs: data preparation (pre-training and fine-tuning), inference, and evaluation. First, at the data preparation stage (see Section 3.1.1), SMs contribute to curating pre-training corpora and instruction-tuning datasets, which can filter low-quality or undesired content, and enable weak-to-strong training paradigms that improve supervision quality. Next, at the inference stage, SMs improve reasoning (See Section 3.1.2) and efficiency (See Section 3.1.3) in practical deployment through techniques such as retrieval augmentation, domain adaptation, model routing, and cascading, which help reduce computational cost while maintaining or even enhancing

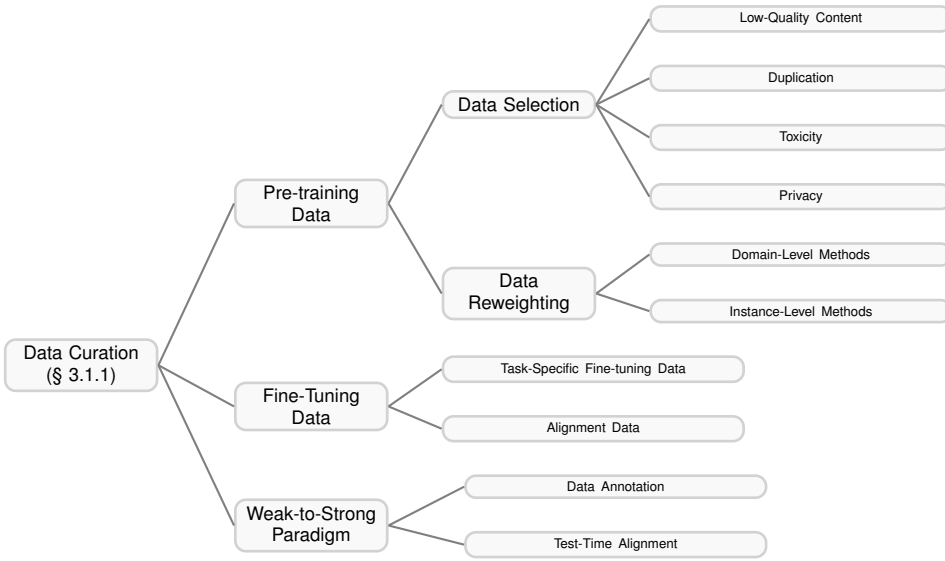


Figure 3: Taxonomy of data curation

reasoning performance. Finally, at the evaluation stage (see Section 3.1.4), SMs serve as lightweight verifiers to assess, diagnose, and repair LLM outputs, increasing reliability and robustness. From this life-cycle perspective, SMs are not simply smaller alternatives to LLMs, but useful tools that help improve LLMs at every stage of the pipeline.

3.1.1 Data Curation.

In the era of LLMs, a prevailing ideology is "more is better" — more data, more parameters, more GPUs. Indeed, the scaling laws reveal that model performance improves with both more parameters and tokens (Kaplan et al. 2020). However, as the data availability reaches its limits, a new paradigm is emerging: rather than simply using ever-larger datasets, the key lies in curating those texts with refined strategies. From filtering noisy web text during pre-training to selecting a small, high-quality set for fine-tuning, recent work demonstrates that less but more refined data enables more efficient and powerful training, and better alignment with human purpose (Albalak et al. 2024). This shift follows a weak-to-strong learning paradigm, in which rich but noisy signals give way to smaller, higher-quality supervision (Burns et al. 2024).

In the following, we present how to use small models to curate data from several aspects: pre-training data, instruction-tuning data, and the weak-to-strong paradigm, as shown in Figure 3.

Curating Pre-training Data. The reasoning capabilities of LLMs are largely attributed to their pre-training on extensive and diverse datasets, typically sourced from web scrapes, books, and scientific literature. Since expanding the quantity and diversity of these training datasets enhances the generalization ability of LLMs, significant efforts have been made to compile large-scale and diverse pre-training corpora, such as C4 (Raffel et al. 2020) and Pile (Gao et al. 2021). However, the idea of creating ever-larger datasets faces a significant challenge: data availability is finite, and there is a looming possibility

that public human text data could soon be exhausted (Villalobos et al. 2024). Moreover, not all data contributes equally to model performance; web-scraped content often includes noise and low-quality text. This has led to a paradigm shift from focusing purely on the quantity of data to prioritizing the quality of data. Recent research supports the notion that "less is more" (Marion et al. 2023), advocating for data selection or pruning techniques to curate high-quality subsets from large datasets, thereby enhancing model performance. We can divide these approaches into two categories, as shown in Figure 3: (1) Data Selection, and (2) Data Reweighting.

(1) Data Selection uses a small model to identify and curate a high-quality subset of the raw pre-training corpus. Earlier approaches largely relied on manual, rule-based heuristics such as blacklist filtering and MinHash deduplication to remove undesirable or duplicated text (Raffel et al. 2020; Tirumala et al. 2023; Penedo et al. 2023; Wenzek et al. 2020). Recognizing the limitations of hand-crafted rules, more recent methods use small proxy models to assess content quality, which enables the selection of high-quality subsets. For instance, a simple classifier can be trained to assess content quality, focusing on the removal of *noisy* (Brown et al. 2020; Du et al. 2022; Chowdhery et al. 2023), *toxic* (Arnett et al. 2024; Kamphuis 2024), and *private* (Subramani et al. 2023; Yu et al. 2024) data. For example, FineWeb-Edu (Penedo et al. 2024) trains a linear regression model on text embeddings to select high-quality data dedicated to the educational field. Moreover, a small language model may compute perplexity scores as a proxy for textual coherence and quality (Wenzek et al. 2020; Marion et al. 2023). Other recent methods introduce the *Importance Resampling* frameworks that select a subset of a large raw unlabeled dataset to match a desired target distribution (Xie et al. 2023b). Data selection reduces the dataset size but aims to raise the average usefulness of training tokens.

Another important role of small models in pre-training data curation is language identification (LID). When constructing multilingual corpora, accurately identifying the main language of each document is a critical step. Because this process must be applied to billions of documents, LID systems need to be both computationally efficient and inexpensive to run. Consequently, lightweight classifiers are commonly employed to filter and select texts for specific languages, which is particularly important for ensuring adequate coverage of low-resource languages (Kargaran et al. 2023; Burchell et al. 2023; NLLB Team 2024). Recent large-scale datasets such as FineWeb (Penedo et al. 2024, 2025) and HPLT (Oepen et al. 2025) rely on such LID models to curate high-quality multilingual resources for training LLMs.

(2) Data Reweighting goes beyond simple filtering by adjusting sampling probabilities. The key idea is the use of a *small proxy model* whose job is to estimate how much weight should be assigned to each data partition. We can divide reweighting into two key categories: (1) *Domain-Level Methods.* In this approach, a small proxy model is trained over different domains or sources, e.g., Wikipedia, books, web texts, to evaluate the performance or loss across each domain. For example, the framework DoReMi (Xie et al. 2023a) uses a 280 million parameter proxy model to compute domain weights for training an 8B parameter model, improving downstream accuracy by 6.5 points and reaching baseline accuracy 2.6× faster. Meanwhile, AutoScale (Kang et al. 2024) computes optimal domain compositions at small scale and then uses a predictor to extrapolate to large scale. Finally, small agents are introduced to learn how to reweight each domain via reinforcement learning (Yang et al. 2025). (2) *Instance-Level Methods.* Here the focus is on how to assign weights to individual examples (instances), and again

a small model often plays a key role in estimating how informative or representative these instances are. For example, the method PRESENCE (Thakkar et al. 2023) uses a proxy model to compute self-influence scores, which approximate how much giving a training example more weight during training would affect its own loss. Since these scores serve as a proxy for the degree to which an example is memorized by the model, they are subsequently used to guide instance-level data reweighting during pre-training. Small proxy models here serve as a cheap but predictive surrogate of how a large model would perform, and their assessments guide the weight assignments, which helps make data usage more efficient.

Curating Fine-Tuning Data. LLMs acquire substantial knowledge through pre-training, and fine-tuning is used to adapt these capabilities either to specific tasks or to align them with human preferences (Ouyang et al. 2022; Bai et al. 2022). In this light, it is useful to distinguish two types of data curation. (1) *Task-Specific Fine-Tuning Data.* This category of data is used for downstream tasks or specific domains such as classification and domain-specific QA. The curation focuses on selecting samples that are representative of the target distribution and avoid redundancy. A small language model can be leveraged as a data-selector. For example, we might use a smaller model to compute embedding similarity for each candidate data point, and then rank and select the most representative examples for fine-tuning while discarding the less informative ones (Liu et al. 2025). Another approach is to use the internal signals of a small model such as attention patterns to score data points, and then select the most informative subset for a specific task (Wang et al. 2025c). (2) *Alignment Data.* The goal is to steer the model toward desired instructions and align with human preferences such as honesty and safety. Here, data selection focuses more on using small models to efficiently collect high-quality instruction-response pairs rather than simply increasing quantity. Specifically, the study, *Less is More for Alignment*, demonstrates that fine-tuning on just 1,000 carefully curated instruction examples can yield a well-aligned model (Zhou et al. 2023a). This highlights the importance of selecting high-quality data for efficient instruction tuning (Longpre et al. 2023; Agarwal et al. 2025). Model-oriented data selection (MoDS) (Du, Zong, and Zhang 2023) is one approach that employs a small language model, DeBERTa (He et al. 2021), to evaluate instruction data based on quality, coverage, and necessity. Additionally, the LESS framework (Xia et al. 2024b) demonstrates that smaller models can be used to select influential data not only for larger models but also for models from different families. This underscores the potential of using targeted data selection techniques to optimize instruction tuning processes.

Weak-to-Strong Paradigm. LLMs are typically aligned with human values through reinforcement learning with human feedback (RLHF), where behaviors favored by humans are rewarded, and those rated poorly are penalized (Shen et al. 2023a). However, as LLMs continue to evolve and surpass human capabilities in various tasks, they are becoming *superhuman models*, capable of performing complex and creative tasks that may exceed human understanding. For instance, these models can generate thousands of lines of specialized code, engage in intricate mathematical reasoning, and produce lengthy, creative novels. Evaluating the correctness and safety of such outputs poses significant challenges for human evaluators. This scenario introduces a new paradigm for aligning superhuman models, termed *weak-to-strong generalization*, which involves using weaker (smaller) models as supervisors for stronger (larger) models (Burns et al. 2024). The goal is for strong models to learn from limited or imperfect signals and generalize beyond their weaker teachers' capabilities. We categorize recent advances

in weak-to-strong generalization into two primary classes: (1) Data Annotation, and (2) Test-Time Alignment.

(1) Data Annotation. This foundational approach fine-tunes a large model using labels or preferences generated by weaker models. Burns et al. (2024) formalize the paradigm and demonstrate that strong models can extrapolate from weak supervision. Guo and Yang (2024) introduce an approach that enhances weak-to-strong generalization by incorporating reliability estimation across multiple answers provided by weak models. This method improves the alignment process by filtering out uncertain data or adjusting the weight of reliable data. Beyond data labeling, weak models can also collaborate with large models during the inference phase to further enhance alignment. Instead of relying on a single weak teacher, recent methods employ ensembles of weak or specialized models to improve label quality and diversity. For example, Liu and Alahi (2024) suggest using a diverse set of specialized weak teachers, rather than relying on a single generalist model, to collectively supervise the strong student model. Lang, Huang, and Li (2025) introduce debate mechanisms between weaker models to refine signals before training a strong model. This weak-to-strong paradigm is not limited to language models but has also been extended to vision foundation models (Guo et al. 2024).

(2) Test-Time Alignment. Beyond training-time labeling, weak models can help strong models during inference to refine responses dynamically. Weak-to-Strong Search (Zhou et al. 2024) approaches the alignment of a large model as a test-time greedy search, aiming to maximize the log-likelihood difference between small tuned and untuned models, which function as a dense reward signal and a critic, respectively. Aligner (Ji et al. 2024) employs a small model to learn the correctional residuals between preferred and dispreferred responses, enabling direct application to various upstream LLMs for aligning with human preferences.

Summary

Given the limits of high-quality human-generated data, scaling language models can no longer rely solely on collecting more text. Instead, progress increasingly depends on curating existing data more effectively and embracing the principle that "less is more". In this context, SMs play a crucial role throughout both pre-training and fine-tuning. During pre-training, SMs support data selection by filtering low-quality, duplicated, toxic, or privacy-sensitive content, and by reweighting domains or instances to better shape the learning distribution. During fine-tuning, SMs can help construct task-specific and alignment datasets, improving annotation quality and reducing reliance on expensive human supervision.

