Fine-Tuning Small Language Models for Domain-Specific AI: An Edge AI Perspective

Rakshit Aralimatti * , Syed Abdul Gaffar Shakhadri , Kruthika KR , Kartik Angadi

Posted Date: 27 February 2025

doi: 10.20944/preprints202502.2128.v1

Keywords: Shakti; Small Language Model; Edge Device; Domain Specific Task; Performance Optimization

 doi:10.20944/preprints202502.2128.v1

Article

Fine-Tuning Small Language Models for Domain-Specific AI: An Edge AI Perspective

Rakshit Aralimatti *, Kruthika KR, Syed Abdul Gaffar Shakhadri and Kartik Basavaraj Angadi

SandLogic Technologies Pvt Ltd, India
* Correspondence: rakshit.aralimatti@sandlogic.com

Abstract: The Shakti series of 100M, 250M, and 500M models offers compact, resource-efficient language models designed for edge AI deployment. Unlike large models like GPT-3 and LLaMA that demand cloud-based infrastructure, Shakti models operate seamlessly on low-resource devices, including smartphones, smart TVs, IoT systems, drones, and low-end GPUs. They ensure minimal energy consumption, privacy-preserving computation, and real-time performance without internet dependency. Optimized for efficiency, Shakti models come in quantized versions (int8, int5, int4) for even faster, lighter execution on edge devices. The 2.5B Shakti model has demonstrated strong performance while maintaining low latency, paving the way for the smaller, highly efficient 100M, 250M, and 500M models. Built on Responsible AI principles, Shakti prioritizes fairness, transparency, and trust while mitigating risks such as bias, privacy concerns, and high carbon footprints. These models are ideal for sensitive domains like finance, healthcare, and legal services, providing cost-effective, sustainable, and scalable AI solutions with on-device data security. Each model is tailored for specific applications. Shakti-100M excels in text generation, summarization, and chatbots for IoT and mobile apps. Shakti-250M specializes in domain-specific tasks such as contract analysis and personalized financial or healthcare advice. Shakti-500M, a versatile model, enhances customer support, content creation, and virtual assistants with multilingual capabilities and long-context understanding. By decentralizing AI, the Shakti series democratizes access to intelligent, ethical, and impactful AI solutions across industries.

Keywords: Shakti; small language model; edge device; domain specific task; performance optimization

1. Introduction

The rise of Edge AI represents a transformative shift in the deployment of artificial intelligence, emphasizing decen tralized, on-device processing for improved real-time performance and data privacy. Unlike traditional cloud-based systems, Edge AI operates locally on devices like smartphones, IoT systems, wearables, and consumer electronics (e.g., refrigerators, washing machines, and air conditioners), making it an essential solution for resource-constrained environments. This growing demand for efficient, low-latency AI solutions highlights the limitations of Large Lan guage Models (LLMs) such as GPT-3 [1] and Llama [2]. Despite their benchmark-setting performance in tasks like text generation and question answering, LLMs face challenges like high computational costs, energy demands, and dependency on centralized infrastructure. These constraints have accelerated the need for Small Language Models (SLMs) capable of offering robust NLP performance while remaining practical for low-power deployments.

The Shakti-2.5B [3], released earlier, demonstrated how high-performance language models can be tailored for resource constrained environments, establishing the feasibility of deploying advanced NLP capabilities on edge devices while maintaining competitive performance against larger models. Building on this foundation, the Shakti series—featuring Shakti-100M, Shakti-250M, and Shakti-500M—delivers a suite of compact, scalable, and resource-efficient language models tailored for on-device applications. These models are available in quantized versions with int8, int5, and int4

 doi:10.20944/preprints202502.2128.v1

precision, enabling further reductions in memory footprint and computational load, making them ideal for real-time deployment on edge devices, including drones and consumer electronics. The models are built to overcome the constraints of edge-based environments through optimized architectures and processing techniques. For instance, Rotary Positional Embeddings (RoPE) [4] enhance sequential data handling across the series, with Shakti-500M additionally incorporating RoPE [4] scaling for extended context processing. Furthermore, the models implement advanced attention mechanisms: Shakti-500M uses Block Sparse Attention [5] for computational efficiency, while Shakti-250M and Shakti-100M adopt Variable Grouped Query Attention (GQA) [6] for accelerated lightweight tasks. All models support sliding window inference, enabling faster real-time processing without compromising performance.

The Shakti series owes its efficiency and accuracy to a meticulously designed training strategy that combines pre-training on diverse, large-scale text corpora with fine-tuning on curated, high-quality instruction data. These efforts are reflected in the benchmark results, where the models demonstrate competitive performance not only in standard metrics but also in Responsible AI benchmarks, highlighting their fairness, robustness, and ethical alignment. Shakti-500M leverages Reinforcement Learning from Human Feedback (RLHF) [7] to ensure contextually adaptive responses suitable for dynamic, real-world applications. Meanwhile, Shakti-250M and Shakti-100M utilize Direct Preference Optimization (DPO) [8] for task-specific optimization, catering to domain-specific workflows with efficiency. Together, these strategies underscore the role of proper training methodologies and curated data in achieving superior performance for edge applications.

Designed to address distinct use cases, each Shakti model aligns with diverse operational requirements. Shakti-100M, with its lightweight design, is optimized for general-purpose tasks such as chatbot functionalities and smart assistant applications on low-power devices. Shakti-250M specializes in industry-specific tasks within sectors like finance, healthcare, and legal services, excelling in handling specialized terminology and mid-length contexts through efficient inference techniques. At the top of the series, Shakti-500M combines multilingual support, long-context processing, and nuanced conversational capabilities, making it well-suited for high-demand use cases like customer support and multilingual virtual assistants.

By integrating advanced techniques and adhering to Responsible AI principles, the Shakti series demonstrates how SLMs can bridge the gap between state-of-the-art NLP performance and the practical requirements of real-world applications. Building on the success of Shakti-2.5B [3], these models pave the way for energy-efficient, privacy preserving, and scalable AI solutions, ensuring robust deployment across resource-constrained environments while addressing the challenges of modern AI usage in a sustainable manner.

2. Shakti-SLMs Architecture

The Shakti series of small language models, comprising Shakti-100M, Shakti-250M, and Shakti-500M, are built with optimized architectural enhancements to deliver efficient natural language processing (NLP) capabilities, specifically designed for edge devices. Each model in the series is designed with core components aimed at achieving low latency, efficient memory usage, and strong NLP performance across different devices, including mobile phones, IoT systems, and consumer electronics. The models leverage multiple techniques to ensure that they can operate effectively under resource-constrained environments, providing scalability without sacrificing accuracy or real-time performance. While each model shares core design elements as shown in Table 1, they are tailored with unique architectural enhancements suited to their respective applications and operational environments.

