Training Strategies and Hardware Considerations for Text-to-Text Generation Models

I. Introduction: The Landscape of Text-to-Text Generation Training

The creation of high-performing text-to-text models has become increasingly vital across a multitude of applications, ranging from the generation of creative content and the translation of languages to the summarization of lengthy documents and the provision of answers to complex questions [1]. These models, particularly those categorized as large language models (LLMs) and boasting parameter counts in the billions, present a unique set of challenges. The sheer scale of these models demands substantial computational resources, often requiring extensive training periods and incurring significant financial costs [3]. Consequently, the pursuit of efficient and accurate text-to-text models necessitates a thorough exploration of advanced training techniques and optimization strategies designed to address these inherent complexities.

This report aims to provide a comprehensive analysis of various training methodologies applicable to text-to-text generation, with a specific emphasis on knowledge distillation as a potent optimization technique [3]. Furthermore, it will examine the crucial role of model size, quantified by the number of parameters, in influencing both the performance and the resource demands of these models [7]. Guidance on the strategic selection of appropriate teacher models for knowledge distillation will also be provided [3]. A detailed comparative analysis of GPU and TPU infrastructure available on both Microsoft Azure and Google Cloud will inform specific hardware recommendations tailored to these platforms [7]. Finally, the report will discuss cloud-specific services and best practices relevant to LLM training and distillation on these leading cloud platforms [15]. The objective is to serve as a holistic guide, encompassing the entire lifecycle of training a text-to-text model, from the initial selection of a training approach to the final considerations for deployment infrastructure.

II. Foundational Training Approaches for Text-to-Text Models

A variety of training paradigms form the bedrock of developing effective text-to-text models. These approaches lay the groundwork for the model's ability to understand, generate, and manipulate textual data.

The process often begins with **pre-training**, where a model is exposed to massive quantities of unlabeled text data. This phase allows the model to learn fundamental

patterns and representations of language, establishing a broad base of linguistic knowledge that can be leveraged for more specific tasks later on [20]. For instance, models from OpenAI and others undergo extensive pre-training on vast textual datasets [20]. This initial training equips the model with an understanding of grammar, syntax, and the relationships between words and concepts.

Following pre-training, **fine-tuning** is frequently employed to adapt the model to a particular downstream task. This involves training the pre-trained model on a smaller, labeled dataset that is specific to the desired task [7]. For example, GPT-J, a 6 billion parameter model, can be effectively fine-tuned for various natural language processing tasks [7]. The fine-tuning process allows the model to specialize its capabilities, becoming more adept at performing specific text-to-text generation tasks such as summarization, translation, or question answering.

Transfer learning is a closely related concept that emphasizes the leveraging of knowledge gained during pre-training on a large dataset to enhance performance on a different, but related, task where less data might be available [7]. The unified text-to-text transformer approach, for example, demonstrates the power of transfer learning by using the same model architecture and pre-trained starting point for many different tasks [7]. Models like T5 are specifically designed as text-to-text transfer transformers, treating every NLP task as a text-to-text problem [22]. This approach significantly reduces the training time and data requirements for new tasks by capitalizing on the knowledge already acquired by the model.

In scenarios where labeled data is scarce, **few-shot learning** and **zero-shot learning** offer alternative training paradigms. Few-shot learning enables a model to learn to perform new tasks with only a handful of labeled examples, often achieved through sophisticated prompting techniques or by fine-tuning on the limited data [30]. For instance, knowledge-augmentation methods have significantly improved few-shot table-to-text generation [30]. Zero-shot learning takes this a step further, allowing a model to perform tasks without having seen any task-specific labeled data. This relies entirely on the model's pre-existing knowledge and the effectiveness of the prompts used to guide its behavior [31]. These techniques highlight the remarkable generalization abilities that can be achieved by large pre-trained models.

Beyond these standard approaches, **multi-task learning** involves training a single model to handle multiple distinct tasks concurrently [1]. By learning from a variety of tasks, the model can often develop more robust and versatile representations, potentially improving performance on individual tasks and reducing the need for extensive task-specific fine-tuning in some cases [21]. For example, training on multiple

text generation tasks like translation and writing assistance can lead to a more generally capable model.

Finally, **Generative Adversarial Networks (GANs)** represent an alternative approach to text generation. GANs employ a generator network to create text and a discriminator network to evaluate the authenticity of the generated text, training these two networks in an adversarial manner [38]. While GANs can produce highly realistic text, they can be challenging to train effectively and are not as commonly used for general text-to-text tasks compared to transformer-based models that rely on the aforementioned training paradigms.

III. The Power of Knowledge Distillation in LLM Training

Knowledge distillation (KD) has emerged as a powerful technique in the training of large language models. It centers on the idea of transferring the knowledge and capabilities of a large, complex model, often referred to as the "teacher" model, to a smaller, more efficient "student" model [3]. The primary goal of this process is to create a compact model that can achieve a performance level comparable to its larger counterpart while demanding significantly fewer computational resources [3].

At its core, knowledge distillation involves training the student model to mimic the behavior of the teacher model. This is often accomplished by using the teacher's "soft targets" – the probability distributions it assigns to different possible outcomes – rather than just the "hard targets," which are the definitive ground truth labels [4]. These soft targets provide a richer and more nuanced learning signal for the student, conveying not only what the correct answer is but also the teacher's level of confidence in other potential answers and its underlying reasoning to some extent.