As LLM capabilities continue to improve, human annotation becomes increasingly difficult. Stronger models may exceed human ability in certain reasoning tasks, which makes direct supervision less reliable. The Weak-to-Strong paradigm addresses this challenge by showing that weaker supervisors (including small models) can still guide stronger models effectively. By carefully designing supervision signals and alignment objectives, SMs can help extract knowledge from powerful models and contribute to building reliable reward models.

In practice, the benefits of using SMs for data curation are threefold. (1) Less is More. Small models can reduce dataset size while retaining or even improving quality for LLM training. (2) Cost Saving. When human annotation is costly and time-consuming,

smaller models provide a scalable and economical alternative. (3) More specialized Datasets. Small models can ensure high-relevance data, particularly for task- or domain-specific domains, such as low-resource multilingual, legal, and medical tasks.

Future Directions

(1) Humans can learn from relatively few examples, which contrasts with the massive data requirements of current LLM pre-training paradigms (Warstadt et al. 2023). It is therefore crucial to investigate methods that use less data to obtain better reasoning models, and to explore how small models in particular can play a key role in achieving this goal.

(2) While data curation offers clear advantages, LLMs still have a tendency to produce hallucinated and toxic content. Moreover, removing low-quality or toxic text could potentially degrade certain capabilities, such as generality (Longpre et al. 2024). Therefore, it is essential to define more nuanced criteria for evaluating data quality, including dimensions like factuality, safety, and diversity (Wettig et al. 2024; Liu et al. 2024c). Investigating the use of small models to develop effective and efficient data selection methods is a valuable area of study.

(3) Synthetic data serves as a valuable supplement to the limited amount of human-generated data (Long et al. 2024), yet the potential of small models in curating synthetic data remains largely unexplored.

(4) We tend to choose data that is clear and representative for alignment tuning, but including a subset of overly difficult examples can harm alignment performance (Gao et al. 2025). It is therefore crucial to dynamically estimate a model's current capacity and then tailor the selection so it contains an appropriate mix of easy and hard samples.

(5) While the weak-to-strong framework is effective in eliciting knowledge from stronger models, it is still far from recovering the full performance gap between weak and strong models. It is crucial to ensure that the strong model has a deep, intuitive understanding of the task at hand, is capable of correcting the weak model's errors, and naturally aligns with the objectives of the task (Burns et al. 2024). Future work should focus on identifying properties and methods that help achieve this goal.

(6) The current understanding of weak-to-strong generalization is limited. Researchers should develop a deep understanding of the underlying mechanisms that govern the success or failure of alignment methods, e.g. theoretical analysis (Lang, Sontag, and Vijayaraghavan 2024), errors in weak supervision (Guo and Yang 2024), and extrapolating generalization errors using scaling laws (Kaplan et al. 2020).

3.1.2 Augmented Reasoning.

Despite their impressive scale, LLMs rely on the internal knowledge encoded in their parameters during the pre-training stage. While this internalized knowledge enables strong generalization across many tasks, it is inherently limited. The model may lack up-to-date information (Kasai et al. 2023), domain-specific expertise (Feng et al. 2023), or robust multi-step reasoning ability (Mondorf and Plank 2024). As a result, when a task exceeds the model's internal knowledge capacity (Li et al. 2025b), LLMs can generate responses that appear plausible but are factually incorrect or hallucinated (Huang et al. 2025a).

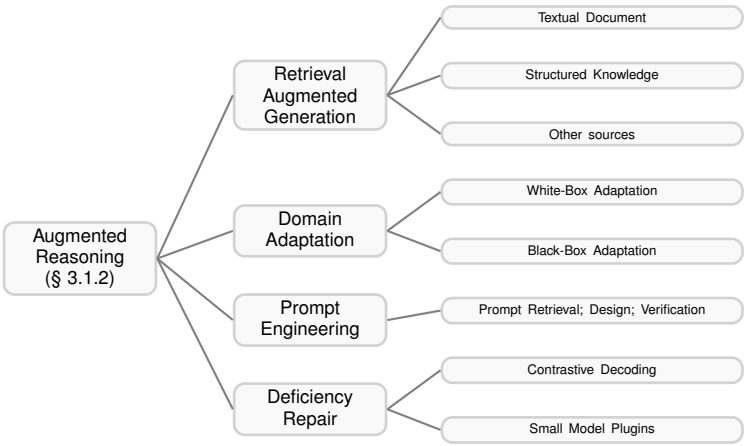


Figure 4: Taxonomy of augmented reasoning

From this perspective, many recent advances can be understood as efforts to bridge the gap between internal knowledge (what the model has memorized) and external knowledge (information and guidance introduced at run time). When the task falls beyond the model's internal capacity, it is necessary to augment reasoning by using external signals. In this section, we organize existing approaches into four categories: (1) Retrieval-Augmented Generation, which enriches internal knowledge with retrieved documents. (2) Domain Adaptation, which reshapes internal knowledge toward specialized distributions. (3) Prompt Engineering, which guides the model to better use its existing knowledge. (4) Deficiency Repair, which identifies and corrects systematic weaknesses. Together, these strategies illustrate different ways of balancing internal and external knowledge to improve reasoning reliability.

Retrieval Augmented Generation. LLMs exhibit impressive reasoning capabilities, yet their ability to memorize specific knowledge is somewhat limited. Consequently, LLMs may struggle with tasks that require domain-specific expertise or up-to-date information. To address these limitations, Retrieval-Augmented Generation (RAG) enhances LLMs by employing a lightweight retriever to find relevant document fragments from external knowledge bases, document collections, or other tools (Gao et al. 2023b; Lewis et al. 2020). By incorporating external knowledge, RAG effectively mitigates the issue of generating factually inaccurate content, often referred to as hallucinations (Shuster et al. 2021). RAG methods can be broadly categorized into three types based on the nature of the retrieval source.

(1) Textual Document is the most commonly used retrieval sources in RAG methods, encompassing resources such as Wikipedia (Trivedi et al. 2023; Asai et al. 2024), cross-lingual translations (Nie et al. 2023; Wang et al. 2025a) and domain-specific corpus (e.g. medical (Xiong et al. 2024) and legal (Yue et al. 2023) domains). These approaches generally employ lightweight retrieval models, such as sparse BM25 (Robertson, Zaragoza et al. 2009) and dense BERT-based (Izacard et al. 2022) retrievers, to extract relevant text from these sources.

(2) Structured Knowledge encompasses sources such as knowledge bases and databases, which are typically verified and can provide more precise information. For

example, KnowledgeGPT (Wang et al. 2023) enables LLMs to retrieve information from knowledge bases, while T-RAG (Pan et al. 2022) enhances answers by concatenating retrieved tables with the query. StructGPT (Jiang et al. 2023a) further augments generation by retrieving from hybrid sources, including knowledge bases, tables, and databases. The retriever in these methods can be a lightweight entity linker, query executor, or API.

(3) Other Sources include code, tools, and even images, which enable LLMs to leverage external information for enhanced reasoning. For instance, DocPrompting (Zhou et al. 2023b) employs a BM25 retriever to obtain relevant code documentation before code generation. Similarly, Toolformer (Schick et al. 2023) demonstrates that LMs can self-learn to use external tools, such as translators, calculators, and calendars, through simple APIs, leading to significant performance improvements.

Domain Adaptation. General-purpose LLMs still require further customization to achieve optimal performance in specific use cases (*e.g.* coding) and domains (*e.g.* medical tasks) (Feng et al. 2023). While fine-tuning on specialized data is one approach to adapting LLMs, this process has become increasingly resource-intensive, and in some cases, it is not feasible—especially when access to internal model parameters is restricted, as with models like ChatGPT (Ling et al. 2025). Recent research has explored adapting LLMs using smaller models, which can be categorized into two approaches: White-Box Adaptation and Black-Box Adaptation, depending on whether access to the model's internal states is available.

(1) White-Box Adaptation typically involves fine-tuning a small model to adjust the token distributions of frozen LLMs for a specific target domain. For instance, CombLM (Ormazabal, Artetxe, and Agirre 2023) learns a linear function to combine the probability distributions from the large black-box model with those from a smaller domain-specific expert model. IPA (Lu et al. 2023) introduces a lightweight adapter that tailors a large model toward desired objectives during decoding without requiring fine-tuning. IPA achieves this by optimizing the combined distribution using reinforcement learning. Proxy-tuning (Liu et al. 2024a) fine-tunes a smaller language model, contrasting the probabilities between the tuned model (the expert) and its untuned version (the anti-expert) to guide the larger base model. These approaches only modify the parameters of small domain-specific experts, allowing LLMs to be adapted to specific domain tasks. However, white-box adaptation is not applicable to API-only modeling services, where access to internal model parameters is restricted.

(2) Black-Box Adaptation involves using a small domain-specific model to guide LLMs toward a target domain by providing textual relevant knowledge. Retrieval Augmented Generation (RAG) can extract query-relevant knowledge from an external document collection or knowledge base, and thus enhance general LLMs by leveraging their in-context learning ability. It involves first using a lightweight retriever to find relevant content from the domain corpus, which is then incorporated into the LLM's input to improve its understanding of domain-specific knowledge (Siriwardhana et al. 2023; Shi et al. 2024a; Gao et al. 2023b). Another approach employs small expert models to generate background knowledge for the base LLM. For example, BLADE (Li et al. 2025a) and Knowledge Card (Feng et al. 2024a) first pre-train a small expert model on domain-specific data, which then generates expertise knowledge in response to a query, thereby enhancing the base LLM's performance. MedAdapter (Shi et al. 2024b) fine-tunes a small BERT-sized adapter to rank candidate solutions generated by LLMs.

Prompt Engineering. Prompt-based learning is a prevalent paradigm in LLMs where prompts are crafted to facilitate few-shot or even zero-shot learning, enabling adaptation to new scenarios with minimal or no labeled data (Liu et al. 2023). This approach leverages the power of In-Context Learning (ICL) (Dong et al. 2024), which operates without performing parameter updates. Instead, it relies on a prompt context that includes a few demonstration examples structured within natural language templates.

In this learning process, small models can be employed to enhance prompts, thereby improving the performance of larger models. For instance, Uprise (Cheng et al. 2023) optimizes a lightweight retriever that autonomously retrieves prompts for zero-shot tasks, thereby minimizing the manual effort required for prompt engineering. Similarly, DaSLaM (Juneja et al. 2023) uses a small model to break down complex problems into subproblems that necessitate fewer reasoning steps, leading to performance improvements in larger models across multiple reasoning datasets.

Beyond prompt retrieval and decomposition, small models can also enhance in-context learning (ICL), a widely used prompt engineering technique, by improving the quality of demonstrations included in prompts. For example, some methods fine-tune small models to generate pseudo labels for input examples (Xu et al. 2024a; Lee et al. 2024), which produce higher-quality demonstrations that lead to better ICL performance than the original prompts. Furthermore, in multi-stage prompting pipelines, small models can verify (Hsu et al. 2024) or rewrite (Vernikos et al. 2024) the outputs of LLMs, further improving response quality without requiring LLM fine-tuning.

Deficiency Repair. Powerful LLMs may generate repeated, untruthful, and toxic content, and small models can be used to repair these defects. We introduce two ways to achieve this goal: contrastive decoding and small model plug-ins.

(1) Contrastive Decoding exploits the contrasts between a larger model (expert) and a smaller model (amateur) by choosing tokens that maximize their log-likelihood difference. Existing work has explored the synergistic use of logits from both LLMs and SMs to reduce repeated text (Li et al. 2023b), mitigate hallucinations (Sennrich, Vamvas, and Mohammadshahi 2024), augment reasoning capabilities (O'Brien and Lewis 2023) and safeguard user privacy (Zhang et al. 2024a). Since fine-tuning LLMs is computing-intensive, proxy tuning proposes fine-tuning a small model and contrasting the difference between the original LLMs and small models to adapt to the target task (Liu et al. 2024a).

(2) Small Model Plugins fine-tune a specialized small model to address some of the shortcomings of the larger model. For example, the performance of LLMs may degrade when encountering out-of-vocabulary (OOV) words. To address this issue, we can train a small model to mimic the behavior of the large model and impute representations for unseen words (Pinter, Guthrie, and Eisenstein 2017; Chen, Varoquaux, and Suchanek 2022). In this way, we can make large models robust with little cost. Additionally, LLMs may generate hallucinated texts and we can train a small model to detect hallucinations (Cheng et al. 2024) or calibrate confidence scores (Chen et al. 2024c).