The attention mechanism employed in the Shakti series varies to balance memory efficiency with processing capability, inspired by recent advancements in Multi-Head and Multi-Query Attention (e.g., Mistral 7B [9], LLaMA [2]) and further advanced in Shakti2.5B [3]. Shakti-100M and Shakti-250M employ Variable Grouped Query Attention (GQA) [3,6], which enables multiple queries to utilize a single key, thereby reducing memory usage and enhancing inference speed. Shakti-500M, on the other hand, leverages Block Sparse Attention [5] for advanced memory efficiency, allowing it to

 doi:10.20944/preprints202502.2128.v1

handle higher computational loads effectively, and making it well-suited for complex, large-scale applications. Additionally, Shakti-500M supports RoPE [4] scaling for longer context processing, essential for applications requiring extended input sequences, such as document summarization.

All models integrate Rotary Positional Embeddings (RoPE) [4] for handling sequential data without a significant increase in memory requirements and utilize sliding window inference [10] to enable efficient processing of lengthy inputs by segmenting them into manageable chunks. All models employs Pre-Normalization and SiLU [11] (Sigmoid Linear Unit) activation functions to stabilize the training process. Pre-Normalization aids in preventing instability during training, while SiLU provides smooth, continuous activations that improve gradient flow, thereby boosting the model's ability to learn complex relationships within the data. This results in better performance, especially in smaller models designed for resource-constrained environments. This results in more efficient training, particularly for smaller models, and ensures better overall performance.

For training, each model incorporates Pre-training and Supervised Fine-Tuning (SFT) [12] on instruction data. Notably, Shakti-500M utilizes Reinforcement Learning from Human Feedback (RLHF) [7] for more nuanced, user-aligned responses, while Shakti-250M and Shakti-100M are optimized with Direct Preference Optimization (DPO) [8], focusing on fast, accurate, domain-specific output.

Shakti-100M is designed with a compact 100-million parameter configuration, featuring 10 layers and a model dimension of 640, making it ideal for ultra-low-power applications on edge devices like chatbots and smart assistants. The 2560 FFN dimension ensures efficient input processing. Variable GQA allows the model to share key-value pairs across queries, reducing memory usage which is ideal for lightweight, real-time applications. Sliding window inference [11] further optimizes input handling in low-resource environments. Fine-tuning through pre-training data, SFT [12], and DPO [8] aligns Shakti-100M for delivering contextually relevant outputs and easily adaptable to domain specific by further fine-tuning.

Shakti-250M features 250 million parameters, with 16 layers and a model dimension of 1024, balancing computational efficiency and task complexity for deployment on mobile devices. The expanded 4096 FFN dimension enables handling of more dynamic NLP tasks, making it well-suited for specialized industries such as finance and healthcare. Like the 100M model, Variable GQA [6] efficiently manages memory and computational load by sharing key-value pairs across multiple queries, while RoPE [4] maintains sequence information for long-context processing without increased memory demands. Sliding window inference [10] enhances input processing efficiency, and DPO [8] fine-tuning ensures domain-specific accuracy.

Shakti-500M is the series' most advanced model, with 500 million parameters, 24 layers, and a model dimension of 2048. The robust 5632 FFN dimension empowers it to handle high-demand NLP tasks across diverse domains. Block Sparse Attention [5] enhances memory efficiency for processing extended contexts, crucial for applications needing deeper, context-aware responses, such as enterprise customer support. Additionally, RoPE [4] scaling allows for effective long-context handling, preserving sequence information in complex, extended inputs. Fine-tuning through pre-training, SFT, and RLHF supports Shakti-500M's high-performance, scalable deployment.

 doi:10.20944/preprints202502.2128.v1

Table 1. Architecture Configuration of Shakti Series of Small Language Model.

Configuration	Shakti-100M	Shakti-250M	Shakti-500M
Model Parameters	100 Million	250 Million	500 Million
Layers	10	16	24
Model Dimension	640	1024	2048
FFN Dimension	2560	4096	5632
Attention Heads	16	16	16
Key/Value Heads	8	16	16
Activation Function	SiLu	SiLu	SiLu
Vocabulary Size	64000	64000	64000

3. Training and Fine-Tuning Methodologies

The Shakti model series—comprising Shakti-500M, Shakti-250M, and Shakti-100M—undergoes a structured training regimen to optimize performance across various applications. This process includes foundational pre-training, supervised fine-tuning (SFT), and preference alignment through either Reinforcement Learning from Human Feedback (RLHF) or Direct Preference Optimization (DPO). The training methodologies of Shakti-500M, Shakti-250M, and Shakti-100M are built upon the principles outlined in the Shakti-2.5B model [3] but have been adapted to meet the resource constraints and deployment requirements of the smaller variants.

3.1. Pre-Training

The pre-training process is a foundational phase in training Shakti models, designed to establish a comprehensive understanding of language patterns, grammar, and general knowledge. This phase leverages large-scale, diverse text corpora to expose the models to varied linguistic structures, ensuring adaptability across multiple domains and contexts. Using an unsupervised token prediction approach, the models learn to predict subsequent tokens in sequences, capturing linguistic nuances, contextual relationships, and semantic depth.

Shakti models, built on Transformer-based architectures like their predecessors (e.g., GPT-2 [13] and LLaMA [14]), utilize the self-attention mechanism to effectively learn dependencies between tokens, even in long sequences. While the general-purpose pre-training corpus includes sources such as Common Crawl and curated datasets, the Shakti-250M model incorporates domain-specific texts to enhance applicability in specialized fields such as healthcare, finance, and legal services. This tailored approach ensures the Shakti-250M model is better equipped to address industry-specific requirements.

A key innovation in the Shakti-500M model is the inclusion of quantization-aware training (QAT) [15]. This technique optimizes performance for low-resource devices by reducing memory consumption with low-bit representations while preserving model accuracy, making the model highly efficient for deployment in resource-constrained environments.

By combining large-scale, diverse datasets with targeted domain-specific corpora in Shakti-250M, Shakti models achieve a robust understanding of language, forming a strong foundation for specialized task adaptation. This approach ensures the models generalize effectively across multiple domains while maintaining flexibility for further fine-tuning to address specific use cases.

3.2. Supervised Fine-Tuning (SFT)

In the supervised fine-tuning [12] phase, all Shakti models are trained on instruction-specific and task-specific labeled datasets. This process enables the models to adapt their foundational language understanding to specialized applications, ensuring alignment with the unique requirements of domains such as conversational AI, finance, and healthcare. By leveraging labeled datasets, the

 doi:10.20944/preprints202502.2128.v1

models refine their outputs, improving their ability to generate accurate, contextually relevant, and domain-specific responses.