The adoption of knowledge distillation offers several key advantages. Firstly, it enables **model compression**, resulting in a significantly smaller model size [3]. This reduction in size makes the model easier to deploy on devices with limited storage and memory capacity, such as mobile phones or embedded systems. Secondly, distilled models often exhibit **inference speedup** [3]. The smaller size and reduced computational demands of the student model translate to faster processing times and lower latency, which is particularly crucial for real-time applications like chatbots and interactive AI systems. Thirdly, utilizing smaller models leads to **lower computational costs** [3]. Running these models requires less energy and fewer computational resources, resulting in reduced operational expenses and making advanced AI more accessible. Interestingly, in some instances, a carefully distilled student model can even achieve **potential performance improvements** over its teacher, especially when trained on

meticulously prepared distilled data [4]. This can occur because the distillation process can act as a form of regularization, helping the student model to generalize better and avoid overfitting to the training data.

Various techniques and strategies exist for implementing knowledge distillation. **Response-based distillation** is perhaps the most common approach, where the student model is trained to directly match the final output logits (the raw, unnormalized predictions) of the teacher model [40]. This method focuses on replicating the teacher's ultimate predictions for given inputs. **Feature-based distillation**, on the other hand, involves transferring knowledge from the intermediate layers of the teacher model to corresponding layers in the student [15]. This allows the student to learn more complex representations and potentially mimic the teacher's internal reasoning processes more closely. In situations where the original training data for the teacher is not available, **data-free knowledge distillation** can be employed. Here, the teacher model is used to generate synthetic data, which is then used to train the student [15]. **Multi-teacher distillation** leverages the knowledge of multiple teacher models to train a single student [42]. This approach can lead to a more robust and well-rounded student by combining the strengths of different expert models. Finally, **self-distillation** is a unique scenario where a single model acts as both the teacher and the student, often in iterative training stages where the model learns from its own predictions at different points in training [11].

Despite its numerous benefits, knowledge distillation also presents certain challenges and limitations. One primary concern is the potential **loss of information** during the transfer process [4]. A smaller student model may not have the capacity to fully capture all the nuances and complexities learned by the larger teacher. This can sometimes lead to **generalization issues**, where the student model might not perform as well as the teacher on a wide range of tasks or in diverse domains [42]. Furthermore, achieving effective knowledge transfer often requires careful **hyperparameter tuning** [42]. Parameters such as the temperature used to soften the teacher's probability distributions and the weights assigned to different loss terms need to be appropriately adjusted to optimize the learning process. The performance of the student is also inherently dependent on the **quality of the teacher model** [4]. A teacher model with flaws or biases can inadvertently transfer these shortcomings to the student. Lastly, while the resulting student model is more efficient, the distillation process itself can add to the **increased training complexity** [42], as it involves training both the teacher and the student models.

IV. Navigating Model Size: Parameter Count Considerations

The size of a language model, often quantified by the number of parameters it contains, plays a critical role in determining its capabilities and resource requirements. Generally, **larger models with more parameters** possess a greater capacity to learn intricate patterns and achieve superior performance across various natural language processing tasks [7]. For instance, GPT-J with 6 billion parameters demonstrates strong performance on many language tasks, suggesting that models with even larger parameter counts can potentially achieve even higher levels of proficiency [7]. The performance of Google's Gemma models also tends to improve as the number of parameters increases [8]. Phi-4, with its 14 billion parameters, is another example of a model designed for complex reasoning tasks [9]. Gemma 3 models are available in sizes ranging from 1 billion to 27 billion parameters, offering a spectrum of capabilities [10]. This increased capacity often translates to enhanced abilities in reasoning, understanding context, and generating coherent and relevant text.

However, the relationship between model size and performance is not always strictly linear. Smaller, well-trained models can sometimes outperform their larger counterparts on specific tasks, particularly after undergoing fine-tuning or knowledge distillation [3]. Research indicates that smaller models, when fine-tuned on high-quality labeled data, can even surpass the performance of zero-shot and few-shot learning with much larger models like GPT-4 [3]. MiniLLM, in some experiments, has even shown the ability to outperform its teacher models [4]. Furthermore, distilled models like those derived from DeepSeek-R1 have achieved impressive results on reasoning benchmarks, sometimes exceeding the performance of larger, non-reasoning focused models [47]. These findings underscore the importance of training data quality and the effectiveness of the training methodology, suggesting that model size alone is not the sole determinant of performance.

The size of a model has significant implications for several factors. **Training time** is directly affected, with larger models requiring substantially more computational resources and a longer duration to train [9]. For example, the training of Phi-4, with its 14 billion parameters, likely consumed a considerable amount of time [9]. Training large LLMs like GPT-3 and LLaMA can take weeks or even months, often necessitating the use of large clusters of powerful hardware [12]. Similarly, **computational resources** are heavily influenced by model size. Larger models demand more memory, both in terms of RAM and GPU/TPU memory, as well as greater processing power for both the training and inference phases [5]. Running inference with models like GPT-J can require specialized hardware like IPUs [7]. Deploying and running very large models locally, such as QwQ-32B or the larger versions of DeepSeek-R1, often requires high-end GPU setups [50]. Consequently, the **deployment feasibility** of a model is also tied to its size.