Summary

LLMs store parametric knowledge, i.e., facts and patterns learned in their weights during pre-training, but this kind of knowledge is static and limited in depth for domain-

specific tasks, leading to plausible yet hallucinated outputs. When a task exceeds what an LLM can reliably know or reason about on its own, it becomes important to integrate external knowledge and auxiliary reasoning mechanisms. In practice, four strategies using smaller models to augment LLM reasoning have been introduced in this section: Retrieval-Augmented Generation (RAG), domain adaptation, prompt engineering, and deficiency repair.

In real-world settings, practitioners can choose among them based on task requirements and the nature of the external knowledge involved.

- When the task demands up-to-date or domain-specific facts, RAG is often the primary and low-cost strategy. Retrieving relevant information at inference time and prioritizing grounded sources rather than static parametric knowledge. This approach reduces hallucinations and allows real-time integration of new information without retraining the LLM.

- When a task requires adapting an LLM to a specialized domain but full fine-tuning is expensive or impractical, domain adaptation via small models offers a cost-effective alternative. Domain-expert models can capture targeted, high-precision knowledge and provide structured guidance to the general-purpose LLM. Rather than modifying the large model's parameters, these small experts inject domain-specific signals at the inference stage.

- When the input pattern is not aligned with the distribution of parametric knowledge, prompt engineering can be introduced to guide the LLM's reasoning process. Optimized prompts can surface relevant external knowledge more effectively and help steer the model toward correct inference patterns without high computational cost.

- When outputs require specific properties, such as safety, interpretability, or uncertainty expression, deficiency repair mechanisms can be employed to address these limitations. Here, small auxiliary models evaluate or adjust the LLM's outputs, effectively correcting or flagging problematic content.

Future Directions

(1) The retrieval augmented text generation performance is very sensitive to the retrieval quality (Yoran et al. 2024). However, the widely-used dense-retrieval systems, i.e., a small embedding model with a similarity calculation, have theoretical limitations. In particular, their ability to represent different combinations of relevant documents can be constrained by the embedding dimension (Weller et al. 2025). Therefore, more expressive and efficient retrieval approaches may be needed for advanced retrieval settings.

(2) LLMs are still good at memorizing patterns rather than true logical reasoning. Recent findings indicate that LLMs struggle with classic reasoning problems and fail to generalize beyond the patterns seen in their training data, such as deductive reasoning (Seals and Shalin 2024) and compositional reasoning (Sinha, Premsri, and Kordjamshidi 2024). It is important to explore how to use small models to enhance the logical reasoning

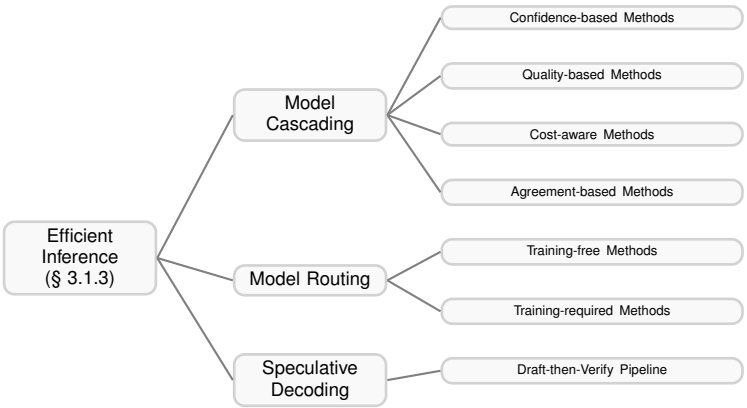


Figure 5: Taxonomy of efficient inference

capability of LLMs.

(3) Current domain adaptation methods often necessitate pre-training or fine-tuning a domain-specific expert, which is impractical for resource-constrained tasks. Exploring efficient methods to transfer and adapt models across many domains in low-resource settings is a valuable area of research (Sun et al. 2024).

3.1.3 Efficient Inference.

LLMs have demonstrated remarkable capabilities, but their gains in performance come with substantial costs. Increasing model size typically entails slower inference, higher API costs, and significant environmental and energy burdens (Wu et al. 2022). As model deployment scales, these costs become not only financial concerns but also sustainability challenges.

In contrast, smaller models offer clear efficiency advantages: they are faster, cheaper to run, and more energy-efficient. While they may not match the performance of their larger counterparts, many real-world queries do not require the full capacity of a large model. User requests span a broad spectrum of difficulties, which suggests that uniform reliance on large models is often unnecessary.

This observation motivates a core idea in efficient inference. Allocating computational resources adaptively rather than uniformly. By orchestrating a list of models with different capacities, systems can use expensive large models for genuinely hard queries while allowing smaller models to handle simpler ones. Such adaptive allocation preserves performance where needed while substantially reducing overall cost. Broadly, existing approaches to this paradigm fall into three categories: model cascading, model routing, and speculative decoding, as illustrated in Figure 5.

Model Cascading. This process involves the sequential use of multiple models to make predictions or decisions, where each model in the cascade has a different level of complexity. The output of one model may trigger the activation of the next model in the sequence. This approach allows for the collaboration of models of varying sizes, enabling smaller models to handle simpler input queries while transferring more complex tasks to larger models. The critical step in this process is determining whether a given

model is capable of addressing the input question. This method effectively optimizes inference speed and reduces financial costs.

(1) Confidence-based Methods. The smaller model produces a confidence score or probability and, if that score falls below a threshold, the query is forwarded. For instance, this work, query-level uncertainty (Chen et al. 2026), proposes a training-free method to estimate whether a model is capable of addressing a given query without generating any answers, which can achieve effective model cascading while reducing inference costs. Confidence-based methods aim to assess the knowledge gap, and then allocate a task to the model with an appropriate size (Chen et al. 2024a; Enomoto and Eda 2021).

(2) Quality-based Methods. The system evaluates the quality of the output by using a verifier model and triggers escalation if the output quality falls short. For example, some existing techniques train a small evaluator to assess the output correctness (Kag et al. 2023; Chen, Zaharia, and Zou 2020, 2024), thereby deciding whether to escalate the query to a more complex model.

(3) Cost-aware Methods. Beyond simple correctness, the trigger mechanism may incorporate cost-aware objectives such as latency or monetary costs (Wang et al. 2018). For example, we can study various models with different sizes, comparing their accuracy, financial cost, and latency, and then select which model to use based on budget or latency limitation (Wang et al. 2024a; Zhang et al. 2024c).

(4) Agreement-based Methods. Given that LLMs can perform self-verification (Dhuliawala et al. 2024) and provide confidence levels in their responses (Tian et al. 2023), we can use the consistency between multiple answers as a trigger. If the outputs disagree or diverge, the task is deferred to a larger model. AutoMix (Aggarwal et al. 2024) employs verification prompts to query the model multiple times, using the consistency of these responses as an estimated confidence score. The framework then determines whether the current model's output should be accepted or if the query should be forwarded to other models for enhanced performance.

Model Routing. This process optimizes the deployment of multiple models of varying sizes by dynamically directing input data to the most appropriate models, thereby enhancing both efficiency and effectiveness in practical applications. The core component of this approach is the development of a router that assigns input to one or more suitable models within the pool. Existing approaches can be divided into two main categories: (1) Training-free Router and (2) Training-required Router.

(1) Training-free Router. These methods do not require a specific supervised training step for a routing model. Instead, they rely on heuristics such as similarity ranking, uncertainty or other lightweight signals. For example, the router can be guided by estimated uncertainty of predictions, which allows the system to dynamically select between larger and smaller models (Su et al. 2026). Or we can use the agreement (or similarity) among outputs from different models to select the best model for each input. The intuition is that if a model's output is very different compared to the other models' outputs for the same input, then that model's output is more likely to be of lower quality and unreliable (Guha et al. 2024). We can also use semantic tags, i.e., high-level semantic descriptions of a query (e.g., task type, domain, or intent), to route requests to appropriate models (Chen et al. 2025). These tags are predicted by a lightweight tagger and enable effective routing for open-domain text generation tasks. In this setting, the

router avoids the cost of data annotation and supervised router training, which makes it easy to adapt to new domains. However, the inference efficiency depends on the routing strategy. While routers based on lightweight signals such as uncertainty incur only a small overhead, methods requiring multiple model outputs may increase inference cost.

(2) Training-required Router. These methods train a dedicated routing model, such as a classifier or ranking model, to learn how to allocate user tasks to the appropriate size models. A straightforward approach is to consider the input-output pairs from all models and select the best-performing one (Jiang, Ren, and Lin 2023; Ding et al. 2024b). However, this comprehensive ensemble strategy does not significantly reduce inference costs. To address this, some methods train efficient, reward-based routers that select optimal models without needing to access the models' outputs (Lu et al. 2024; Narayanan Hari and Thomson 2023). OrchestraLLM (Lee, Cheng, and Ostendorf 2024) introduces a retrieval-based dynamic router that assumes instances with similar semantic embeddings share the same difficulty level. This allows for the selection of an appropriate expert based on the embedding distances between the testing instance and those in expert pools. Similarly, RouteLLM (Ong et al. 2025) formulates routing as a preference prediction problem by training a lightweight router on human preference data and augmented examples to predict whether a query requires a stronger LLM or can be handled by a weaker one. Based on the predicted user preference, the router dynamically selects the appropriate model, which can reduce inference cost while maintaining response quality and generalizing well to out-of-domain queries. FORC (Šakota, Peyrard, and West 2024) proposes a meta-model (a regression model) to assign queries to the most suitable model without requiring the execution of any large models during the process. The meta-model is trained on existing pairs of queries and model performance scores. In this category, the router has to be maintained, but can often yield stronger selection performance.

Furthermore, recent benchmarks for model routing have been established (Hu et al. 2024b; Shnitzer et al. 2024), facilitating the collaboration between larger and smaller models.

Speculative Decoding. This technique aims to speed up the decoding process of a generative model, which often involves using a smaller, faster auxiliary model alongside the main, larger model, which consists of a two-phase "draft-then-verify" pipeline. A smaller, faster draft model first generates multiple candidate tokens, which are then verified by the larger target model in a single parallel forward pass rather than being generated one by one (Leviathan, Kalman, and Matias 2023; Chen et al. 2023a; Xia et al. 2024a). Consequently, when most drafted tokens are accepted, the number of sequential decoding steps required by the target model is substantially reduced, which leads to lower inference latency while preserving the original output distribution.

Summary

From a practitioner's perspective, the choice among model cascading, routing, and speculative decoding depends on deployment goals. **Model cascading** is most suitable when query difficulty varies widely, and cost reduction is the primary objective. A small model handles easy cases, while harder queries are deferred to larger models. This step can ensure robustness with minimal engineering efforts. **Model routing** is preferable when the task involves multiple types (e.g., coding and summarization) or

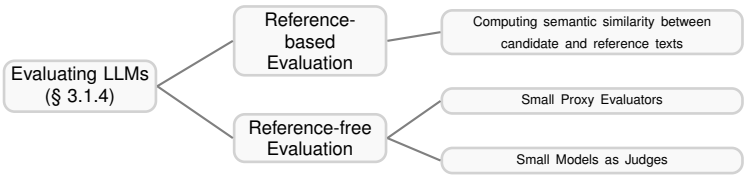


Figure 6: Taxonomy of evaluating LLMs

domains (e.g., law and healthcare), as a routing mechanism can directly assign each query to the most appropriate model in a single pass. In contrast, speculative decoding is particularly suitable when a large model is required for high-quality generation. A smaller model drafts multiple candidate tokens that are verified by the large model in parallel, which is able to reduce sequential decoding steps and inference latency without sacrificing output quality.

Future Directions

(1) Existing model ensembling methods typically rely on a limited, pre-defined list of models, yet the real world encompasses open-domain and constantly evolving LLMs, such as those available on HuggingFace. Exploring ways to leverage these extensive model libraries to create intelligent and efficient systems holds significant promise (Shen et al. 2023b).

(2) In some existing approaches to speculative decoding, the draft model is required to share the vocabulary with the main model, which typically limits speculative decoding to models from the same family (for example, different sizes of GPT). However, it would be beneficial to explore collaborations between models from diverse sources and vocabularies, thereby expanding the set of viable auxiliary models and enabling more flexible, cost-efficient inference (Remy et al. 2024).

3.1.4 Evaluating LLMs. Evaluating open-ended text generation remains one of the significant challenges in deploying LLMs (Chang et al. 2024). Unlike classification tasks with clear ground-truth labels, generative outputs can be valid in many different forms, making quality assessment inherently subjective and challenging. Traditional metrics such as BLEU (Papineni et al. 2002) and ROUGE (Lin 2004) rely on surface-level overlap with reference texts, but lexical similarity alone often fails to capture semantic relatedness and reasoning correctness (Liu et al. 2016). As models become more capable and outputs more diverse, these limitations become increasingly pronounced.
This motivates a shift from string matching to model-based evaluation, where smaller models are used as automated judges to approximate human assessment. Rather than measuring n-gram overlap, these evaluators aim to assess semantic similarity, coherence, factual consistency, or overall quality. Broadly, such approaches fall into two categories: (1) *reference-based methods*, which compare outputs against ground-truth texts using learned semantic metrics, and (2) *reference-free methods*, which directly judge quality without requiring explicit references, as illustrated in Figure 6. These paradigms reflect a broader transition toward scalable evaluation frameworks for open-ended generation.