3.3. Reinforcement Learning from Human Feedback (RLHF)

The Shakti-500M model employs RLHF to fine-tune outputs based on human evaluative feedback, adjusting responses to better align with human preferences regarding relevance, coherence, and accuracy [7]. This method incorporates feedback from human evaluators to adjust the model's outputs based on criteria such as relevance, coherence, and accuracy. RLHF fine-tunes the model to better align with human preferences, enabling Shakti-500M to deliver responses suitable for complex, multi-turn interactions. This makes it an ideal choice for enterprise-level applications requiring nuanced and human-like conversational abilities.

3.4. Direct Preference Optimization (DPO)

In contrast, Shakti-250M and Shakti-100M models employ Direct Preference Optimization [8] (DPO) as a computationally efficient alternative to RLHF [7]. DPO aligns these models with user preferences while minimizing computational resource demands.This approach enables these models to achieve high domain-specific accuracy and deliver quality real-time responses, making them suitable for deployment in resource-constrained environments such as mobile devices and IoT applications. Recent studies have demonstrated that DPO can fine-tune language models to align with human preferences effectively, offering a simpler and more stable alternative to traditional RLHF method. This approach allows Shakti-250M and Shakti-100M to achieve high domain-specific accuracy and deliver high-quality, real-time responses. These attributes make them particularly suited for deployment in resource-constrained environments, such as mobile devices and IoT applications.

Shakti models have several unique advantages that distinguish them from other models in their category. Deployment flexibility is a key advantage, as Shakti models are uniquely optimized for deployment on low-resource devices, such as smartphones, IoT systems, and wearables. This differentiates them from other larger models, which require significant computational power and are less suitable for edge deployment. Additionally, the inclusion of quantization-aware training during pre-training and fine-tuning makes Shakti models more efficient for int4, int5, and int8 precision deployments, which is not a feature of many other comparative models in the same parameter range. The use of Direct Preference Optimization (DPO) allows Shakti-250M and Shakti-100M to achieve alignment with human preferences at a significantly lower computational cost compared to models relying solely on RLHF. This enables real-time application capabilities without sacrificing quality.

4. Shakti-SLMs Training Dataset Details

4.1. Shakti-100M

For Shakti-100M's foundational training, a custom dataset containing 1 trillion tokens was compiled from Common Crawl [16] and other open-source, text-rich sources. This dataset provides a broad and diverse base, enabling the model to acquire general language understanding, domain knowledge, and flexibility in handling various tasks across different domains.

In the Supervised Fine-Tuning (SFT) [12] phase, Shakti-100M is refined for instructional and conversational tasks using specialized datasets. Cosmopedia v2 [17] (39M samples) and FineWeb-Edu-Dedup [18] (220B tokens) enhance its educational domain knowledge, while Magpie-Pro-300K-Filtered-H4 [19] and OpenHermes-2.5-H4 [20] (300K and 1M samples, respectively) improve instruction-following abilities. self-oss-instruct-sc2-H4 [21] (50K samples) fine-tunes the model for code generation and self-instruction, and everyday-conversations-llama3.1-2k [22] and instruct-data-basics-smollm-H4 [23] (basic interactions) equip it for handling elementary conversational tasks.

In the DPO stage, ultrafeedback_binarized [24] (122K samples) is used for preference and reward modeling. This dataset provides prompts and model completions scored by GPT-3 [1], facilitating the model's training on human-aligned responses through preference splits. This fine-tuning step ensures

 doi:10.20944/preprints202502.2128.v1

Shakti-100M's alignment with user preferences and enhances its quality in generating desirable and human-friendly responses.

4.2. Shakti-250M

The Shakti-250M model is initially trained on a vast dataset to establish foundational language knowledge and domain-specific understanding. Pre-training incorporates both general datasets and domain-specific corpora in finance and legal fields, enabling the model to learn domain-relevant language patterns. For instance, the Custom Dataset and AIR-Bench/qa_finance_en [25] datasets provide broad general and finance-focused knowledge, while Vidhaan/LegalCitationWorthiness [26] enhances the model's legal language capabilities.

In the Supervised Fine-Tuning (SFT) stage, Shakti-250M is refined for instruction-following and specialized domain tasks in healthcare, finance, and legal applications. Healthcare datasets like lavita/medical-qa-datasets [27], ruslanmv/ai-medical-chatbot [28], and axiong/pmc_llama_instructions [29] enhance the model's performance in medical Q&A, patient-doctor dialogues, and instruction handling, building proficiency in medical terminology and conversational patterns vital for healthcare applications. For finance, datasets such as winddude/reddit_finance_43_250k [30] and Marina-C/question-answer-Subject-Finance-Instruct [31] strengthen the model's ability to manage finance-related discussions and Q&A, supporting applications in financial conversational AI. Legal datasets like umarbutler/open-australian-legal-qa [32] and mb7419/legal-advice-reddit [33] equip the model for legal question answering, summarization, and advice-based interactions, making it suitable for legal support and conversational tasks.

DPO fine-tunes the model to align its outputs with preferred behaviors in domain-specific contexts. This stage employs datasets specifically curated for financial, healthcare, and legal chatbot applications, including NickyNicky/nano_finance_200k_en_es_chatML_gemma_orpo_dpo [34] and Dhananjayg22/legal-dpo [35], to enhance the model's performance in generating domain-specific responses tailored to user preferences.

4.3. Shakti-500M

Pre-training establishes the foundational knowledge and general language understanding of the Shakti models. During this stage, the models are exposed to vast, diverse corpora to learn linguistic patterns, semantic relationships, and domain knowledge. For Shakti-500M, the TxT360 [36] Dataset was used, containing 14+ trillion tokens from deduplicated CommonCrawl [16] snapshots and other high-quality sources. For our model, we have used only 1.4 trillion out of 14+ trillion. This stage equips the model to generalize across various tasks and domains, preparing it for specialized fine-tuning stages.

Supervised Fine-Tuning (SFT) aligns the model with specific tasks by training on labeled, instruction-following data. For Shakti-500M, the Tome Dataset [37] and Infinity-Instruct [38] Dataset were employed, together comprising 8.75 million instruction samples. SFT helps the model adapt its foundational language skills to tasks like problem-solving, conversational AI, and coding, making it more responsive and accurate for instruction-based applications across domains.

RLHF tailors the model's outputs to better reflect human preferences, using feedback from annotators to reward truthfulness, helpfulness, and alignment with human expectations. Shakti-500M leverages the Openbmb UltraFeedback [24] Dataset, containing 64,000 prompts with 256,000 responses annotated by GPT-4. This fine-tuning phase ensures the model generates responses that are not only accurate but also contextually relevant and aligned with human intent, enhancing its suitability for complex conversational tasks.