Smaller models are generally easier and more cost-effective to deploy, especially in environments with limited resources, such as edge devices or mobile applications [3]. Knowledge distillation plays a crucial role in enabling the deployment of high-performing models in scenarios where the deployment of their larger counterparts would be impractical.

When considering the appropriate parameter scale for a text-to-text generation model, several factors should be taken into account. For initial experimentation and for tasks with less stringent performance requirements, models in the range of 1 billion to 10 billion parameters, such as GPT-J (6B) or the smaller variants of Gemma 3 (1B-4B), can offer a favorable balance between performance and resource efficiency [7]. Starting with a smaller model allows for faster iteration and a lower initial investment in computational infrastructure. However, for more complex tasks demanding state-of-the-art performance, models with tens to hundreds of billions of parameters, like DeepSeek-R1 (32B) or Gemma 3 (27B), might be necessary [8]. It is important to recognize that these larger models come with significantly higher resource demands. In the context of knowledge distillation, the size of the student model should be considerably smaller than that of the teacher model to realize the benefits of compression and efficiency. However, the student model must still possess sufficient capacity to effectively learn and retain the essential knowledge being transferred. For example, the distillation of a 405 billion parameter Llama model to an 8 billion parameter version demonstrates a significant reduction in size while aiming to preserve performance [17].

V. Strategic Selection of Teacher Models for Distillation

The choice of the teacher model is a critical decision in the knowledge distillation process, as the student model's potential is inherently linked to the capabilities and characteristics of the teacher. Several key criteria should guide the selection of an appropriate teacher model.

Firstly, the **performance on the target task** is paramount [3]. The teacher model should demonstrate strong proficiency in the specific text-to-text generation task for which the student model is being trained. The student model's ability to learn and excel in the desired task is fundamentally limited by the teacher's own expertise. Secondly, the **availability and accessibility** of the teacher model are important practical considerations [7]. The teacher model should be readily available, whether as an open-source model that can be downloaded and used, or through an API that allows for querying and obtaining its outputs. Access to the teacher model's outputs is essential for generating the training data for the student, and access to its internal

states can be beneficial for more advanced distillation techniques. Thirdly, while not always strictly necessary, **architectural compatibility** between the teacher and student models can be advantageous, particularly when using logit-based distillation [60]. If both models share a similar underlying architecture, the transfer of knowledge, especially at the logit level, can be more straightforward. Fourthly, it is crucial to consider the **licensing and usage restrictions** associated with the teacher model [4]. Some model providers may have terms of service that prohibit the use of their models' outputs for training potentially competitive models. Finally, the **quality of the training data** used to train the teacher model indirectly influences the knowledge it can impart [3]. Teacher models that have been trained on high-quality, diverse datasets are more likely to possess a broad and robust understanding of language, leading to more valuable knowledge transfer to the student.

Several open-source models present themselves as potential teacher models. **GPT-J (6B)** stands out as a well-performing and openly accessible alternative to GPT-3 [7]. The **Gemma series** from Google, with models ranging from 1 billion to 27 billion parameters, offers a variety of options, with the larger 27B variant being a strong candidate for teaching smaller models in the series [8]. **DeepSeek-R1**, available in various sizes, is recognized for its robust reasoning capabilities, making its larger versions (e.g., 32B, 70B) suitable for distilling reasoning abilities into smaller models [47]. Alibaba Cloud's **Qwen series** also includes models of different scales, with the larger ones potentially serving as effective teachers [47]. Meta's **Llama series**, particularly larger versions like Llama 3.1 405B, are being used as teacher models on platforms like Azure [17].

Proprietary models accessible through APIs also offer compelling options for teacher models. **GPT-4** from OpenAI is a highly capable model that could serve as an excellent teacher, although its API terms may restrict its use for distilling knowledge into potentially competitive models [3]. Google's **Gemini** is another flagship model known for its strong performance [4]. Microsoft's **Phi series**, while consisting of relatively smaller models, with the Phi-4 14B variant being a prominent example, could potentially act as teachers for even smaller, more specialized models within the Phi family [3].

Ultimately, the selection of the most suitable teacher model will depend on the specific performance targets for the student model, the nature of the text-to-text generation task, and the resources and access available to the development team.

VI. Hardware Infrastructure for LLM Training: GPUs and TPUs

The hardware infrastructure chosen for training large language models significantly impacts the efficiency, speed, and cost of the process. GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units) are the two primary types of accelerators used for this computationally intensive task.

GPUs are characterized by their highly parallel architecture, featuring thousands of small, efficient cores that excel at general-purpose parallel processing. This makes them well-suited for the matrix multiplications and other parallel computations that are fundamental to deep learning [13]. The GPU ecosystem is mature, with extensive software support through libraries like CUDA and cuDNN from NVIDIA, as well as seamless integration with popular deep learning frameworks such as PyTorch and TensorFlow [13]. GPUs are also widely available from various vendors, including NVIDIA, AMD, and Intel, making them a readily accessible option for a broad range of users [13]. However, compared to TPUs, GPUs can sometimes be less efficient for the specific tensor operations that are highly prevalent in LLMs [13]. Additionally, the memory bandwidth of some GPUs might be lower than that of high-end TPUs, potentially creating bottlenecks for very large models [14].