(1) Reference-based Evaluation approaches rely on comparing outputs to human-written references. For instance, BERTSCORE (Zhang et al. 2020) uses a BERT encoder to compute semantic similarity between candidate and reference texts, and BARTScore (Yuan, Neubig, and Liu 2021) leverages the BART encoder-decoder model to evaluate aspects like informativeness, fluency, and factuality. Although useful when references are available, these methods may struggle in highly open-ended tasks due to limited reference coverage.

(2) Reference-free Evaluation approaches evaluate model outputs without relying on human-written references. This includes using a small proxy evaluator to estimate how well the generation would perform, such as fluency, relevance, coherence, and factuality. For example, we can apply small natural language inference (NLI) models to estimate the uncertainty of LLM responses (Manakul, Liusie, and Gales 2023; Kuhn, Gal, and Farquhar 2023), or employ proxy models to predict LLM performance (Anugraha et al. 2025), which substantially reduces the computational costs associated with fine-tuning and inference during model selection. Recent advances include frameworks that treat an LLM as a judge (Gu et al. 2024a), which introduce another judge LM to evaluate the quality of generated texts. Existing findings indicate that smaller judges can provide reasonable signals for ranking tasks even if their absolute alignment was weaker (Thakur et al. 2025).

Summary

From a deployment perspective, the choice between reference-based and reference-free evaluation depends mainly on the availability of high-quality references. Reference-based methods are most suitable when reliable human-written references exist, and outputs are relatively constrained (e.g., translation, summarization). In such settings, semantic metrics like BERTScore or BARTScore provide stable and reproducible comparisons at low cost. However, when tasks are highly open-ended (e.g., creative writing and reasoning) or reference coverage is sparse, reference-based metrics may not be applicable. In these cases, reference-free methods become more appropriate, as they directly assess fluency, coherence, factual consistency, or reasoning quality without relying on gold texts. Small proxy models or NLI-based evaluators are particularly useful when scalability and cost are priorities. Meanwhile, using an LLM as a judge is preferable when high-fidelity evaluation is required, and computational cost is less restrictive, for example in reasoning-required tasks.

Future Directions

(1) As large models advance, they increasingly generate lengthy and complex texts, such as specialized code and scientific papers, which are challenging for humans to evaluate. Consequently, it is essential to develop efficient evaluators to assess various aspects of the generated content, such as factuality (Min et al. 2023), safety (Zhang et al. 2024d), and uncertainty (Huang et al. 2025b).

(2) Small evaluators may not be robust, and may be sensitive to the order of answers in pairwise comparisons (Li et al. 2024). This judge can also introduce bias that undermines the reliability and trustworthiness of the evaluation (Ye et al. 2025). There remains significant room for using small models as judges.

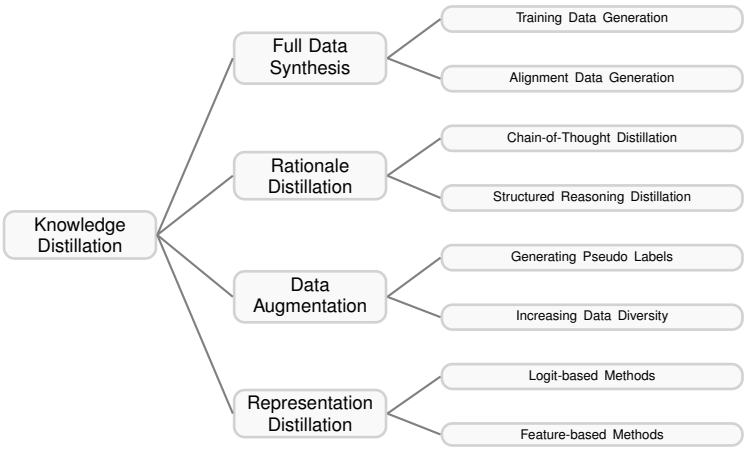


Figure 7: Taxonomy of knowledge distillation

3.2 LLMs Enhance Small Models

LLMs can enhance small models mainly through *Knowledge Distillation* (Hinton, Vinyals, and Dean 2015; Gou et al. 2021; Zhu et al. 2024b; Xu et al. 2024b), where knowledge learned by a large, powerful teacher model is transferred to a smaller, more efficient student model. Rather than directly increasing model size or training data, this paradigm uses LLMs to provide richer supervision and guidance during training, which enables small models to achieve stronger performance under limited capacity and computational budgets. In practice, knowledge distillation from LLMs can be performed at multiple levels of granularity. First, LLMs can synthesize full training data, including both inputs and labels (Section 3.2.1). Second, they can provide explicit rationales, i.e., the reasoning steps, that guide small models toward more structured and interpretable reasoning (Section 3.2.2). Third, LLMs can support data augmentation by rephrasing and diversifying existing datasets (Section 3.2.3). Finally, distillation can operate on the teacher model's internal representations, where internal features or intermediate states are used to shape the learned representations of small models (Section 3.2.4). The first three approaches correspond to black-box distillation, as they depend only on observable outputs of the teacher model, whereas representation distillation is a white-box method that requires access to internal model states. Together, these approaches transfer the knowledge from LLMs to small models, improving their reasoning ability without increasing model size.

3.2.1 Full Data Synthesis. Human-created data is expensive and finite, and there is a concern that publicly available human text may soon be depleted (Villalobos et al. 2024). In response, using LLMs to generate full training data to learn a task-specific small model is both efficient and practical.

(1) Training Data Generation. The idea of this category emphasizes generating both *inputs* and *labels* from scratch (Nadăș, Dioșan, and Tomescu 2025; Chung, Kamar, and Amershi 2023). For example, Ye et al. (2022) propose generating a synthetic dataset from an LLM in an unsupervised zero-shot fashion, and then training a much smaller

downstream model on this dataset to achieve efficient inference. Meng et al. (2022) use a larger model to generate class-conditioned synthetic texts for zero-shot language understanding tasks. Subsequent studies have extended this method to various tasks, including text classification (Li et al. 2023c), clinical text mining (Tang et al. 2023), information extraction (Josifoski et al. 2023), and hate speech detection (Hartvigsen et al. 2022).

(2) Alignment Data Generation. Beyond task-oriented synthetic generation, another line of recent work focuses on alignment data generation, where LLMs synthesize instruction–response pairs to fine-tune the target model, aiming to align it with human-preferred properties. Examples include CodecLM, which adaptively generates high-quality synthetic instruction data tailored to specific instruction distributions (Wang et al. 2024c), and Bonito, which converts unannotated text into instruction tuning datasets to improve zero-shot adaptation of LLMs (Nayak et al. 2024). Rather than merely scaling up instruction datasets, these alignment synthesis methods highlight the importance of data quality and distributional matching.

3.2.2 Rationale Distillation. This distillation focuses on transferring reasoning rationales from teacher to student, going beyond simple final-answer supervision.

(1) Chain-of-Thought (CoT) Distillation. This approach uses explicit step-by-step reasoning steps generated by large language models to provide richer supervision for smaller models. In CoT distillation, the student model is trained not only on task inputs and outputs but also on multi-step textual rationales from the teacher, which has been shown to enhance reasoning capabilities on complex tasks (Li et al. 2022; Hsieh et al. 2023; Shridhar, Stolfo, and Sachan 2023; Magister et al. 2023; Li et al. 2023a; Fu et al. 2023; Tian et al. 2024). Recent work has also explored alternative textual reasoning transfer. For example, adversarial distillation frameworks such as Lion (Jiang et al. 2023b) iteratively generate and focus on hard instructions to improve student performance via feedback from the teacher model. Counterfactual distillation methods leverage counterfactual examples to make SMs more robust to out-of-distribution (OOD) data (Feng et al. 2024b).

(2) Structured Reasoning Distillation. Apart from natural-language CoT distillation, there is a growing line of work that moves beyond surface textual chains to distill reasoning in more structured forms. We refer to this as *Structured Reasoning Distillation*, where reasoning knowledge is captured in structured signals and discrete logic. For instance, Program-aided Distillation (Zhu et al. 2024a) leverages synthetic reasoning programs generated by LLMs for fine-tuning smaller models, which shows promising performance on arithmetic reasoning, symbolic reasoning, and general ability. Reasoning Scaffolding (Wen et al. 2025) distills the structured, algorithmic flow of a teacher's reasoning, moving beyond surface-level text imitation, which leads to more accurate and logically robust reasoning than standard CoT distillation.

3.2.3 Data Augmentation. In this context, data augmentation refers to the use of LLMs to modify existing data points to increase data diversity, which can then be directly used to train smaller models (Ding et al. 2024a; Chen et al. 2023b). The use of LLMs for dataset augmentation has become a powerful strategy in NLP, offering a way to enrich training samples when labeled data is scarce.

(1) Generating Pseudo Labels. One common form of augmentation uses LLMs to assign labels to existing unlabeled or weakly labeled inputs, which can produce pseudo-

labeled data that supplements supervised learning (Wang et al. 2021; Gao et al. 2023a). This parallels knowledge-distillation via data labeling, where teacher models bootstrap learning by providing labels for otherwise unlabeled inputs.

(2) Increasing Data Diversity. Another major category involves transforming or rewriting existing data to create variations that preserve semantic content while improving dataset diversity. LLMs can paraphrase or rewrite texts to generate additional training samples, which has been shown to improve model robustness and generalization (Mi, Xie, and Zhang 2022; Witteveen and Andrews 2019). This technique has been applied in tasks such as information retrieval, where LLMs rewrite queries to better match target documents (Ma et al. 2023a). More broadly, LLM-based rewriting has been used to augment datasets for text classification (Ding et al. 2024a). Furthermore, data augmentation can be applied to various tasks such as personality detection (Hu et al. 2024a), intent classification (Sahu et al. 2022), and dialogue understanding (Chen et al. 2022). Fine-tuning smaller models with these augmented samples can significantly enhance their efficacy and robustness.

3.2.4 Representation Distillation. This approach involves using internal states of the teacher model, which provides transparency in the training process of the student model. This approach leverages the teacher's output distributions and internal features such as hidden representations and attention maps to guide the student's learning process.

(1) Logit-based Methods. The student imitates the teacher's final output distributions, e.g., soft labels or logits. This is the most direct form of knowledge transfer. For example, Hinton, Vinyals, and Dean (2015) propose using the teacher's softened softmax outputs as targets for the student. More recently, Gu et al. (2024b) in MiniLLM adopt the reverse KL divergence between student and teacher distributions as the training objective. Similarly, Yang et al. (2023) propose a unified approach to both teacher-student and self-distillation by using normalized loss functions and customized soft labels, thereby refining how soft targets are used in distillation.

(2) Feature-based Methods. Beyond just outputs, the student learns to imitate the internal representations of the teacher model, such as hidden states, attention maps, or layer activations. This process provides richer information about the teacher model's internal representations than using its output predictions alone. For example, DistilBERT (Sanh et al. 2019) and ZeroQuant (Yao et al. 2022) include a distance loss between the hidden states of teacher and student in the training objective. Liang et al. (2023) propose a task-aware layer-wise distillation method, where the student aligns with the teacher's hidden representations through a task-specific filtering to select only the informative knowledge, thereby reducing redundancy. Additionally, Liu et al. (2024d) use synthetic generations from the teacher as pseudo-data to perform feature-based distillation for a fully quantized student model. This is achieved by aligning the teacher's intermediate representations, including hidden activations and KV-cache, thereby preserving the student's internal representations after quantization.