5. Evaluation and Competitive Study

5.1. Comparative Performance Analysis

In this section, we compare the Shakti series models (Shakti-100M, Shakti-250M, and Shakti-500M) against other leading models in the same or larger parameter range, based on academic benchmark

 doi:10.20944/preprints202502.2128.v1

results across a variety of tasks. The comparison provides insights into the relative performance, efficiency, and suitability of the Shakti models for various real-world applications. The performance of the Shakti series models was evaluated on standard benchmarks to ensure consistency and fairness. For comparison models, results from available benchmarks were used, and for those not available, evaluations were conducted by us.

5.1.1. Popular Benchmark and Result Analysis for Shakti-100M

The Shakti-100M model demonstrates strong benchmark performance, as shown in Table 2, despite being significantly smaller than many competing models. It consistently matches or outperforms larger models in key evaluations, highlighting the effectiveness of its optimized training process on a carefully curated dataset. This approach enables the model to extract intricate patterns and generate accurate predictions, proving that size alone is not the sole determinant of performance.

A critical factor in Shakti-100M's success is the balanced size of its pre-training dataset. Models trained on datasets that are either too large or too small often struggle to achieve optimal results. With a 1T token pre-training dataset, Shakti-100M maintains this balance, delivering strong performance across diverse tasks. These results emphasize the importance of strategic data selection and curation in achieving high accuracy and efficiency in language models.

Table 2. Comparison results on academic benchmarks for Shakti-100M, Boomer-634M [39], SmolLM-135M [40], SmolLM-360M [40], and AMD-Llama-135M [41], which are in the same parameter range. Bolded values indicate the highest scores, and underlined values indicate the second highest.

Benchmark	Shakti-100M	Boomer-634M	SmolLM-135M	SmolLM-360M	AMD-Llama-135M
MMLU	25.96	25.91	30.2	**34.4**	23.02
BigBenchHard	**30.12**	21.11	23	24.4	18.71
IFEval	**24.3**	22.22	15.9	19.8	22
Hellaswag	51.34	39.24	41.2	**51.8**	30.48
Anli	21.34	27.5	-	-	**30.73**
Piqa	69.2	62.57	68.4	**71.6**	64.20
OpenbookQA	**37.9**	35.76	34	37.2	30.73
Truthfulqa (MC2)	**29.2**	27.57	-	-	22.56
WinoGrande	**61.3**	50.67	51.3	52.8	50.12
ARC Easy	45.8	**62.57**	42.4	50.1	43.64
SQuAD	31.5	**57.5**	-	-	25
MedQA	**28.3**	14	11.02	12.36	15.57
GPQA	**14.9**	12.1	9.89	11	12.4
Bool Q	**29.4**	22.9	17.3	21.3	23.54
SocialQA	**23.34**	14.5	16.9	19	19.1
CommonsenseQA	**35.8**	29	32.7	35.3	22.56
Trivia QA	**15.3**	2.73	4.3	9.1	7.54
GSM8K	**9.2**	1.67	1	1.69	-
MATH	13.9	**23.38**	14	19	20.64
Humaneval	**7.8**	-	-	-	5.1

5.1.2. Popular Benchmark and Result Analysis for Shakti-250M

The Shakti-250M model demonstrates outstanding efficiency and performance, competing effectively against larger models3 like Boomer-1B [42] and Llama 3.2 1B [43]. Despite its smaller size and a more limited training dataset, it achieves impressive results across various NLP tasks. This

 doi:10.20944/preprints202502.2128.v1

strong performance highlights its ability to handle diverse language challenges while maintaining computational efficiency.

A key factor behind Shakti-250M's success is its optimized training process, which leverages clean and well-curated datasets. This approach ensures that the model captures essential linguistic patterns and nuances, enabling high accuracy even with fewer pre-training tokens. While larger models may excel in specific scenarios, Shakti-250M strikes an optimal balance between model size, efficiency, and accuracy. Its ability to compete with models trained on significantly larger datasets underscores the impact of well-structured data selection and efficient learning methodologies.

 doi:10.20944/preprints202502.2128.v1

Table 3. Comparison results on academic benchmarks for Shakti-250M, Boomer-1B [42], Boomer-634M [39], Qwen2.5-0.5B [44], SmolLM-360M [40], and Llama 3.2 1B [43]. Bolded values indicate the highest scores, and underlined values indicate the second highest.

Benchmark	Shakti-250M	Boomer-1B	Boomer-634M	Qwen2.5-0.5B	SmolLM-360M	Llama 3.2 1B
MMLU	28.98	25.92	25.23	**47.5**	34.4	32.2
BigBenchHard	13.75	28.65	21.11	20.3	24.4	**30.93**
IFEval	12.83	23.81	22.22	27.9	19.8	**59.5**
Hellaswag	29.96	31.66	34.08	**52.1**	51.8	41.2
Anli	**33.40**	32.57	27.5	26.85	-	22.56
Piqa	63.22	60.78	62.57	72.50	71.6	**80.64**
OpenbookQA	16.60	22.56	35.76	30.73	**37.2**	37
Truthfulqa (MC2)	20.69	25.69	27.57	**40.2**	-	30.7
WinoGrande	52.97	45.79	51.07	56.3	52.8	**60**
ARC Challenge	41.20	40.78	**62.57**	35.6	50.1	32.8
SQuAD	23.25	**67**	57.5	52.94	-	49.2
Trivia QA	1.68	25.25	2.73	12.5	9.1	**25.69**
GSM8K	2.35	1.5	0.91	41.6	-	**44.4**
MATH	21.71	-	**23.38**	19.5	-	-

 doi:10.20944/preprints202502.2128.v1

5.1.3. Popular Benchmark and Result Analysis for Shakti-500M

The Shakti-500M model delivers exceptional performance as shown in the Table 4 across various NLP tasks, competing effectively with both similar-sized and larger models. Its strong results stem from a well-balanced and carefully curated training dataset, allowing it to maximize the efficiency of its optimized architecture. Despite its relatively smaller size, the model consistently achieves competitive benchmark scores, demonstrating its ability to handle diverse language challenges effectively.

A key contributor to Shakti-500M's success is its emphasis on data quality and architecture optimization. By leveraging a thoughtfully curated dataset, it achieves high accuracy without relying on excessive model parameters. While larger models may have advantages in specific areas, Shakti-500M maintains an optimal balance between size and efficiency, proving that a well-structured training approach can drive strong performance across a wide range of tasks.

Table 4. Comparison results on academic benchmarks for Shakti-500M, Boomer-1B [42], Boomer-634M [39], Qwen2.5-0.5B [44], and Llama 3.2 1B [43]. Bolded values indicate the highest scores, and underlined values indicate the second highest.