TPUs, on the other hand, are custom-designed by Google specifically for machine learning workloads. Their architecture is highly optimized for tensor operations, which are the core computations in LLMs, often leading to faster training times for many LLM-related tasks [8]. TPUs typically boast higher memory bandwidth compared to many GPUs, enabling more efficient processing of the large tensors involved in training LLMs [10]. Within the Google Cloud Platform (GCP) ecosystem, TPUs offer excellent scalability, particularly through the use of TPU Pods, which allow for massive parallel training across thousands of interconnected TPU chips [7]. Furthermore, for many LLM workloads, TPUs are often more power-efficient than comparable GPUs [10]. A primary limitation of TPUs is their availability, as they are primarily accessible through Google Cloud Platform, which might restrict their use for those not already invested in that ecosystem [13]. While the ecosystem and development tools for TPUs are continually growing, they might not be as extensive as the well-established GPU ecosystem, although support for key frameworks like JAX and TensorFlow is robust [8]. In some high-end configurations, the total memory capacity of TPUs might also be lower than that of certain high-end GPUs [14].

The choice between GPUs and TPUs is influenced by several factors. The **scale of training** is a major consideration; for very large models and datasets, distributed training across multiple accelerators, whether GPUs or TPUs, is a necessity [12]. **Budget** also plays a crucial role, as the cost of GPU and TPU instances can vary, and the overall expense depends on the training duration and the number of accelerators

utilized [13]. Teams that have already invested in a particular **ecosystem and have familiarity** with its tools and libraries (e.g., NVIDIA's CUDA or Google's JAX) might have a preference for sticking with that hardware type. Finally, the **specific requirements of the model architecture** being trained might make one type of hardware more suitable or better optimized than the other.

Table 1: Comparative Analysis of GPUs and TPUs for LLM Training

Feature	GPUs	TPUs
Architecture	Parallel cores, versatile	Specialized for tensor operations
Performance for LLMs	Good, widely used	Often superior for training, especially at scale
Memory Bandwidth	High, but generally lower than high-end TPUs	Higher bandwidth
Memory Capacity	Higher total memory in some high-end options	Generally lower
Scalability	Scalable with multi-GPU setups, available on various cloud platforms	Excellent scalability in Google Cloud with TPU Pods
Availability	Widely available from multiple vendors, for consumers and businesses	Primarily accessible through Google Cloud Platform (GCP)
Ecosystem & Tools	Mature, broad support (CUDA, PyTorch, TensorFlow)	Strong support for JAX and TensorFlow, growing ecosystem
Cost	Varies by vendor and model, generally competitive	Can be cost-effective for large-scale training due to performance gains

Power Efficiency	Varies by model	Generally more power-efficient for LLM workloads
Cloud Platforms	Widely available (AWS, Azure, GCP)	Primarily Google Cloud

VII. Recommendations for Microsoft Azure

Microsoft Azure provides a robust infrastructure for training large language models, offering both GPU and TPU resources.

In terms of **GPUs**, Azure offers a range of NVIDIA GPU-based virtual machines (VMs) that are well-suited for LLM training. These include the NDv5 series, which utilizes A100 GPUs, the ND H100 v5 series, powered by H100 GPUs, and the NV-series VMs, which feature various generations of NVIDIA GPUs [7]. These offerings provide access to some of the most powerful GPUs currently available for artificial intelligence training workloads. Azure also provides access to **Google Cloud TPUs** through its partnership with Google, allowing users to leverage the specialized architecture of TPU v3 and TPU v4 for their training needs. This dual offering provides users with flexibility in choosing the hardware that best aligns with their model requirements and performance goals.

Azure also provides specific services and best practices to facilitate **model distillation**. **Azure AI Foundry** serves as a platform dedicated to model distillation, supporting a variety of teacher and student models. For instance, it supports Meta Llama 3.1 405B Instruct as a teacher and Meta Llama 3.1 8B Instruct and various Phi models as students [17]. This platform simplifies the distillation process by providing integrated tools and pre-configured environments. The **Stored Completions** feature in Azure allows for the capture and storage of input-output pairs from Azure OpenAI models like GPT-4o. This collected data can then be used as a high-quality dataset for fine-tuning smaller models through distillation [15]. To assess the effectiveness of the distillation process, **Azure OpenAI Evaluation** enables users to evaluate the performance of both the teacher and student models [15]. This comprehensive evaluation capability is crucial for iteratively refining the distilled models.

For best practices on Azure, it is recommended to leverage parameter-efficient fine-tuning techniques such as LoRA in conjunction with distillation. This can significantly reduce the memory footprint and training time required [26]. Utilizing advanced prompting techniques, such as chain-of-thought reasoning, with the

teacher model can also help in generating higher-quality synthetic data for the distillation process [19]. Finally, it is essential to adhere to robust security best practices when handling any sensitive data used during the training and distillation workflows [80].

For concrete recommendations on Azure VMs, for training smaller models (in the 1B to 10B parameter range) and for conducting initial distillation experiments, the NDv5 series featuring A100 GPUs (available with 40GB or 80GB of memory) offers a good balance of performance and cost. For training larger models (with tens of billions of parameters) or for accelerating the distillation of very large teacher models, the ND H100 v5 series, equipped with state-of-the-art H100 80GB GPUs, provides the highest level of performance. For users interested in leveraging TPUs on Azure, exploring the options available through Azure's integration with Google Cloud TPUs, considering TPU v3 for its cost-effectiveness and TPU v4 for its enhanced performance, is advisable.