Summary

Knowledge distillation facilitates the transfer of knowledge from a larger model to a smaller one, which enables the development of more cost-effective and efficient models. In practice, the choice of distillation strategy depends on whether the goal is data expansion, reasoning transfer, or model compression. Full data synthesis is useful

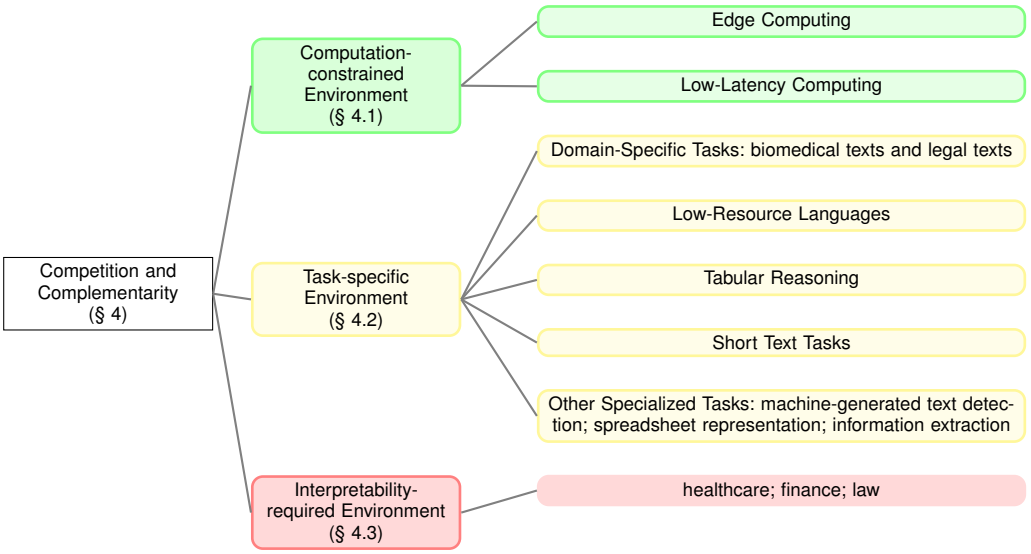


Figure 8: Competition and complementarity between SMs and LLMs

when human-labeled data is scarce, and scaling supervision is the priority. Rationale distillation emphasizes transferring reasoning capabilities from a large model to a smaller one, such as improving multi-step reasoning or explainable decision-making. Data augmentation methods aim to improve robustness and diversity when a baseline dataset already exists but lacks coverage or diversity. Finally, representation distillation is ideal when the goal is model compression and the internal state of the teacher model is accessible.

Future Directions

(1) Currently, closed-source LLMs remain more powerful than their open-source counterparts. However, using closed-source models for data synthesis may raise privacy and security concerns, particularly in sensitive contexts such as medical scenarios involving patient data (Ollion et al. 2023). Addressing how to protect data privacy during this process is a critical area of concern.

(2) Generating training data with large-scale models is costly, making it essential to explore methods for reducing the expense while still producing high-quality data. For instance, recent research suggests that smaller, less powerful models can sometimes generate better training data points (Bansal et al. 2025).

(3) Although data distillation and synthetic data generation have shown promising results, they may also introduce noisy, hallucinated content, and distribution bias (Liu et al. 2024b; Chen et al. 2024b). Developing effective data quality assessment, filtering, and verification methods to identify reliable synthetic data before model training remains an important research direction (Long et al. 2024).

4. Competition and Complementarity

In this section, we examine the relationship between small models and large models by emphasizing the competition and complementarity. Small models offer distinct advantages in cost-efficiency and interpretability (small models or shallow networks tend to be more interpretable than their large counterparts) (Gilpin et al. 2018; Barceló et al. 2020). At the same time, large models can boost small ones through knowledge distillation and data-generation (Gou et al. 2021; Nadăș, Dioșan, and Tomescu 2025). We argue that rather than replacing one with the other, the optimal ecosystem is hybrid. Small models serving specialized or cost-effective roles, with large models built to support, guide, and augment them. Below, we outline three scenarios—computation-constrained environments (Section 4.1), task-specific environments (Section 4.2), and interpretability-critical environments (Section 4.3), as shown in Figure 8. Small models are not in opposition to large models but form an indispensable and complementary part of the particular deployment settings.

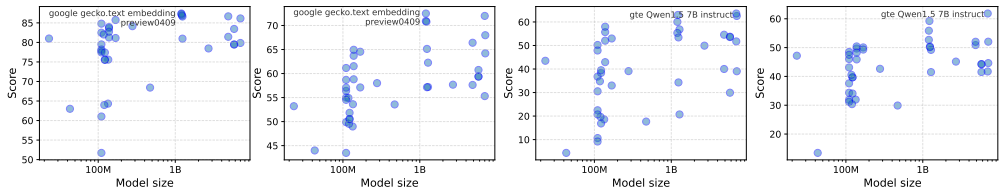
It is important to note that when we refer to small models in this section, we are not limiting the concept to Transformer-only architectures. The notion of small models may naturally extend to other architectures, such as shallow neural networks or even statistical models. This inclusive definition enables our discussion to remain relevant even as architectures evolve and model sizes continue to rise in the future. If tomorrow a new class of model emerges, what we classify as "small" and "large" will simply shift in relative terms, but the principles we adopt in this work remain applicable.

4.1 Computation-constrained Environment

As LLMs become more capable, their deployments come with substantial computational demands, which is impractical in many operational settings, e.g., edge computing (Wan et al. 2023; Dhar et al. 2024) and low-latency computing. By contrast, smaller models offer an alternative in environments where latency or hardware are constrained. In this section, we focus on two key categories: (1) Edge Computing and (2) Low-Latency Computing.

Edge Computing. In edge devices such as mobile phones and IoT devices, memory, power and connectivity are severely limited (Dhar et al. 2024). Large models are often infeasible for these resource-constrained conditions. Existing surveys have evaluated small language models, such as Phi-3-mini (3.8B) (Abdin et al. 2024), MiniCPM (Hu et al. 2024c), and Gemma-2B (Gemma Team et al. 2024), on performance and running time for edge settings (Lu et al. 2025; Jang and Morabito 2025), which offers a comprehensive view of SMs across hardware and systems. These studies advocate that smaller models are viable approaches for resource-constrained deployment scenarios like smartphones and Web-of-Things devices.

Low-Latency Computing. In low-latency inference scenarios, such as search engines, document retrieval, or recommendation systems, the ability to process many requests quickly and return responses with minimal delay is very important. Many tasks, how-



(a) Cosine Similarity on `Text Similarity` (b) Accuracy on `Text Classification` (c) NDCG@10 on `Information Retrieval` (d) v_measure on `Text Clustering`

Figure 9: Performance of representative text embedding models, including both encoder-only and decoder-only architectures, on four MTEB task categories. Results are averaged over five datasets for each task.

ever, are not knowledge-intensive and do not demand complex reasoning that can be effectively handled by smaller models.

To provide an empirical comparison between model scale and task performance, we evaluate a collection of representative text embedding models, including both encoder-only and decoder-only architectures, with different parameter sizes on the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al. 2023). We select four representative tasks, namely text similarity, text classification, information retrieval, and text clustering, and report the average performance over five datasets for each task. MTEB was chosen because it is a widely adopted benchmark covering diverse text understanding and retrieval tasks, which makes it suitable for analyzing how model scaling affects performance under practical deployment scenarios.

For instance, Figure 9 illustrates the relationship between performance and model size across four tasks in MTEB (Muennighoff et al. 2023), where we observe diminishing returns from increasing model sizes, particularly in tasks like text similarity and classification. In the case of information retrieval (Figure 9c), which involves computing similarity between a query and a document collection, faster inference speed is critical. Under these conditions, the lightweight encoder-based embedding models remain widely used in the IR task (Reimers and Gurevych 2019; Samarinas and Zamani 2025; Yu et al. 2025).

Small models are increasingly valuable in scenarios where computational resources are limited. Techniques such as knowledge distillation (Xu et al. 2024b) allow the transfer of knowledge from LLMs to smaller models, enabling these smaller models to achieve similar performance while significantly reducing model size. Consequently, smaller models are strategic components for edge deployment and low-latency computing.

4.2 Task-specific Environment

Training LLMs requires trillions of tokens (Raffel et al. 2020; Kaplan et al. 2020; Gao et al. 2021), but sufficient data is often unavailable for certain specialized domains (e.g., biomedical text) or tasks (e.g., tabular reasoning). In such cases, pre-training a large foundational model is not feasible, and small models can offer promising returns in

this case. We outline several task-specific scenarios where small models can deliver comparable results.

Domain-Specific Tasks. Domains such as biomedical or legal fields often have fewer training tokens available. Recent studies have shown that fine-tuning small models on domain-specific datasets can outperform general LLMs on various biomedical (Lehman et al. 2023; Bucher and Martini 2024) and legal (Chalkidis 2023) tasks.

Low-Resource Languages. Low-resource languages often lack sufficient annotated data to effectively train large, powerful models. In such settings, smaller multilingual models, such as mBERT (Devlin et al. 2019), offer a more practical and promising alternative (Gurgurov et al. 2025). In low-resource language translation, specialized small models are commonly used to synthesize training data by generating large-scale parallel corpora from monolingual text (Sennrich, Haddow, and Birch 2016). This approach enables the creation of sufficient pretraining data for low-resource languages, while small encoder–decoder models can outperform general-purpose LLMs in these settings (Doshi, Dabre, and Bhattacharyya 2024; Wang et al. 2025b).

Tabular Reasoning. Tabular datasets are typically smaller than benchmarks in other domains, such as text or image data, and are highly structured, consisting of heterogeneous data types (e.g., numerical, categorical, ordinal). Due to these characteristics, small tree-based models can achieve competitive performance compared to large deep-learning models for tabular data (Grinsztajn, Oyallon, and Varoquaux 2022).

Short Text Tasks. Short text representation and reasoning do not generally require extensive background knowledge. As a result, small models are particularly effective for tasks such as text classification (Zhang et al. 2024b), phrase representation (Chen, Varoquaux, and Suchanek 2024), and entity retrieval (Chen, Varoquaux, and Suchanek 2021).

Other Specialized Tasks. In certain niche areas, smaller models can surpass larger ones. Examples include machine-generated text detection (Mireshghallah et al. 2023), spreadsheet representation (Joshi et al. 2024), and information extraction (Ma et al. 2023b).

Across these settings, the advantages of small models arise from limited data availability and well-defined task structures. In domain-specific tasks and low-resource languages, training data is limited, which makes smaller models more data-efficient, easier to adapt, and often more practical to deploy. For tasks with distinctive and localized patterns, such as short-text understanding, the narrow semantic scope reduces the need for large contextual understanding. As a result, small models tend to show more stable behavior in task-constrained settings, and these advantages arise not from model size alone, but from a better alignment between task complexity and available data.

4.3 Interpretability-required Environment

The goal of interpretability is to provide a human-understandable explanation of a model's internal reasoning process (Lipton 2018; Gilpin et al. 2018), i.e., how the model works (*transparency*). Generally, smaller (e.g. shallow) and simpler (e.g. tree-based)

models offer better interpretability compared to larger (e.g. deep), more complex models (e.g. neural) (Barceló et al. 2020; Gosiewska, Kozak, and Biecek 2021).

In practice, industries such as healthcare (Caruana et al. 2015), finance (Kurshan et al. 2021), and law (Eliot 2021) often favor smaller, more interpretable models because the decisions produced by these models must be understandable to non-experts (e.g., doctors, financial analysts). In high-stakes decision-making contexts, models that can be easily audited and explained are typically preferred.

When selecting LLMs or SMs to use, it is important to make trade-offs for balancing model complexity with the need for human understanding.

Summary

Smaller models emerge as a viable alternative, which can achieve competitive performances in resource-limited scenarios. In domains such as healthcare and law, where transparency and interpretability are essential, practitioners often favour compact models because they are more auditable and easier to understand. At the same time, techniques such as knowledge distillation further enable small models to obtain capability from large models. In sum, small models represent not only a cost-sensitive substitute but also a strategic complement to large models.

Future Directions

(1) According to recent industry research, by 2028 roughly 68% of customer-service interactions are expected to be processed by *agentic systems* (Cisco Systems 2025). Smaller models are particularly well-suited for agentic systems that perform many small, specialized, and repetitive tasks. It is therefore important to explore how these compact models can serve as the economical components of agentic AI (Belcak et al. 2025).

(2) For some specific tasks such as domain-specific tasks and tabular reasoning, the performance of dedicated small models can even outperform general-purpose LLMs. How to organically combine this level of specialization with generality, and enable AI systems to expand their knowledge boundaries and handle a broader range of tasks, is a research topic worth studying.

5. Open Challenges for Small Models

Reliable Small Models. Despite their advantages in efficiency, small models continue to face several fundamental limitations. A major limitation is the capability gap compared with LLMs, particularly on knowledge reasoning tasks. Scaling laws suggest that performance generally improves with increasing model size (Kaplan et al. 2020), while several advanced capabilities emerge only beyond certain scales (Wei et al. 2022). Moreover, small models tend to be less robust in real-world deployment. They are more susceptible to distribution shifts, perturbations, and catastrophic forgetting during continual adaptation, which leads to degraded performance on unseen domains and evolving tasks (Chen, Varoquaux, and Suchanek 2022; Ramasesh, Lewkowycz, and Dyer 2022). Addressing both the capability gap and robustness remains a key challenge for the next generation of small models.