Benchmark	Shakti-500M	Boomer-1B	Boomer-634M	Qwen2.5-0.5B	Llama 3.2 1B
MMLU	<u>38.90</u>	25.92	25.23	**47.5**	32.2
BigBenchHard	**33.1**	28.65	21.11	20.3	<u>30.93</u>
IFEval	<u>36.62</u>	23.81	22.22	27.9	**59.5**
Hellaswag	**68.56**	31.66	34.08	**52.1**	41.2
Anli	**40.70**	<u>32.57</u>	27.5	26.85	22.56
Piqa	<u>74.59</u>	60.78	62.57	72.50	**80.64**
Med MCQA	32.61	17.56	37.50	**42.5**	<u>37.57</u>
OpenbookQA	**39.80**	22.56	35.76	30.73	<u>37</u>
WinoGrande	<u>60.67</u>	45.79	51.07	56.3	**60**
SQuAD	**71.40**	<u>67</u>	57.5	52.94	49.2
Trivia QA	**31.11**	25.25	<u>27.6</u>	12.5	25.69
GSM8K	24.92	1.5	0.91	<u>41.6</u>	**44.4**
MATH	**31.97**	-	<u>23.38</u>	19.5	-

 doi:10.20944/preprints202502.2128.v1

5.2. Domain Specific Performance Analysis

Shakti-250M, tailored with domain-specific training on Finance, Legal, and Healthcare datasets, showcases its specialized capabilities. This section highlights its performance on domain-specific benchmarks and prompt-based evaluation for each domain, emphasizing its efficiency in handling specialized tasks compared to other models.

5.2.1. Domain Specific Benchmark Result

Shakti-250M demonstrates exceptional performance in the healthcare and finance domains as summarized in Table 5, making it a versatile model for domain-specific applications. In healthcare, it excels in tasks requiring complex medical reasoning and shows strong capabilities in understanding and applying clinical knowledge, outperforming expectations of its size. Similarly, domain-specific expertise and nuanced reasoning.

The model's compact size and efficiency make it an excellent choice for edge devices and IoT deployment in both healthcare and finance applications. Its ability to deliver reliable and accurate insights under resource-constrained environments opens possibilities for real-time medical assistance, remote diagnostics, on-device health monitoring, financial forecasting, and decision-making tools. This balance of performance, scalability, and efficiency bridges the gap between advanced AI and practical domain-specific solutions.

Table 5. Comparison results on medical and finance domain benchmarks for Shakti-250M, Phi-1.5-1.3B [45], Gemma-2B [46], and Opt-2.7B [47] models, specifically for the Medical domain.

Benchmark	Shakti-250M	Phi-1.5-1.3B	Gemma-2B	Opt-2.7B
MedQA	**41.25**	31.11	29.22	27.1
MedMCQA	**34.87**	34.31	30.22	25.63
PubMedQA	58.21	**67.8**	66.4	60.8
MMLU Professional Medicine	28.4	**29.04**	18.01	16.54
MMLU Medical genetics	31.42	**42**	28	23
MMLU College Medicine	30.45	**37.57**	31.21	24.86
MMLU College Biology	31.25	**34.03**	33.33	20.14
MMLU Clinical Knowledge	36.78	**46.04**	35.47	23.02
MMLU Anatomy	**39.42**	39.26	37.04	32.59
PatronusAI finance-bench-test	**32.2**	-	-	-
jan-hq finance-benchmark mcq	**23.1**	-	-	-

5.2.2. Prompt-Based Evaluation

Table 6 presents the performance of the Shakti-250M model across healthcare, finance, and legal domains using Answer Relevancy Scores. The model demonstrates strong domain adaptability, achieving scores of 0.85 in healthcare, 0.86 in finance, and 0.81 in the legal domain, highlighting its ability to provide contextually relevant and accurate responses. In the legal domain, it also achieves a summarization score of 0.86, reflecting its capability to generate concise summaries with moderate fidelity and coverage. Additionally, in the finance domain, the model attains an average factual accuracy score of 0.83, showcasing its ability to extract essential information while leaving some room for improvement in precision and detail.

With its lightweight 250M architecture, Shakti-250M is well-suited for real-time applications such as legal chatbots, compliance tools, and contract analysis. In healthcare, it can support tasks like patient record summarization, quick medical FAQs, and triage assistance, ensuring privacy and efficiency in sensitive environments. In finance, the model is effective for financial statement analysis, risk assessment, and personalized investment recommendations, providing reliable support

 doi:10.20944/preprints202502.2128.v1

for professionals in high-stakes decision-making scenarios. These capabilities position Shakti-250M as a robust solution for professionals seeking efficient and accurate AI-driven assistance across multiple specialized domains.

Table 6. Average Answer Relevancy score of Shakti-250M model across domains as mentioned, Summarization score for Shakti-250M model for Legal domain, Average Factual Score for Finance domain.Answer Relevancy: Answer relevancy measures the degree to which the model-generated response aligns with the expected or correct answer, reflecting its accuracy and contextual relevance.Summarization Score: Summarization Score calculates the alignment and coverage of the summary generated for the input.Factual Score: Factual Score evaluates the correctness of factual information in the model's output, measuring how well it captures and reproduces essential details from the input. A score near to 1 indicates better performance of the model in the respective task.

Domain	Average Answer Relevancy Score	Summarization Score	Factual Score
HealthCare	0.85	-	-
Legal	0.81	0.86	-
Finance	0.86	-	0.83

6. Shakti-SLMs Multilingual Capabilities

The Shakti models are designed with robust multilingual capabilities, enabling them to cater to a wide range of linguistic contexts and applications. This is achieved through a specialized tokenizer that supports multiple languages, ensuring efficient representation and processing of diverse linguistic structures. The models can be fine-tuned or aligned with data from various languages, including Indian languages such as Kannada, Hindi, Telugu, and Tamil, as well as widely spoken global languages like Spanish, French, and German. This flexibility makes the Shakti series particularly valuable in multilingual environments, where seamless language adaptation is crucial for effective communication and user engagement. By supporting such a broad linguistic spectrum, the Shakti models democratize access to AI-powered solutions across different regions, breaking language barriers and fostering inclusivity.

7. Quantization

Quantization reduces model weight precision from FP32 to lower-bit formats (int4, int5, int8), significantly improving memory efficiency and inference speed while maintaining accuracy. This optimization enables Shakti-100M, Shakti-250M, and Shakti-500M models to run efficiently on resource-constrained hardware, including mobile devices, IoT systems, and drones.