VIII. Recommendations for Google Cloud

Google Cloud provides a leading infrastructure for large language model training, particularly through its specialized TPU offerings.

In terms of **GPUs**, Google Cloud offers NVIDIA GPU instances, including the powerful A100 and H100 GPUs, through its Compute Engine service. These instances provide the necessary computational power for a wide range of LLM training tasks. However, Google Cloud's strength in this domain lies in its **TPU** infrastructure. GCP offers various versions of TPUs, including TPU v3, TPU v4, TPU v5e, and the most recent TPU v5p [7]. For very large models, Google Cloud's TPU Pods enable massive parallel training across thousands of TPU chips, offering unparalleled scalability [7].

Google Cloud provides several cloud-specific services and promotes best practices for efficient LLM training. **Cloud TPUs** are custom-designed for machine learning, often providing significant speedups compared to GPUs for LLM training. Utilizing TPU Pods is highly recommended for distributed training of extremely large models [7]. Google's **JAX framework** is tightly integrated with TPUs and optimized for high-performance numerical computation, making it an ideal choice for training LLMs on this infrastructure [8]. **MaxText** is a performant and scalable JAX-based LLM implementation provided by Google, offering a ready-to-use solution for training LLMs on TPUs [48]. **Accurate Quantized Training (AQT)** allows for training with reduced numerical precision (INT8) on TPUs, which can further enhance performance and potentially improve model quality [48]. Best practices on GCP include optimizing data loading from Cloud Storage, especially for large-scale training jobs [48]. Leveraging

techniques like knowledge distillation and quantization is also encouraged to reduce model size and inference costs. Google Cloud AI Platform provides tools for managing and deploying trained models.

For concrete recommendations on GCP resources, for smaller models and initial distillation experiments, utilizing TPU v3 or TPU v4 is a good starting point. For large-scale training of models with tens to hundreds of billions of parameters, leveraging TPU v4 or the latest TPU v5e and v5p, and utilizing TPU Pods for distributed training, is highly recommended. For users who prefer GPU-based training on GCP, exploring the A100 and H100 instances available within Compute Engine is a viable option. Google Cloud offers a range of TPU versions with varying levels of performance and cost, catering to diverse LLM training requirements.

IX. Conclusion: Strategic Guidance for LLM Training and Distillation

In summary, the training of text-to-text generation models is a multifaceted endeavor that benefits from a strategic approach encompassing foundational training methods, optimization techniques like knowledge distillation, careful consideration of model size, and the selection of appropriate hardware infrastructure. Foundational methods such as pre-training and fine-tuning are essential for establishing the core capabilities of these models. Knowledge distillation offers a powerful pathway to creating more efficient and deployable models without significant sacrifices in performance. The number of parameters in a model dictates its capacity and resource demands, requiring a balanced consideration of performance needs and computational constraints. The selection of a suitable teacher model is a critical step in effective knowledge distillation. Finally, both GPUs and TPUs provide powerful acceleration for LLM training, each with their own strengths and considerations.

For users on Microsoft Azure, the platform offers a strong ecosystem for LLM distillation through services like Azure AI Foundry and Stored Completions. Its NVIDIA GPU offerings, particularly the NDv5 and ND H100 v5 series VMs, provide excellent performance for a wide range of model sizes. The integrated access to Google Cloud TPUs also offers specialized acceleration capabilities. On the other hand, Google Cloud stands out with its leading infrastructure for large-scale LLM training, featuring custom-designed TPUs and the optimized JAX framework. Leveraging TPU v3, v4, v5e, and v5p, depending on the training scale and budget, is highly recommended for those deeply invested in the Google Cloud ecosystem.

Ultimately, the optimal choice between GPUs and TPUs depends on the specific priorities and context of the user. For those prioritizing ease of access, a broad

ecosystem, and flexibility across different cloud platforms, **NVIDIA GPUs** represent a robust choice, with strong support on both Azure and Google Cloud. For users primarily within the Google Cloud ecosystem and focused on maximizing performance and cost-efficiency for large-scale LLM training, **Google Cloud TPUs** are highly recommended due to their specialized hardware and tightly integrated software stack. The selection should be guided by a careful evaluation of the specific project requirements, cloud platform preference, scale of operation, and familiarity with the respective hardware and software ecosystems.