Open-world Model Collaboration. Current inference collaboration paradigms between small and large models, e.g., model routing and speculative decoding, are mainly designed for a fixed pool of homogeneous models, often from the same model family. As AI systems evolve toward open-world deployment, future collaboration is expected to involve a much larger ecosystem of heterogeneous models, including open-weight and API-only models with diverse capabilities. Designing collaboration mechanisms that scale to such heterogeneous and open environments remains largely unexplored. Although RouteLLM (Ong et al. 2025) demonstrates strong transferability, existing routing approaches generally assume a predefined set of candidate models, which cannot fully address open-world settings in which the candidate model pool may evolve dynamically. Future research should investigate how to accurately estimate the knowledge boundary and capability of each model (Chen et al. 2026), so that queries can be dynamically allocated to the most suitable model. Furthermore, existing approaches should explicitly consider collaborations in different architectures and modalities (Shen et al. 2023b; Remy et al. 2024).

Trustworthy Small Evaluators. As AI-generated content becomes increasingly widespread across society, efficiently evaluating its quality and reliability will become crucial. Small models are promising candidates for evaluating factuality, safety, and fairness of AI-generated content. However, ensuring these small evaluators are trustworthy remains a major challenge. Existing studies show that small judges can be sensitive to answer ordering (Li et al. 2024) and may exhibit biases (Ye et al. 2025), which leads to inconsistent and unreliable evaluation results. Future research should therefore focus on developing more robust and fair small evaluators, together with benchmarks and evaluation protocols for measuring their reliability.

Agentic Small Models. With the rapid usage of agentic AI across real-world applications (Cisco Systems 2025), small models are expected to become economical building blocks of future intelligent systems (Belcak et al. 2025). A key open challenge is how to design small models as specialized agents that efficiently perform dedicated functions, such as planning, retrieval, tool selection, code generation, document verification, or memory management, while collaborating with other agents to solve complex tasks. Despite their promising role, agentic small models remain largely underexplored. Developing specialized small agents that can efficiently collaborate with tools and other models is an important direction for future research.

6. Conclusion

In this work, we systematically analyze the relationship between LLMs and SMs from both collaborative and competitive perspectives. However, the notion of "small" is relative. As model sizes continue to grow, potentially toward ever-larger multimodal foundation models, today's large models may become tomorrow's small ones. Regarding the future relationship between small models and large models, we expect to observe an evolutionary dynamic. Although LLMs outperform smaller models in most tasks, small models persist by occupying ecological niches where large models are maladapted, such as low-latency inference, on-device deployment, and low-resource environments.

More importantly, the dominant trend might shift from competition to cooperation. Small models and LLMs increasingly work together through knowledge distillation,

efficient inference techniques, data curation, and evaluation frameworks. This division and cooperation create a diversified and balanced AI ecosystem in which models of different scales specialize and complement each other. Combining SMs and LLMs makes systems more scalable across different tasks, more energy-efficient, and better able to adapt to changing needs.

In summary, small models and large models coexist by specializing in different resource niches and increasingly cooperating to build a more efficient and robust AI ecosystem. Our relative definition of model size ensures that this core idea remains relevant as model scales continue to evolve.

References

Abdin, Marah, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Agarwal, Ishika, Krishnateja Killamsetty, Lucian Popa, and Marina Danilevsky. 2025. DELIFT: Data efficient language model instruction fine-tuning. In *The Thirteenth International Conference on Learning Representations*.

Aggarwal, Pranjal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, et al. 2024. Automix: Automatically mixing language models. *Advances in Neural Information Processing Systems*, 37:131000–131034.

Albalak, Alon, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. 2024. A survey on data selection for language models. *Transactions on Machine Learning Research*.

Anugraha, David, Genta Indra Winata, Chenyue Li, Patrick Amadeus Irawan, and En-Shiun Annie Lee. 2025. Proxylm: Predicting language model performance on multilingual tasks via proxy models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1981–2011.

Arnett, Catherine, Eliot Jones, Ivan P. Yamshchikov, and Pierre-Carl Langlais. 2024. Toxicity of the commons: Curating open-source pre-training data. *arXiv preprint arXiv:2410.22587*.

Asai, Akari, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Bai, Yuntao, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Bansal, Hritik, Arian Hosseini, Rishabh Agarwal, Vinh Q. Tran, and Mehran Kazemi. 2025. Smaller, weaker, yet better: Training LLM reasoners via compute-optimal sampling. In *The Thirteenth International Conference on Learning Representations*.

Barceló, Pablo, Mikaël Monet, Jorge Pérez, and Bernardo Subercaseaux. 2020. Model interpretability through the lens of computational complexity. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Belcak, Peter, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. 2025. Small language models are the future of agentic ai. *arXiv preprint arXiv:2506.02153*.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In

Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Bucher, Martin Juan José and Marco Martini. 2024. Fine-tuned'small'llms (still) significantly outperform zero-shot generative ai models in text classification. *arXiv preprint arXiv:2406.08660*.

Burchell, Laurie, Alexandra Birch, Nikolay Bogoychev, and Kenneth Heafield. 2023. An open dataset and model for language identification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 865–879.

Burns, Collin, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. 2024. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In *Forty-first International Conference on Machine Learning*.

Caruana, Rich, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*.

Chalkidis, Ilias. 2023. Chatgpt may pass the bar exam soon, but has a long way to go for the lexglue benchmark. *arXiv preprint arXiv:2304.12202*.

Chang, Yupeng, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, (3).

Chen, Charlie, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023a. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Chen, Dong, Yueting Zhuang, Shuo Zhang, Jinfeng Liu, Su Dong, and Siliang Tang. 2024a. Data shunt: Collaboration of small and large models for lower costs and better performance. In *Proc. of AAAI*, 10.

Chen, Jiaao, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2023b. An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*.

Chen, Jie, Yupeng Zhang, Bingning Wang, Wayne Xin Zhao, Ji-Rong Wen, and Weipeng Chen. 2024b. Unveiling the flaws: Exploring imperfections in synthetic data and mitigation strategies for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14855–14865, Association for Computational Linguistics, Miami, Florida, USA.

Chen, Lihu, Gerard de Melo, Fabian M. Suchanek, and Gaël Varoquaux. 2026. Query-level uncertainty in large language models. In *The Fourteenth International Conference on Learning Representations*.

Chen, Lihu, Alexandre Perez-Lebel, Fabian Suchanek, and Gaël Varoquaux. 2024c. Reconfidencing llms from the grouping loss perspective. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1567–1581.

Chen, Lihu, Gael Varoquaux, and Fabian Suchanek. 2022. Imputing out-of-vocabulary embeddings with love makes languagemodels robust with little cost. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3488–3504.

Chen, Lihu, Gaël Varoquaux, and Fabian M Suchanek. 2021. A lightweight neural model for biomedical entity linking. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 12657–12665.

Chen, Lihu, Gaël Varoquaux, and Fabian M Suchanek. 2024. Learning high-quality and general-purpose phrase representations. In *EACL 2024-The 18th Conference of the European Chapter of the Association for Computational Linguistics*.

Chen, Lingjiao, Matei Zaharia, and James Zou. 2024. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*.

Chen, Lingjiao, Matei Zaharia, and James Y. Zou. 2020. Frugalml: How to use ML prediction apis more accurately and cheaply. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Chen, Maximillian, Alexandros Papangelis, Chenyang Tao, Andy Rosenbaum, Seokhwan Kim, Yang Liu, Zhou Yu, and Dilek Hakkani-Tur. 2022. Weakly supervised data augmentation through prompting for dialogue understanding. In *NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research*.

Chen, Zhou, Zhiqiang Wei, Yuqi Bai, Xue Xiong, and Jianmin Wu. 2025. Tagrouter: Learning route to llms through tags for open-domain text generation tasks. In *Findings of ACL 2025*.

Cheng, Daixuan, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Weiwei Deng, and Qi Zhang. 2023. Uprise: Universal prompt retrieval for improving zero-shot evaluation. In *Proc. of EMNLP*.

Cheng, Xiaoxue, Junyi Li, Wayne Xin Zhao, Hongzhi Zhang, Fuzheng Zhang, Di Zhang, Kun Gai, and Ji-Rong Wen. 2024. Small agent can also rock! empowering small language models as hallucination detector. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14600–14615.

Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, (240).

Chung, John, Ece Kamar, and Saleema Amershi. 2023. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In *Proc. of ACL*.

Cisco Systems, Inc. 2025. Agentic ai poised to handle 68% of customer service and support interactions by 2028. Press Release, Cisco Newsroom.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bing-Li Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dong-Li Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, JingChang Chen, Jingyang Yuan, Junjie Qiu, Jun-Long Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, M. Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, Rui-Qi Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shao-Kang Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wen-Xia Yu, Wentao Zhang, Wangding Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xin-Yu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyu Jin, Xi-Cheng Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yi Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yu-Jing Zou, Yujia He, Yunfan Xiong, Yu-Wei Luo, Yu mei You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yao Li, Yi Zheng, Yuchen Zhu, Yunxiang Ma, Ying Tang, Yukun Zha, Yuting Yan, Zehui Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhen guo Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zi-Long Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633 – 638.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT*.

Dhar, Nobel, Bobin Deng, Dan Lo, Xiaofeng Wu, Liang Zhao, and Kun Suo. 2024. An empirical analysis and resource footprint study of deploying large language models on edge devices. In *Proceedings of the 2024 ACM Southeast Conference*.

Dhuliawala, Shehzaad, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the association for computational linguistics: ACL 2024*, pages 3563–3578.

Ding, Bosheng, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Luu Anh Tuan, and Shafiq Joty. 2024a. Data augmentation using llms: Data perspectives, learning paradigms and challenges. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1679–1705.

Ding, Dujian, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. 2024b. Hybrid llm: Cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations*.

Dong, Qingxiu, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Association for Computational Linguistics, Miami, Florida, USA.

Doshi, Meet, Raj Dabre, and Pushpak Bhattacharyya. 2024. Pretraining language models using translationese. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5843–5862.

Du, Nan, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P. Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen S. Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research.

Du, Qianlong, Chengqing Zong, and Jiajun Zhang. 2023. Mods: Model-oriented data selection for instruction tuning. *arXiv preprint arXiv:2311.15653*.

Dubey, Abhimanyu, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Eliot, Dr Lance B. 2021. The need for explainable ai (xai) is especially crucial in the law. *Available at SSRN 3975778*.

Enomoto, Shohei and Takeharu Eda. 2021. Learning to cascade: Confidence calibration for improving the accuracy and computational cost of cascade inference systems. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*.

Feng, Shangbin, Weijia Shi, Yuyang Bai, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. 2024a. Knowledge card: Filling llms' knowledge gaps with plug-in specialized language models. In *The Twelfth International Conference on Learning Representations*.

Feng, Tao, Yicheng Li, Li Chenglin, Hao Chen, Fei Yu, and Yin Zhang. 2024b. Teaching small language models reasoning through counterfactual distillation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5831–5842.

Feng, Zhangyin, Weitao Ma, Weijiang Yu, Lei Huang, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. Trends in integration of knowledge and large language models: A survey and taxonomy of methods, benchmarks, and applications.

Fu, Yao, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning*, PMLR.

Gao, Chengqian, Haonan Li, Liu Liu, Zeke Xie, Peilin Zhao, and Zhiqiang Xu. 2025. Principled data selection for alignment: The hidden risks of difficult examples. In *International Conference on Machine Learning*, pages 18386–18409, PMLR.

Gao, Jiahui, Renjie Pi, Lin Yong, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2023a. Self-guided noise-free data generation for efficient zero-shot learning. In *The Eleventh international conference on learning representations*.

Gao, Leo, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2021. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Gao, Yunfan, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Gilpin, Leilani H, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, IEEE.

Gosiewska, Alicja, Anna Kozak, and Przemysław Biecek. 2021. Simpler is better: Lifting interpretability-performance trade-off via automated feature engineering. *Decision Support Systems*.

Gou, Jianping, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, (6).

Grinsztajn, Léo, Edouard Oyallon, and Gaël Varoquaux. 2022. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*.

Gu, Jiawei, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024a. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Gu, Yuxian, Li Dong, Furu Wei, and Minlie Huang. 2024b. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Guha, Neel, Mayee F Chen, Trevor Chow, Ishan S. Khare, and Christopher Re. 2024. Smoothie: Label free language model routing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Guo, Jianyuan, Hanting Chen, Chengcheng Wang, Kai Han, Chang Xu, and Yunhe Wang. 2024. Vision superalignment: Weak-to-strong generalization for vision foundation models. *arXiv preprint arXiv:2402.03749*.