We apply advanced quantization techniques like block-wise quantization with scaling factors, converting weights into 4-bit (Q4_0, Q4_1), 5-bit (Q5_0, Q5_1), and 8-bit (Q8_0) formats. Block-wise quantization enhances precision by assigning individual scaling factors to weight blocks. Memory mapping (mmap) minimizes RAM usage by directly accessing weights from disk. Additionally, CPU-specific optimizations (e.g., AVX2, ARM NEON) accelerate inference, ensuring efficient processing for real-time applications.

7.1. Throughput Performance of Quantized Models on Different Hardware

We evaluated the throughput performance of our quantized Shakti models across various hardware platforms , including high-end GPUs, CPUs, and edge devices like the Raspberry Pi and mobile phones. Using advanced quantization techniques such as 4-bit (Q4) compression, our models achieve significant memory savings and faster inference without compromising accuracy. The results are shown in the Table 7

On high-performance hardware, the Shakti-500-Q4 model excels, delivering 583.88 tokens per second (TPS) on an NVIDIA L40s GPU (Linux-based VM with AMD EPYC 7R13 processor, 40 GB RAM), surpassing SmolLM2-360M-Q4 (281.98 TPS). On an Intel Xeon Platinum 8488C CPU (8 cores, 15 GB RAM), Shakti-500-Q4 achieves 72.02 TPS, outperforming Qwen2.5-0.5B-Q4 (45.89 TPS). The

 doi:10.20944/preprints202502.2128.v1

models also demonstrate strong performance on Apple's MacBook Pro (M3 Max, 36 GB RAM, macOS), where Shakti-250-Q4 reaches 385.00 TPS, showcasing their efficiency for general-purpose and high-performance computing.

Shakti models maintain impressive throughput on resource-constrained devices. On the Raspberry Pi 5 (ARM Cortex-A76, 8 GB RAM, Raspberry Pi OS), Shakti-500-Q4 achieves 29.54 TPS, outperforming SmolLM2-360M-Q4 (28.99 TPS). On the iPhone 14 (A15 Bionic, 6 GB RAM, iOS 18), Shakti-500-Q4 delivers 62.4 TPS, while Shakti-100-Q4 reaches 153.7 TPS, enabling real-time AI applications in low-power environments. Their efficiency in tasks like language translation, voice assistance, and text summarization highlights their scalability, making them ideal for edge AI, IoT, and mobile-based AI solutions.

Table 7. This summarises the throughput of Shakti series models and other similar models on different hardware architecture. GPU Token/sec: Tokens generated per second on GPU. CPU Token/sec: Tokens generated per second on CPU. Mac Token/sec: Tokens generated per second on Mac architecture. Raspberry Pi 5 Token/sec: Tokens generated per second on Raspberry Pi 5. Mobile Token/sec: Tokens generated per second on mobile devices.

Models	Raw Model Size	Quantized Model Size	GPU Token/sec	CPU Token/sec	Mac Token/sec	Raspberry Pi 5 Token/sec	Mobile Token/sec
Shakti-500-Q4	1.03 GB	303 MB	583.88	72.02	281.43	29.54	62.4
Shakti-250-Q4	496 MB	148 MB	816.7	156.25	385.00	48.911	88.11
Shakti-100-Q4	287 MB	126 MB	512.79	101.62	365.00	60.74	153.7
SmolLM2-360M-Q4	724 MB	271 MB	281.98	56.01	182.81	28.99	-
SmolLM2-135M-Q4	280 MB	105 MB	392.32	93.96	227.21	32.355	-
Qwen2.5-0.5B-Q4	988 MB	398 MB	319.16	45.89	173.82	18.45	-

8. Responsible AI

The Shakti models embody a strong commitment to Responsible AI principles, addressing key aspects such as fairness, transparency, and environmental sustainability. By leveraging on-device processing, these models prioritize user data privacy, minimizing the risk of exposure to security vulnerabilities inherent in cloud-based systems. The adoption of quantization techniques further reduces the carbon footprint associated with model deployment, aligning with global sustainability goals. Additionally, deliberate efforts to mitigate biases during training enhance the trustworthiness of the models across diverse applications. The Shakti series fosters equitable access to advanced technology through decentralized AI solutions, promoting inclusivity and upholding ethical AI practices.

8.1. Benchmarking on Responsible AI Datasets

Shakti models were evaluated on multiple Responsible AI benchmarking datasets to assess their fairness, bias detection, and toxicity mitigation capabilities. On the Bias Benchmark for QA (BBQ) [48] dataset, which measures biases across gender, race, and religion, Shakti-500M achieved 54.08% accuracy, outperforming Shakti-250M (50.2%). The ToxiGen [49] dataset, designed to detect implicit and explicit toxicity across 13 minority groups, saw Shakti-500M scoring 51.5% accuracy, compared to 47.5% for Shakti-250M, demonstrating its improved ability to differentiate toxic and benign statements. For implicit hate speech detection, Shakti-500M reached 69.04% accuracy on the Implicit Hate Speech dataset [50], significantly higher than Shakti-250M (63%), indicating its effectiveness in identifying subtle and masked hate speech.

On the CrowS-Pairs [51] dataset, which evaluates biases in age, disability, gender, and race through log-likelihood differences, Shakti-500M achieved a lower likelihood difference of 3.02 and a 51.9% stereotype percentage, compared to 3.11 and 52.07% for Shakti-250M. Lower values indicate reduced bias and a more equitable response to both stereotypical and non-stereotypical statements. These results, summarized in Tables 8 and 9 , highlight Shakti models' capability to mitigate biases, detect nuanced toxicity, and enhance fairness, reinforcing their alignment with Responsible AI principles.

 doi:10.20944/preprints202502.2128.v1

Table 8. Accuracy scores of the Shakti models on Bias Benchmark for QA (BBQ), ToxiGen, and Implicit Hate Speech datasets. Higher accuracy indicates the model's improved ability to mitigate biases, detect nuanced toxicity, and accurately classify implicit hate speech, showcasing alignment with Responsible AI principles.

Dataset	Accuracy of Shakti-250M	Accuracy of Shakti-500M
BBQ Average	50.2	54.08
Toxigen	47.5	51.5
ImplicitHate	63	69.04

Table 9. Evaluation of the Shakti models on the Crows-Pairs dataset using Likelihood Difference and Percentage of Stereotypes metrics. Lower values in Likelihood Difference indicate reduced preference for stereotypical over non-stereotypical sentences, while a lower Percentage of Stereotypes signifies the model's fairness and ability to minimize bias..

Model	likelyhood diff	pct stereotypes
Shakti-250M	3.11	52.07
Shakti-500M	3.02	51.9

9. Conclusions

The Shakti series of small language models represents a paradigm shift in delivering efficient, secure, and high-performance AI solutions tailored for resource-constrained environments. Designed to address the limitations of large language models, Shakti models—spanning 100M, 250M, and 500M parameters—demonstrate the potential of small language models in enabling real-time, privacy-preserving computation for edge deployment. Leveraging advanced quantization techniques such as int8, int5, and int4, these models minimize memory usage and maximize throughput, achieving exceptional token-per-second performance across diverse hardware platforms, including mobile phones, IoT devices, and GPUs. This makes them ideal for applications requiring low latency and high efficiency, ensuring scalability without compromising accuracy.