Works cited

1. What is Text Generation? - Hugging Face, accessed on March 20, 2025, https://huggingface.co/tasks/text-generation
2. What is Text Generation? - DataCamp, accessed on March 20, 2025, https://www.datacamp.com/blog/what-is-text-generation
3. predibase/llm_distillation_playbook: Best practices for distilling large language models. - GitHub, accessed on March 20, 2025, https://github.com/predibase/llm_distillation_playbook
4. LLM distillation demystified: a complete guide | Snorkel AI, accessed on March 20, 2025, https://snorkel.ai/blog/llm-distillation-demystified-a-complete-guide/
5. Accelerating Large Language Models: Strategies for Optimizing Inference Time - Medium, accessed on March 20, 2025, https://medium.com/mobius-labs/accelerating-large-language-models-strategies-for-enhancing-your-ai-inference-speed-a77dddc33bb2
6. Model Distillation - Humanloop, accessed on March 20, 2025, https://humanloop.com/blog/model-distillation
7. Fine-tune GPT-J: a cost-effective GPT-4 alternative for many NLP tasks - Graphcore, accessed on March 20, 2025, https://www.graphcore.ai/posts/fine-tuned-gpt-j-a-cost-effective-alternative-to-gpt-4-for-nlp-tasks
8. google / gemma-3-27b-it - NVIDIA API Documentation, accessed on March 20, 2025, https://docs.api.nvidia.com/nim/reference/google-gemma-3-27b-it
9. Phi-4 quantization and inference speedup - Microsoft Community Hub, accessed on March 20, 2025, https://techcommunity.microsoft.com/blog/machinelearningblog/phi-4-quantization-and-inference-speedup/4360047
10. google/gemma-3-27b-it - Hugging Face, accessed on March 20, 2025, https://huggingface.co/google/gemma-3-27b-it
11. Step-By-Step Guide to Effective LLM Distillation for Scalable AI - Lamatic Labs, accessed on March 20, 2025, https://blog.lamatic.ai/guides/llm-distillation/
12. Azure sets a scale record in large language model training | Microsoft Azure Blog, accessed on March 20, 2025, https://azure.microsoft.com/en-us/blog/azure-sets-a-scale-record-in-large-language-model-training/

13. Understanding TPUs vs GPUs in AI: A Comprehensive Guide | DataCamp, accessed on March 20, 2025, https://www.datacamp.com/blog/tpu-vs-gpu-ai
14. GPU vs TPU for LLM Training: A Comprehensive Analysis - Incubity by Ambilio, accessed on March 20, 2025, https://incubity.ambilio.com/gpu-vs-tpu-for-llm-training-a-comprehensive-analysis/
15. Model Distillation. Making AI Models Leaner and Meaner: A... | by Naveen Krishnan | Towards AI, accessed on March 20, 2025, https://pub.towardsai.net/model-distillation-41ccb09eb312
16. How to use Azure OpenAI Service stored completions & distillation - Microsoft Learn, accessed on March 20, 2025, https://learn.microsoft.com/en-us/azure/ai-services/openai/how-to/stored-completions
17. AzureML Model Distillation - Code Samples - Microsoft Learn, accessed on March 20, 2025, https://learn.microsoft.com/en-us/samples/azure/azureml-examples/azureml-model-distillation/
18. Introducing Model Distillation in Azure OpenAI Service | Microsoft Community Hub, accessed on March 20, 2025, https://techcommunity.microsoft.com/blog/azure-ai-services-blog/introducing-model-distillation-in-azure-openai-service/4298627
19. Distillation in Azure AI Foundry portal (preview) - Microsoft Learn, accessed on March 20, 2025, https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/concept-model-distillation
20. Fine-tuning - OpenAI API, accessed on March 20, 2025, https://platform.openai.com/docs/guides/fine-tuning
21. Fine-Tuning Large Language Models: A Comprehensive Guide - Analytics Vidhya, accessed on March 20, 2025, https://www.analyticsvidhya.com/blog/2023/08/fine-tuning-large-language-models/
22. Transfer Learning in Natural Language Processing (NLP): A Game ..., accessed on March 20, 2025, https://medium.com/@hassaanidrees7/transfer-learning-in-natural-language-processing-nlp-a-game-changer-for-ai-models-b8739274bb02
23. Transformers and Transfer Learning: Leveraging Pre-Trained Models for Quick Wins | by Hassaan Idrees | Medium, accessed on March 20, 2025, https://medium.com/@hassaanidrees7/transformers-and-transfer-learning-leveraging-pre-trained-models-for-quick-wins-99eee633948b
24. Transfer Learning in NLP - GeeksforGeeks, accessed on March 20, 2025, https://www.geeksforgeeks.org/transfer-learning-in-nlp/
25. Exploring Transfer Learning with T5: the Text-To-Text Transfer Transformer, accessed on March 20, 2025, https://research.google/blog/exploring-transfer-learning-with-t5-the-text-to-text-transfer-transformer/