Guo, Yue and Yi Yang. 2024. Improving weak-to-strong generalization with reliability-aware alignment. *arXiv preprint arXiv:2406.19032*.

Gurgurov, Daniil, Ivan Vykopal, Josef van Genabith, and Simon Ostermann. 2025. Small models, big impact: Efficient corpus and graph-based adaptation of small multilingual language models for low-resource languages. *arXiv preprint arXiv:2502.10140*.

Hartvigsen, Thomas, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proc. of ACL*.

He, Kai, Rui Mao, Qika Lin, Yucheng Ruan, Xiang Lan, Mengling Feng, and Erik Cambria. 2023. A survey of large language models for healthcare: from data, technology, and applications to accountability and ethics. *arXiv preprint arXiv:2310.05694*.

He, Pengcheng, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *Proc. of ICLR*.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hsieh, Cheng-Yu, Chun-Liang Li, CHIH-KUAN YEH, Hootan Nakhost, Yasuhisa Fujii, Alex Jason Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Hsu, I-Hung, Zifeng Wang, Long Le, Lesly Miculicich Werlen, Nanyun Peng, Chen-Yu Lee, and Tomas Pfister. 2024. Calm: Contrasting large and small language models to verify grounded generation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12782–12803.

Hu, Linmei, Hongyu He, Duokang Wang, Ziwang Zhao, Yingxia Shao, and Liqiang Nie. 2024a. Llm vs small model? large language model based text augmentation enhanced personality detection model. In *Proc. of AAAI*, 16.

Hu, Qitian Jason, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024b. Routerbench: A benchmark for multi-LLM routing system. In *Agentic Markets Workshop at ICML 2024*.

Hu, Shengding, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, dahai li, Zhiyuan Liu, and Maosong Sun. 2024c. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *First Conference on Language Modeling*.

Huang, Lei, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

ACM Transactions on Information Systems, 43(2):1–55.
Huang, Yuheng, Jiayang Song, Zhijie Wang, Shengming Zhao, Huaming Chen, Felix Juefei-Xu, and Lei Ma. 2025b. Look before you leap: An exploratory study of uncertainty analysis for large language models. *IEEE Transactions on Software Engineering*, 51(2):413–429.
Izacard, Gautier, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.
Jang, SiYoung and Roberto Morabito. 2025. Edge-first language model inference: Models, metrics, and tradeoffs. In *2025 IEEE 45th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 309–314, IEEE.
Ji, Jiaming, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Tianyi Qiu, Juntao Dai, and Yaodong Yang. 2024. Aligner: Efficient alignment by learning to correct. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
Jiang, Dongfu, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.
Jiang, Jinhao, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023a. Structgpt: A general framework for large language model to reason over structured data. In *Proc. of EMNLP*.
Jiang, Juyong, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. 2026. A survey on large language models for code generation. *ACM Transactions on Software Engineering and Methodology*, 35(2):1–72.
Jiang, Yuxin, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023b. Lion: Adversarial distillation of proprietary large language models. In *Proc. of EMNLP*.
Joshi, Harshit, Abishai Ebenezer, José Cambronero Sanchez, Sumit Gulwani, Aditya Kanade, Vu Le, Ivan Radiček, and Gust Verbruggen. 2024. Flame: A small language model for spreadsheet formulas. In *Proc. of AAAI*, 12.
Josifoski, Martin, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting asymmetry for synthetic training data generation: Synthie and the case of information extraction. In *Proc. of EMNLP*.
Juneja, Gurusha, Subhabrata Dutta, Soumen Chakrabarti, Sunny Manchanda, and Tanmoy Chakraborty. 2023. Small language models fine-tuned to coordinate larger language models improve complex reasoning. In *Proc. of EMNLP*.
Kag, Anil, Igor Fedorov, Aditya Gangrade, Paul Whatmough, and Venkatesh Saligrama. 2023. Efficient edge inference by selective query. In *The Eleventh International Conference on Learning Representations*.
Kamphuis, Michiel. 2024. Tiny-toxic-detector: A compact transformer-based model for toxic content detection. *arXiv preprint arXiv:2409.02114*.
Kang, Feiyang, Yifan Sun, Bingbing Wen, Si Chen, Dawn Song, Rafid Mahmood, and Ruoxi Jia. 2024. Autoscale: Automatic prediction of compute-optimal data composition for training llms. *arXiv preprint arXiv:2407.20177*.
Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.
Kargaran, Amir Hossein, Ayyoob Imani, François Yvon, and Hinrich Schütze. 2023. Glotlid: Language identification for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6155–6218.
Kasai, Jungo, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, et al. 2023. Realtime qa: What's the answer right now? *Advances in neural information processing systems*, 36:49025–49043.
Kuhn, Lorenz, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.
Kurshan, Eren, Jiahao Chen, Victor Storchan, and Hongda Shen. 2021. On the current and emerging challenges of developing fair and ethical ai solutions in financial services. In *Proceedings of the second ACM international conference on AI in finance*.
Lang, Hao, Fei Huang, and Yongbin Li. 2025. Debate helps weak-to-strong generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27410–27418.

Lang, Hunter, David Sontag, and Aravindan Vijayaraghavan. 2024. Theoretical analysis of weak-to-strong generalization. *Advances in neural information processing systems*, 37:46837–46880.

Lee, Chia-Hsuan, Hao Cheng, and Mari Ostendorf. 2024. Orchestrallm: Efficient orchestration of language models for dialogue state tracking. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*.

Lee, Jooyoung, Fan Yang, Thanh Tran, Qian Hu, Emre Barut, and Kai-Wei Chang. 2024. Can small language models help large language models reason better?: Lm-guided chain-of-thought. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*.

Lehman, Eric, Evan Hernandez, Diwakar Mahajan, Jonas Wulff, Micah J Smith, Zachary Ziegler, Daniel Nadler, Peter Szolovits, Alistair Johnson, and Emily Alsentzer. 2023. Do we still need clinical language models? In *Conference on health, inference, and learning*, pages 578–597, PMLR.

Leviathan, Yaniv, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, PMLR.

Lewis, Patrick S. H., Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Li, Haitao, Qingyao Ai, Jia Chen, Qian Dong, Zhijing Wu, and Yiqun Liu. 2025a. Blade: Enhancing black-box large language models with small domain-specific models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24422–24430.

Li, Haitao, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Li, Liunian Harold, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023a. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In *Proc. of ACL*.

Li, Moxin, Yong Zhao, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See-Kiong Ng, Tat-Seng Chua, and Yang Deng. 2025b. Knowledge boundary of large language models: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5131–5157, Association for Computational Linguistics, Vienna, Austria.

Li, Shiyang, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, et al. 2022. Explanations from large language models make small reasoners better. *arXiv preprint arXiv:2210.06726*.

Li, Xiang Lisa, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023b. Contrastive decoding: Open-ended text generation as optimization. In *Proc. of ACL*.

Li, Zhuoyan, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023c. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proc. of EMNLP*.

Liang, Chen, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*, PMLR.

Lin, Chin-Yew. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Ling, Chen, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, Tianjiao Zhao, Amit Panalkar, Dhagash Mehta, Stefano Pasquali, Wei Cheng, Haoyu Wang, Yanchi Liu, Zhengzhang Chen, Haifeng Chen, Chris White, Quanquan Gu, Jian Pei, Carl Yang, and Liang Zhao. 2025. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *ACM Comput. Surv.*, 58(3).

Lipton, Zachary C. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, (3).

Liu, Alisa, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024a. Tuning language models by proxy. In *First Conference on Language Modeling*.

Liu, Chia-Wei, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised

evaluation metrics for dialogue response generation. In *Proc. of EMNLP*.
Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, (9).
Liu, Ruibo, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and Andrew M. Dai. 2024b. Best practices and lessons learned on synthetic data. In *First Conference on Language Modeling*.
Liu, Wei, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024c. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.
Liu, Yuejiang and Alexandre Alahi. 2024. Co-supervised learning: Improving weak-to-strong generalization with hierarchical mixture of experts. *arXiv preprint arXiv:2402.15505*.
Liu, Zechun, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2024d. Llm-qat: Data-free quantization aware training for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 467–484.
Liu, Ziche, Rui Ke, Yajiao Liu, Feng Jiang, and Haizhou Li. 2025. Take the essence and discard the dross: A rethinking on data selection for fine-tuning 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)*, pages 6595–6611, Association for Computational Linguistics, Albuquerque, New Mexico.
Long, Lin, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082.
Longpre, Shayne, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, PMLR.
Longpre, Shayne, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, et al. 2024. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*.
Lu, Keming, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. Routing to the expert: Efficient reward-guided ensemble of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1964–1974.
Lu, Ximing, Faeze Brahman, Peter West, Jaehun Jung, Khyathi Chandu, Abhilasha Ravichander, Prithviraj Ammanabrolu, Liwei Jiang, Sahana Ramnath, Nouha Dziri, et al. 2023. Inference-time policy adapters (ipa): Tailoring extreme-scale lms without fine-tuning. In *Proc. of EMNLP*.
Lu, Zhenyan, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Wei Liu, Jian Luan, Xiwen Zhang, Nicholas D Lane, and Mengwei Xu. 2025. Demystifying small language models for edge deployment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14747–14764.
Ma, Xinbei, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023a. Query rewriting in retrieval-augmented large language models. In *Proc. of EMNLP*.
Ma, Yubo, Yixin Cao, Yong Hong, and Aixin Sun. 2023b. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*.
Magister, Lucie Charlotte, Jonathan Mallinson, Jakub Dominik Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.
Manakul, Potsawee, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proc. of EMNLP*.
Marion, Max, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. 2023. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*.

Meng, Yu, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*.

Mi, Chenggang, Lei Xie, and Yanning Zhang. 2022. Improving data augmentation for low resource speech-to-text translation with diverse paraphrasing. *Neural Networks*.

Min, Sewon, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proc. of EMNLP*.

Mireshghallah, Fatemehsadat, Justus Mattern, Sicun Gao, Reza Shokri, and Taylor Berg-Kirkpatrick. 2023. Smaller language models are better black-box machine-generated text detectors. *arXiv preprint arXiv:2305.09859*.

Mondorf, Philipp and Barbara Plank. 2024. Beyond accuracy: Evaluating the reasoning behavior of large language models - a survey. In *First Conference on Language Modeling*.

Muennighoff, Niklas, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*.

Nadăș, Mihai, Laura Dioșan, and Andreea Tomescu. 2025. Synthetic data generation using large language models: Advances in text and code. *IEEE Access*.

Narayanan Hari, Surya and Matt Thomson. 2023. Tryage: Real-time, intelligent routing of user prompts to large language models. *arXiv preprint arXiv:2308.11601*.

Nayak, Nihal, Yiyang Nan, Avi Trost, and Stephen Bach. 2024. Learning to generate instruction tuning datasets for zero-shot task adaptation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12585–12611.

Nie, Ercong, Sheng Liang, Helmut Schmid, and Hinrich Schütze. 2023. Cross-lingual retrieval augmented prompt for low-resource languages. In *Findings of the Association for Computational Linguistics: ACL 2023*.

NLLB Team. 2024. Scaling neural machine translation to 200 languages. *Nat.*, 630(8018):841–846.

O'Brien, Sean and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models. *arXiv preprint arXiv:2309.09117*.

Oepen, Stephan, Nikolay Arefev, Mikko Aulamo, Marta Bañón, Maja Buljan, Laurie Burchell, Lucas Charpentier, Pinzhen Chen, Mariya Fedorova, Ona de Gibert, et al. 2025. Hplt 3.0: Very large-scale multilingual resources for llm and mt. mono-and bi-lingual data, multilingual evaluation, and pre-trained models. *arXiv preprint arXiv:2511.01066*.

Ollion, Etienne, Rubing Shen, Ana Macanovic, and Arnault Chatelain. 2023. Chatgpt for text annotation? mind the hype. *SocArXiv preprint*.

Ong, Isaac, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2025. RouteLLM: Learning to route LLMs from preference data. In *The Thirteenth International Conference on Learning Representations*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ormazabal, Aitor, Mikel Artetxe, and Eneko Agirre. 2023. Comblm: Adapting black-box language models through small fine-tuned models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2961–2974.

Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*.

Pan, Feifei, Mustafa Canim, Michael Glass, Alfio Gliozzo, and James Hendler. 2022. End-to-end table question answering via retrieval-augmented generation. *arXiv preprint arXiv:2203.16714*.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

Penedo, Guilherme, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Penedo, Guilherme, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. 2025. Fineweb2: One pipeline to scale them all–adapting pre-training data processing to every language. *arXiv preprint arXiv:2506.20920*.