Pre-training on diverse, large-scale text corpora equips Shakti models with a deep understanding of linguistic patterns, semantic relationships, and contextual nuances, enabling them to generalize effectively across a wide range of tasks. Fine-tuning techniques, including Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO), further refine these models, aligning their outputs with specific domains and tasks. This combination of foundational pre-training and targeted fine-tuning enables the Shakti models to deliver performance comparable to larger language models with significantly fewer parameters, making them ideal for resource-constrained environments. These advantages are consistently reflected in benchmark results, where the models excel in domains such as healthcare, legal, and finance, delivering superior reasoning, factual reliability, and efficiency.

The robust architecture of Shakti models incorporates Rotary Positional Embeddings, Variable Grouped Query Attention, and Block Sparse Attention, ensuring low latency and computational efficiency. These architectural innovations optimize memory usage and enable scalability across a wide range of hardware platforms. The models are also designed to support quantization (int8, int5, and int4), allowing them to achieve high tokens-per-second throughput on devices ranging from mobile phones to GPUs. This quantization-ready design enhances their compatibility with edge devices while maintaining accuracy and performance. Together, the advanced architecture and efficient training methodologies make Shakti models a benchmark for deploying high-performing AI in resource-constrained environments.

Each model in the Shakti series is optimized for specific use cases, showcasing their versatility and adaptability. Shakti-100M, a lightweight, general-purpose model, is tailored for ultra-low-resource devices like smartwatches, consumer electronics, and IoT systems. It excels in tasks such as text

 doi:10.20944/preprints202502.2128.v1

summarization, chatbot functionalities, and context-aware assistants, making it indispensable for devices with limited computational resources. Shakti-250M is specifically designed for domain-specific applications in healthcare, legal, and finance sectors, with its ability to operate securely on-premise ensuring data privacy and preventing information leakage. This model is particularly adept at specialized tasks such as patient diagnostics, contract analysis, and financial advising, thanks to its domain-specific fine-tuning. Shakti-500M, the most advanced model in the series, balances general-purpose functionality with enhanced capabilities for complex tasks. With support for multilingual processing and long-context understanding, it is ideal for applications such as customer support chatbots, virtual assistants, and content creation, with deployment potential spanning industries like e-commerce, enterprise communication, and media.

By adhering to Responsible AI principles, Shakti models emphasize fairness, trustworthiness, and accountability. Rigorous data filtering ensures unbiased outputs, while on-device processing enhances privacy and aligns with global sustainability goals by reducing reliance on energy-intensive cloud infrastructures. Use cases across industries highlight the practical impact of Shakti models, from delivering real-time insights in healthcare to powering smart assistants in IoT devices. Their ability to operate securely and efficiently underpins their growing significance in sensitive workflows and privacy-critical environments.

In conclusion, the Shakti series exemplifies the future of edge AI, bridging the gap between state-of-the-art performance and practical deployment. With innovative architecture, efficient quantization, and specialized fine-tuning, these models redefine the capabilities of small language models, making them scalable, privacy-centric, and inclusive. By democratizing access to AI and addressing real-world challenges, the Shakti series sets a new benchmark for sustainable, impactful, and responsible AI deployment across industries.

10. Future Scope

Future developments in the Shakti series aim to enhance multilingual capabilities, particularly in underrepresented languages, to further democratize AI accessibility. Additionally, exploring more efficient training methodologies, such as adaptive pre-training and task-specific finetuning, can further optimize resource consumption. Expanding support for edge computing scenarios, such as integrating Shakti models with federated learning frameworks, could provide robust solutions for collaborative and secure AI deployments. Finally, incorporating advanced feedback mechanisms, such as continuous learning from real-world usage, will improve model alignment and responsiveness in dynamic application settings.

References

1. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *arXiv* 2020, arXiv:cs.CL/2005.14165.
2. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv* 2023, arXiv:cs.CL/2302.13971.
3. Shakhadri, S.A.G.; KR, K.; Aralimatti, R. SHAKTI: A 2.5 Billion Parameter Small Language Model Optimized for Edge AI and Low-Resource Environments. *arXiv* 2024, arXiv:cs.CL/2410.11331.
4. Su, J.; Lu, Y.; Pan, S.; Wen, B.; Liu, Y. RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv* 2021, arXiv:cs.CL/2104.09864.
5. Zaheer, M.; Guruganesh, G.; Dubey, A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. Big Bird: Transformers for Longer Sequences. *arXiv* 2020, arXiv:cs.LG/2007.14062.
6. Ainslie, J.; Lee-Thorp, J.; de Jong, M.; Zemlyanskiy, Y.; Lebrón, F.; Sanghai, S. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. *arXiv* 2023, arXiv:cs.CL/2305.13245.
7. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *arXiv* 2022, arXiv:cs.CL/2203.02155.