26. Fine Tune Large Language Model (LLM) on a Custom Dataset with QLoRA | by Suman Das, accessed on March 20, 2025, https://dassum.medium.com/fine-tune-large-language-model-llm-on-a-custom-dataset-with-qlora-fb60abdeba07
27. How to fine-tune a large language model (LLM) | Generative-AI – Weights & Biases - Wandb, accessed on March 20, 2025, https://wandb.ai/byyoung3/Generative-AI/reports/How-to-fine-tune-a-large-language-model-LLM---VmlldzoxMDU2NTg4Mw
28. Transfer Learning NLP|Fine Tune Bert For Text Classification - Analytics Vidhya, accessed on March 20, 2025, https://www.analyticsvidhya.com/blog/2020/07/transfer-learning-for-nlp-fine-tuning-bert-for-text-classification/
29. Fine-Tuning LLMs: A Guide With Examples - DataCamp, accessed on March 20, 2025, https://www.datacamp.com/tutorial/fine-tuning-large-language-models
30. Adapting Knowledge for Few-shot Table-to-Text Generation - arXiv, accessed on March 20, 2025, https://arxiv.org/pdf/2302.12468
31. Few-Shot and Zero-Shot Learning in LLMs: Unlocking Cross ..., accessed on March 20, 2025, https://medium.com/@anicomanesh/mastering-few-shot-and-zero-shot-learning-in-llms-a-deep-dive-into-cross-domain-generalization-b33f779f5259
32. Few-shot Learning Text Generation | Restackio, accessed on March 20, 2025, https://www.restack.io/p/few-shot-learning-answer-text-generation-cat-ai
33. Step-by-Step Guide to Mastering Few-Shot Learning | by Wiem Souai | UBIAI NLP | Medium, accessed on March 20, 2025, https://medium.com/ubiai-nlp/step-by-step-guide-to-mastering-few-shot-learning-a673054167a0
34. Seeking Advice on Training a Model for Multi-Task Text Generation (Translation + Writing Assistance) : r/unsloth - Reddit, accessed on March 20, 2025, https://www.reddit.com/r/unsloth/comments/1iqjo64/seeking_advice_on_training_a_model_for_multitask/
35. MT2ST: Adaptive Multi-Task to Single-Task Learning - arXiv, accessed on March 20, 2025, https://arxiv.org/html/2406.18038v3
36. Learning Easily Updated General Purpose Text Representations with Adaptable Task-Specific Prefix | OpenReview, accessed on March 20, 2025, https://openreview.net/forum?id=XjwNxSE0v8
37. Seeking Advice on Training a Model for Multi-Task Text Generation (Translation + Writing Assistance) : r/MLQuestions - Reddit, accessed on March 20, 2025, https://www.reddit.com/r/MLQuestions/comments/1iqjppb/seeking_advice_on_training_a_model_for_multitask/
38. Novel Methods For Text Generation Using Adversarial Learning & Autoencoders, accessed on March 20, 2025, https://www.topbots.com/ai-research-gan-vae-text-generation/
39. Alternative Ai Models For Text Generation - Restack, accessed on March 20, 2025, https://www.restack.io/p/ai-text-generation-answer-alternative-ai-models-cat-ai

40. What is Knowledge distillation? | IBM, accessed on March 20, 2025, https://www.ibm.com/think/topics/knowledge-distillation
41. What Is Model Distillation? | Built In, accessed on March 20, 2025, https://builtin.com/artificial-intelligence/model-distillation
42. Everything You Need to Know about Knowledge Distillation - Hugging Face, accessed on March 20, 2025, https://huggingface.co/blog/Kseniase/kd
43. LLM Distillation: The Key to Efficient AI Models | by Piyush Kashyap | Feb, 2025 | Medium, accessed on March 20, 2025, https://medium.com/@piyushkashyap045/llm-distillation-the-key-to-efficient-ai-models-cb4026a655bf
44. LLM Distillation Explained: Applications, Implementation & More - DataCamp, accessed on March 20, 2025, https://www.datacamp.com/blog/distillation-llm
45. MiniLLM: Knowledge Distillation of Large Language Models - arXiv, accessed on March 20, 2025, https://arxiv.org/html/2306.08543v3
46. A Detailed Technical Comparison of Fine-Tuning and Distillation in Large Language Models, accessed on March 20, 2025, https://medium.com/@jsmith0475/a-detailed-technical-comparison-of-fine-tuning-and-distillation-in-large-language-models-cccbe629dcba
47. What are DeepSeek-R1 distilled models? | by Mehul Gupta | Data Science in your pocket | Jan, 2025 | Medium, accessed on March 20, 2025, https://medium.com/data-science-in-your-pocket/what-are-deepseek-r1-distilled-models-329629968d5d
48. the world's largest distributed LLM training job on TPU v5e | Google Cloud Blog, accessed on March 20, 2025, https://cloud.google.com/blog/products/compute/the-worlds-largest-distributed-llm-training-job-on-tpu-v5e
49. Large Language Models — the hardware connection | APNIC Blog, accessed on March 20, 2025, https://blog.apnic.net/2023/08/10/large-language-models-the-hardware-connection/
50. QwQ-32B: Features, Access, DeepSeek-R1 Comparison & More | DataCamp, accessed on March 20, 2025, https://www.datacamp.com/blog/qwq-32b
51. Got DeepSeek R1 running locally - Full setup guide and my personal review (Free OpenAI o1 alternative that runs locally??) : r/selfhosted - Reddit, accessed on March 20, 2025, https://www.reddit.com/r/selfhosted/comments/1i6ggyh/got_deepseek_r1_running_locally_full_setup_guide/
52. Why Are All Local AI Models So Bad? No One Talks About This! : r/ollama - Reddit, accessed on March 20, 2025, https://www.reddit.com/r/ollama/comments/1idqxto/why_are_all_local_ai_models_so_bad_no_one_talks/
53. Why enterprises should embrace LLM distillation | Snorkel AI, accessed on March 20, 2025, https://snorkel.ai/blog/why-enterprises-should-embrace-llm-distillation/
54. Huaxin Securities: Alibaba Cloud QWQ-32B makes its global debut, and the open-source model has entered the phase of commercial value release. -