Penedo, Guilherme, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon LLM: Outperforming curated corpora with web data only. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Pinter, Yuval, Robert Guthrie, and Jacob Eisenstein. 2017. Mimicking word embeddings using subword RNNs. In *Proc. of EMNLP*.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*

Ramasesh, Vinay Venkatesh, Aitor Lewkowycz, and Ethan Dyer. 2022. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*.

Reimers, Nils and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proc. of EMNLP*.

Remy, François, Pieter Delobelle, Hayastan Avetisyan, Alfiya Khabibullina, Miryam de Lhoneux, and Thomas Demeester. 2024. Trans-tokenization and cross-lingual vocabulary transfers: Language adaptation of llms for low-resource nlp. In "First Conference on Language Modeling (COLM) 2024", arXiv pre-print arXiv:2408.04303.

Robertson, Stephen, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, (4).

Sahu, Gaurav, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*.

Šakota, Marija, Maxime Peyrard, and Robert West. 2024. Fly-swat or cannon? cost-effective language model choice via meta-modeling. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*.

Samarinas, Chris and Hamed Zamani. 2025. Distillation and refinement of reasoning in small language models for document re-ranking. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, pages 430–435.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Schick, Timo, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Seals, Spencer M. and Valerie L. Shalin. 2024. Evaluating the deductive competence of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8614–8630, Association for Computational Linguistics, Mexico City, Mexico.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 86–96.

Sennrich, Rico, Jannis Vamvas, and Alireza Mohammadshahi. 2024. Mitigating hallucinations and off-target machine translation with source-contrastive and language-contrastive decoding. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Shen, Tianhao, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan Liu, and Deyi Xiong. 2023a. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*.

Shen, Yongliang, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023b. HuggingGPT: Solving AI tasks with chatGPT and its friends in hugging face. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Shi, Weijia, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024a. Replug: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384.

Shi, Wenqi, Ran Xu, Yuchen Zhuang, Yue Yu, Haotian Sun, Hang Wu, Carl Yang, and May D. Wang. 2024b. Medadapter: Efficient test-time adaptation of large language models towards

medical reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024)*, pages 22294–22314, Association for Computational Linguistics, Miami, Florida, USA.

Shnitzer, Tal, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. 2024. Large language model routing with benchmark datasets. In *First Conference on Language Modeling*.

Shridhar, Kumar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models. In *Findings of the Association for Computational Linguistics: ACL 2023*.

Shuster, Kurt, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Sinha, Sania, Tanawan Premsri, and Parisa Kordjamshidi. 2024. A survey on compositional learning of AI models: Theoretical and experimental practices. *Transactions on Machine Learning Research*.

Siriwardhana, Shamane, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*.

Su, Jiayuan, Fulin Lin, Zhaopeng Feng, Han Zheng, Teng Wang, Zhenyu Xiao, Xinlong Zhao, Zuozhu Liu, Lu Cheng, and Hongwei Wang. 2026. Cp-router: An uncertainty-aware router between llm and lrm. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 33065–33073.

Subramani, Nishant, Sasha Luccioni, Jesse Dodge, and Margaret Mitchell. 2023. Detecting personal information in training corpora: an analysis. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 208–220, Association for Computational Linguistics, Toronto, Canada.

Subramanian, Shreyas, Vikram Elango, and Mecit Gungor. 2025. Small language models (slms) can still pack a punch: A survey. *arXiv preprint arXiv:2501.05465*.

Sun, Haotian, Yuchen Zhuang, Wei Wei, Chao Zhang, and Bo Dai. 2024. Bbox-adapter: Lightweight adapting for black-box large language models. In *International Conference on Machine Learning*, pages 47280–47304, PMLR.

Sun, Zhongxiang. 2023. A short survey of viewing large language models in legal aspect. *arXiv preprint arXiv:2303.09136*.

Tang, Ruixiang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining? *arXiv preprint arXiv:2303.04360*.

Thakkar, Megh, Tolga Bolukbasi, Sriram Ganapathy, Shikhar Vashishth, Sarath Chandar, and Partha Talukdar. 2023. Self-influence guided data reweighting for language model pre-training. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2033–2045, Association for Computational Linguistics, Singapore.

Thakur, Aman Singh, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2025. Judging the judges: Evaluating alignment and vulnerabilities in llms-as-judges. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM^2)*, pages 404–430.

Tian, Katherine, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442.

Tian, Yijun, Yikun Han, Xiusi Chen, Wei Wang, and Nitesh V Chawla. 2024. Tinyllm: Learning a small student from multiple large language models. *arXiv preprint arXiv:2402.04616*.

Tirumala, Kushal, Daniel Simig, Armen Aghajanyan, and Ari Morcos. 2023. D4: Improving llm pretraining via document de-duplication and diversification. *Advances in Neural Information Processing Systems*.

Trivedi, Harsh, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proc. of ACL*.

Van Nguyen, Chien, Xuan Shen, Ryan Aponte, Yu Xia, Samyadeep Basu, Zhengmian Hu, Jian Chen, Mihir Parmar, Sasidhar Kunapuli, Joe Barrow, et al. 2024. A survey of small language

models. *arXiv preprint arXiv:2410.20011*.
Varoquaux, Gaël, Alexandra Sasha Luccioni, and Meredith Whittaker. 2025. Hype, sustainability, and the price of the bigger-is-better paradigm in ai. In *FAccT 2025-ACM Conference on Fairness, Accountability, and Transparency*.
Vernikos, Giorgos, Arthur Bražinskas, Jakub Adamek, Jonathan Mallinson, Aliaksei Severyn, and Eric Malmi. 2024. Small language models improve giants by rewriting their outputs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*.
Villalobos, Pablo, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Position: Will we run out of data? limits of llm scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*.
Wan, Zhongwei, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, et al. 2023. Efficient large language models: A survey. *Transactions on Machine Learning Research*.
Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proc. of ICLR*.
Wang, Congchao, Sean Augenstein, Keith Rush, Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Aditya Krishna Menon, and Alec Go. 2024a. Cascade-aware training of language models. *arXiv preprint arXiv:2406.00060*.
Wang, Fali, Zhiwei Zhang, Xianren Zhang, Zongyu Wu, Tzuhao Mo, Qiuhao Lu, Wanjing Wang, Rui Li, Junjie Xu, Xianfeng Tang, et al. 2024b. A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness. *arXiv preprint arXiv:2411.03350*.
Wang, Jiaan, Fandong Meng, Yingxue Zhang, and Jie Zhou. 2025a. Retrieval-augmented machine translation with unstructured knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5858–5871, Association for Computational Linguistics, Suzhou, China.
Wang, Jiayi, Yao Lu, Maurice Weber, Max Ryabinin, David Ifeoluwa Adelani, Yihong Chen, Raphael Tang, and Pontus Stenetorp. 2025b. Multilingual language model pretraining using machine-translated data. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28075–28095.
Wang, Shaobo, Xiangqi Jin, Ziming Wang, Jize Wang, Jiajun Zhang, Kaixin Li, Zichen Wen, Zhong Li, Conghui He, Xuming Hu, and Linfeng Zhang. 2025c. Data whisperer: Efficient data selection for task-specific llm fine-tuning via few-shot in-context learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23287–23305, Association for Computational Linguistics, Vienna, Austria.
Wang, Shuohang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? GPT-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.
Wang, Xin, Yujia Luo, Daniel Crankshaw, Alexey Tumanov, Fisher Yu, and Joseph E. Gonzalez. 2018. IDK cascades: Fast deep learning by learning not to overthink. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 580–590, AUAI Press.
Wang, Xintao, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. 2023. Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases. *arXiv preprint arXiv:2308.11761*.
Wang, Zifeng, Chun-Liang Li, Vincent Perot, Long Le, Jin Miao, Zizhao Zhang, Chen-Yu Lee, and Tomas Pfister. 2024c. Codeclm: Aligning language models with tailored synthetic data. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3712–3729.
Warstadt, Alex, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, et al. 2023. Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34.
Wei, Jason, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Weller, Orion, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. 2025. On the theoretical limitations of embedding-based retrieval. *arXiv preprint arXiv:2508.21038*.

Wen, Xiangyu, Junhua Huang, Zeju Li, Min Li, Jianyuan Zhong, Zhijian Xu, Mingxuan Yuan, Yongxiang Huang, and Qiang Xu. 2025. Reasoning scaffolding: Distilling the flow of thought from llms. *arXiv preprint arXiv:2509.23619*.

Wenzek, Guillaume, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*.

Wettig, Alexander, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. Qurating: Selecting high-quality data for training language models. In *Forty-first International Conference on Machine Learning*.

Witteveen, Sam and Martin Andrews. 2019. Paraphrasing with large language models. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*.

Wu, Carole-Jean, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. 2022. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*.

Xia, Heming, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. 2024a. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7655–7671.

Xia, Mengzhou, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024b. Less: Selecting influential data for targeted instruction tuning. In *Forty-first International Conference on Machine Learning*.

Xie, Sang Michael, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2023a. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems*, volume 36, pages 69798–69818, Curran Associates, Inc.

Xie, Sang Michael, Shibani Santurkar, Tengyu Ma, and Percy S Liang. 2023b. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*.

Xiong, Guangzhi, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6233–6251.

Xu, Canwen, Yichong Xu, Shuohang Wang, Yang Liu, Chenguang Zhu, and Julian McAuley. 2024a. Small models are valuable plug-ins for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 283–294.

Xu, Xiaohan, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024b. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.

Yang, Kailai, Xiao Liu, Lei Ji, Hao Li, Yeyun Gong, Peng Cheng, and Mao Yang. 2025. Data mixing agent: Learning to re-weight domains for continual pre-training. *arXiv preprint arXiv:2507.15640*.

Yang, Zhendong, Ailing Zeng, Zhe Li, Tianke Zhang, Chun Yuan, and Yu Li. 2023. From knowledge distillation to self-knowledge distillation: A unified approach with normalized loss and customized soft labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17185–17194.

Yao, Zhewei, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*.

Ye, Jiacheng, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient zero-shot learning via dataset generation. In *Proc. of EMNLP*.

Ye, Jiayi, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V Chawla, and Xiangliang Zhang. 2025. Justice or prejudice? quantifying biases in LLM-as-a-judge. In *The Thirteenth International Conference on Learning Representations*.

Yoran, Ori, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

Yu, Da, Sivakanth Gopi, Janardhan Kulkarni, Zinan Lin, Saurabh Naik, Tomasz Lukasz Religa, Jian Yin, and Huishuai Zhang. 2024. Selective pre-training for private fine-tuning. *Transactions on Machine Learning Research*.

Yu, Zezhu, Suqing Liu, Paul Denny, Andreas Bergen, and Michael Liut. 2025. Integrating small language models with retrieval-augmented generation in computing education: Key takeaways, setup, and practical insights. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, pages 1302–1308.

Yuan, Weizhe, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*.

Yue, Shengbin, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, et al. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325*.

Zhang, Kaiyan, Jianyu Wang, Ermo Hua, Biqing Qi, Ning Ding, and Bowen Zhou. 2024a. Cogenesis: A framework collaborating large and small language models for secure context-aware instruction following. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4295–4312.

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *Proc. of ICLR*.

Zhang, Wenxuan, Yue Deng, Bing Liu, Sinno Pan, and Lidong Bing. 2024b. Sentiment analysis in the era of large language models: A reality check. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3881–3906.

Zhang, Xuechen, Zijian Huang, Ege Onur Taga, Carlee Joe-Wong, Samet Oymak, and Jiasi Chen. 2024c. Efficient contextual llm cascades through budget-constrained policy learning. In *Advances in Neural Information Processing Systems (NeurIPS 2024)*.

Zhang, Zhexin, Leqi Lei, Lindong Wu, Rui Sun, Yongkang Huang, Chong Long, Xiao Liu, Xuanyu Lei, Jie Tang, and Minlie Huang. 2024d. Safetybench: Evaluating the safety of large language models. In *Proc. of ACL*.

Zhou, Chunting, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhou, Shuyan, Uri Alon, Frank F Xu, Zhengbao Jiang, and Graham Neubig. 2023b. Docprompting: Generating code by retrieving the docs. In *The Eleventh International Conference on Learning Representations*.

Zhou, Zhanhui, Zhixuan Liu, Jie Liu, Zhichen Dong, Chao Yang, and Yu Qiao. 2024. Weak-to-strong search: Align large language models via searching over small language models. *Advances in Neural Information Processing Systems*, 37:4819–4851.

Zhu, Xuekai, Biqing Qi, Kaiyan Zhang, Xinwei Long, Zhouhan Lin, and Bowen Zhou. 2024a. Pad: Program-aided distillation can teach small models reasoning better than chain-of-thought fine-tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2571–2597.

Zhu, Xunyu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024b. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577.