 doi:10.20944/preprints202502.2128.v1

8. Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C.D.; Finn, C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv* 2024, arXiv:cs.LG/2305.18290.
9. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B. *arXiv* 2023, arXiv:cs.CL/2310.06825.
10. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer. *arXiv* 2020, arXiv:cs.CL/2004.05150.
11. Elfwing, S.; Uchibe, E.; Doya, K. Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. *arXiv* 2017, arXiv:cs.LG/1702.03118.
12. Cameron R. Wolfe, P. Understanding and Using Supervised Fine-Tuning (SFT) for Language Models. 2023. Available online: https://cameronrwolfe.substack.com/p/understanding-and-using-supervised,
13. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. *OpenAI* **2018**.
14. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv* 2023, arXiv:cs.CL/2302.13971.
15. Chen, M.; Shao, W.; Xu, P.; Wang, J.; Gao, P.; Zhang, K.; Luo, P. EfficientQAT: Efficient Quantization-Aware Training for Large Language Models. *arXiv* 2024, arXiv:cs.LG/2407.11062.
16. Common Crawl. Available online: https://commoncrawl.org/. (accessed on 31 December 2024).
17. Allal, L.B.; Lozhkov, A.; Penedo, G.; Wolf, T.; von Werra, L. Cosmopedia. 2024. https://huggingface.co/datasets/HuggingFaceTB/cosmopedia.
18. Skymizer AI Team. 2024. Available online: Fineweb-edu-dedup 45B. https://huggingface.co/datasets/skymizer/fineweb-edu-dedup-45B (accessed on 31 December 2024).
19. HuggingFaceTB Team. Magpie-Pro-300K-Filtered-H4. 2024. Available online: https://huggingface.co/datasets/HuggingFaceTB/Magpie-Pro-300K-Filtered-H4 (accessed on 31 December 2024).
20. HuggingFaceTB Team. OpenHermes-2.5-H4. 2024. Available online: https://huggingface.co/datasets/HuggingFaceTB/OpenHermes-2.5-H4 (accessed on 31 December 2024).
21. HuggingFaceTB Team. StarCoder2-Self-Instruct-OSS-50k. 2024. Available online: https://huggingface.co/datasets/HuggingFaceTB/self-oss-instruct-sc2-H4 (accessed on 31 December 2024).
22. HuggingFaceTB Team. Everyday Conversations LLaMA 3.1-2k. 2024. Available online: https://huggingface.co/datasets/HuggingFaceTB/everyday-conversations-llama3.1-2k (accessed on 31 December 2024).
23. HuggingFaceTB Team. Instruct-Data-Basics-smollm-H4. 2024. Available online: https://huggingface.co/datasets/HuggingFaceTB/instruct-data-basics-smollm-H4 (accessed on 31 December 2024).
24. Cui, G.; Yuan, L.; Ding, N.; Yao, G.; Zhu, W.; Ni, Y.; Xie, G.; Liu, Z.; Sun, M. UltraFeedback: Boosting Language Models with High-quality Feedback. *arXiv* 2023, arXiv:cs.CL/2310.01377.
25. AIR-Bench Team. QA Finance EN. 2024. Available online: https://huggingface.co/datasets/AIR-Bench/qa_finance_en (accessed on 31 December 2024).
26. Vidhaan Team. LegalCitationWorthiness. 2024. Available online: https://huggingface.co/datasets/Vidhaan/LegalCitationWorthiness (accessed on 31 December 2024).
27. Lavita Team. Medical QA Datasets. 2024. Available online: https://huggingface.co/datasets/lavita/medical-qa-datasets (accessed on 31 December 2024).
28. Ruslanmv Team. AI Medical Chatbot. 2024. Available online: https://huggingface.co/datasets/ruslanmv/ai-medical-chatbot (accessed on 31 December 2024).
29. Axiong Team. PMC LLaMA Instructions. 2024. Available online: https://huggingface.co/datasets/axiong/pmc_llama_instructions (accessed on 31 December 2024).
30. Winddude Team. Reddit Finance Dataset. 2024. Available online: https://huggingface.co/datasets/winddude/reddit_finance_43_250k (accessed on 31 December 2024).
31. Marina-C Team. Question-Answer Subject Finance. 2024. Available online: https://huggingface.co/datasets/Marina-C/question-answer-Subject-Finance-Instruct (accessed on 31 December 2024).
32. Butler, U. Open Australian Legal QA. 2023. Available online: https://huggingface.co/datasets/umarbutler/open-australian-legal-qa (accessed on 31 December 2024).
33. MB7419 Team. Legal Advice Reddit. 2024. Available online: https://huggingface.co/datasets/mb7419/legal-advice-reddit (accessed on 31 December 2024).
34. NickyNicky Team. Nano Finance 200k EN/ES ChatML Gemma/Orpo/Dpo. 2024. Available online: https://huggingface.co/datasets/NickyNicky/nano_finance_200k_en_es_chatML_gemma_orpo_dpo (accessed on 31 December 2024).

 doi:10.20944/preprints202502.2128.v1

35. Dhananjayg22 Team. Legal DPO. 2024. Available online: https://huggingface.co/datasets/Dhananjayg22/legal-dpo (accessed on 31 December 2024).
36. Tang, L.; Ranjan, N.; Pangarkar, O.; Liang, X.; Wang, Z.; An, L.; Rao, B.; Jin, L.; Wang, H.; Cheng, Z.; et al. TxT360: A Top-Quality LLM Pre-training Dataset Requires the Perfect Blend, 2024.
37. Arcee-AI Team. The Tome. 2024. Available online: https://huggingface.co/datasets/arcee-ai/The-Tome (accessed on 31 December 2024).
38. BAAI Team. Infinity-Instruct. 2024. Available online: https://huggingface.co/datasets/BAAI/Infinity-Instruct (accessed on 31 December 2024).
39. Bud Ecosystem. BOOMER-634M. Available online: https://huggingface.co/budecosystem/boomer-634m. (accessed on 31 December 2024).
40. Allal, L.B.; Lozhkov, A.; Bakouch, E.; von Werra, L.; Wolf, T. SmolLM - blazingly fast and remarkably powerful, 2024. (accessed on 31 December 2024).
41. AMD. AMD-Llama-135M. Available online: https://huggingface.co/amd/AMD-Llama-135m. (accessed on 31 December 2024).
42. Bud Ecosystem. BOOMER-1B. Available online: https://huggingface.co/budecosystem/boomer-1b. (accessed on 31 December 2024).
43. Meta. Llama-3.2-1B. Available online: https://huggingface.co/meta-llama/Llama-3.2-1B. (accessed on 31 December 2024).
44. Team, Q. Qwen2.5: A Party of Foundation Models. 2024. Available online: https://qwenlm.github.io/blog/qwen2.5/ (accessed on 31 December 2024).
45. Microsoft. PHI-1_5. Available online: https://huggingface.co/microsoft/phi-1_5. (accessed on 31 December 2024).
46. Google. GEMMA-2B. Available online: https://huggingface.co/google/gemma-2b. (accessed on 31 December 2024).
47. Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X.V.; et al. OPT: Open Pre-trained Transformer Language Models. *arXiv* 2022, arXiv:cs.CL/2205.01068.
48. Parrish, A.; Chen, A.; Nangia, N.; Padmakumar, V.; Phang, J.; Thompson, J.; Htut, P.M.; Bowman, S.R. BBQ: A Hand-Built Bias Benchmark for Question Answering. *arXiv* 2022, arXiv:cs.CL/2110.08193.
49. Hartvigsen, T.; Gabriel, S.; Palangi, H.; Sap, M.; Ray, D.; Kamar, E. ToxiGen: A Large-Scale Machine-Generated Dataset for Adversarial and Implicit Hate Speech Detection. *arXiv* 2022, arXiv:cs.CL/2203.09509.
50. ElSherief, M.; Ziems, C.; Muchlinski, D.; Anupindi, V.; Seybolt, J.; De Choudhury, M.; Yang, D. Latent Hatred: A Benchmark for Understanding Implicit Hate Speech. In Proceedings of the Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, 2021; pp. 345–363.
51. Nangia, N.; Vania, C.; Bhalerao, R.; Bowman, S.R. CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models. *arXiv* 2020, arXiv:cs.CL/2010.00133.