Moomoo, accessed on March 20, 2025, https://www.moomoo.com/news/post/50290638/huaxin-securities-alibaba-cloud-qwq-32b-makes-its-global-debut
55. Alibaba Stock: China Has Low AI Revenue Compared to United States - IO Fund, accessed on March 20, 2025, https://io-fund.com/artificial-intelligence/ai-platforms/alibaba-stock-china-low-ai-revenue-vs-us
56. davanstrien/models_with_metadata_and_summaries · Datasets at Hugging Face, accessed on March 20, 2025, https://huggingface.co/datasets/davanstrien/models_with_metadata_and_summaries
57. Gemma 3 - GGUFs + recommended settings : r/LocalLLaMA - Reddit, accessed on March 20, 2025, https://www.reddit.com/r/LocalLLaMA/comments/1j9hsfc/gemma_3_ggufs_recommended_settings/
58. Maximize Your AI Model Performance: Evaluating Distilled Models with Azure AI Evaluation SDK, accessed on March 20, 2025, https://techcommunity.microsoft.com/blog/aiplatformblog/the-future-of-ai-maximize-your-fine-tuned-model-performance-with-the-new-azure-a/4284292
59. Knowledge Distillation Using Frontier Open-Source LLMs: Generalizability and the Role of Synthetic Data - arXiv, accessed on March 20, 2025, https://arxiv.org/html/2410.18588v1
60. Syed-Hasan-8503/Gemma-2-2b-it-distilled - Hugging Face, accessed on March 20, 2025, https://huggingface.co/Syed-Hasan-8503/Gemma-2-2b-it-distilled
61. What Is Google Gemma? | IBM, accessed on March 20, 2025, https://www.ibm.com/think/topics/google-gemma
62. Gemma 2, knowledge distillation, llama-agents, and more ai updates | by Nabil W | Medium, accessed on March 20, 2025, https://medium.com/@nabilw/gemma-2-knowledge-distillation-llama-agents-and-more-ai-updates-2ea4a409c1ba
63. Google claims Gemma 3 reaches 98% of DeepSeek's accuracy - using only one GPU, accessed on March 20, 2025, https://www.zdnet.com/article/google-claims-gemma-3-reaches-98-of-deepseeks-accuracy-using-only-one-gpu/
64. Introducing Gemma 3: The Developer Guide, accessed on March 20, 2025, https://developers.googleblog.com/en/introducing-gemma3/
65. Papers Explained 329: Gemma 3 - Ritvik Rastogi, accessed on March 20, 2025, https://ritvik19.medium.com/papers-explained-329-gemma-3-153803a2c591
66. Fine-tune Gemma 3 with Unsloth, accessed on March 20, 2025, https://unsloth.ai/blog/gemma3
67. Towards Widening The Distillation Bottleneck for Reasoning Models - arXiv, accessed on March 20, 2025, https://arxiv.org/html/2503.01461v1
68. Distillation of Phi-4 on DeepSeek R1: SFT and GRPO | Microsoft Community Hub, accessed on March 20, 2025, https://techcommunity.microsoft.com/blog/machinelearningblog/distillation-of-p

hi-4-on-deepseek-r1-sft-and-grpo/4381697
69. kz919/QwQ-0.5B-Distilled-SFT - Hugging Face, accessed on March 20, 2025, https://huggingface.co/kz919/QwQ-0.5B-Distilled-SFT
70. kz919/QwQ-0.5B-Distilled - Hugging Face, accessed on March 20, 2025, https://huggingface.co/kz919/QwQ-0.5B-Distilled
71. Distilling foundation models for robust and efficient models in digital pathology - arXiv, accessed on March 20, 2025, https://arxiv.org/html/2501.16239v1
72. Microsoft Phi-4: The Revolutionary 14B Parameter Language Model | Galaxy.ai, accessed on March 20, 2025, https://galaxy.ai/youtube-summarizer/microsoft-phi-4-the-revolutionary-14b-parameter-language-model-w22WT1bgn5s
73. Phi Open Models - Small Language Models | Microsoft Azure, accessed on March 20, 2025, https://azure.microsoft.com/en-us/products/phi
74. Distillation: Turning Smaller Models into High-Performance, Cost-Effective Solutions, accessed on March 20, 2025, https://techcommunity.microsoft.com/blog/aiplatformblog/distillation-turning-smaller-models-into-high-performance-cost-effective-solutio/4355029
75. phi-4-multimodal-instruct Model by Microsoft - NVIDIA NIM APIs, accessed on March 20, 2025, https://build.nvidia.com/microsoft/phi-4-multimodal-instruct/modelcard
76. microsoft/phi-4 - Hugging Face, accessed on March 20, 2025, https://huggingface.co/microsoft/phi-4
77. Glass water still 4 l/h, Basic PH4 - Laboratory equipment - Auxilab, accessed on March 20, 2025, https://www.auxilab.es/en/laboratory-equipment/glass-water-still-4-l-h-basic-ph4/
78. What is the optimal training time for a dataset in Azure Custom Vision? - Microsoft Learn, accessed on March 20, 2025, https://learn.microsoft.com/en-us/answers/questions/2169221/what-is-the-optimal-training-time-for-a-dataset-in
79. Microsoft Azure Training | CBT Nuggets, accessed on March 20, 2025, https://www.cbtnuggets.com/it-training/microsoft-azure
80. Security Best Practices for LLM Applications in Azure - Microsoft Community Hub, accessed on March 20, 2025, https://techcommunity.microsoft.com/blog/azurearchitectureblog/security-best-practices-for-genai-applications-openai-in-azure/4027885