A Compact Hindi Language Model Based on Gemma-270M

Utpal Anand
3rd Year BS-MS Student
`utpal.anand@students.iiserpune.ac.in`

Supervisor: Dr. Bedartha Goswami
Machine Learning for Weather and Climate Lab
Data Science Department, IISER Pune

Semester Project 2025

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities across various natural language processing tasks. However, their deployment is often constrained by computational requirements and resource availability. Small Language Models (SLMs) with parameters ranging from 1-12B offer a compelling alternative, particularly for specialized tasks and edge deployment scenarios. This report presents the development of a compact Hindi language model based on the Gemma-270M architecture. While Gemma-270M provides multilingual support across 140 languages, its pre-training offers limited depth for Hindi-specific tasks, particularly those involving Devanagari script morphology and domain-specific text. We adapt this architecture through Hindi-focused retraining on a curated corpus of over 1 billion tokens, implementing optimized tokenization for Devanagari script with a 50k vocabulary. This work validates that even small models benefit significantly from language-specific training, enabling practical deployment in low-resource, on-device, and real-time applications.

1 Introduction

1.1 Motivation

The dominance of Large Language Models (LLMs) in natural language processing has led to remarkable advances in text generation, understanding, and reasoning. However, these models typically require substantial computational resources, high-end GPUs, and significant energy consumption, limiting their accessibility and deployment in resource-constrained environments.

Recent research has demonstrated that Small Language Models (SLMs) with fewer than 12 billion parameters are not only sufficient but often superior for specific workloads, particularly in agentic systems and Retrieval-Augmented Generation (RAG) applications. According to NVIDIA's research, SLMs represent the future of agentic AI and edge inference, offering substantial advantages in cost, latency, and energy efficiency while enabling guided decoding strategies.

For low-resource languages like Hindi, the challenge is compounded by limited high-quality training data and the complexity of Devanagari script morphology. While multilingual models provide broad coverage, they often lack the depth required for nuanced understanding of specific languages, resulting in suboptimal performance on downstream tasks.

1.2 Objectives

This project aims to develop a specialized Hindi language model by:

- Adapting the Gemma-270M architecture for Hindi-specific tasks
- Training a custom tokenizer optimized for Devanagari script
- Curating and processing a high-quality Hindi corpus
- Implementing efficient training strategies using mixed-precision computation
- Evaluating the model's performance on Hindi text generation tasks

2 Background and Related Work

2.1 Transformer Architecture

The Transformer architecture, introduced by Vaswani et al. (2017), revolutionized natural language processing by replacing recurrent mechanisms with self-attention. The core components include:

2.1.1 Self-Attention Mechanism

The attention mechanism computes weighted representations of input sequences: $\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$

where Q (queries), K (keys), and V (values) are linear projections of the input, and d_k is the dimension of the key vectors. This allows each position to attend to all positions in the previous layer.

2.1.2 Multi-Head Attention

Multi-head attention allows the model to attend to information from different representation subspaces jointly: $\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_h)W^O$,$\mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V$

2.1.3 Grouped Query Attention (GQA)

To reduce the key-value cache size, GQA shares key and value projections across multiple query heads within a group. This significantly reduces memory requirements during inference while maintaining model quality. In our implementation, all 4 attention heads share a single key-value head, effectively implementing Multi-Query Attention (MQA), which is an extreme case of GQA.

2.1.4 RMS Norm

Instead of traditional LayerNorm, modern architectures use RMSNorm (Root Mean Square Layer Normalization), which normalizes using only the root mean square statistic: $\mathrm{RMSNorm}(x) = \frac{x}{\mathrm{RMS}(x)} \cdot \gamma, \quad \mathrm{RMS}(x) = \sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}$

This is computationally more efficient than LayerNorm as it avoids computing the mean.

2.2 Small Language Models

Small Language Models represent an important paradigm shift in natural language processing. Unlike their larger counterparts, SLMs are characterized by reduced computational requirements that enable edge deployment and lower latency suited for real-time applications. Furthermore, they promote sustainable AI through decreased energy consumption, offer easier fine-tuning for specialized tasks, and generally demonstrate more transparent and interpretable behavior.

2.3 Actual Gemma-270M Architecture

Gemma-270M is Google's lightweight open model designed for efficient language understanding, featuring 270 million parameters and support for 140 languages. Built with a 32,768 token context window and a vocabulary size of 262,144 tokens, the model was pre-trained on 6 trillion tokens. Architecturally, it consists of 18 transformer layers with 4 attention heads, utilizing grouped query attention with a single key-value head. However, while impressive in scope, the multilingual nature of Gemma-270M results in limited depth for individual languages, particularly those with complex scripts like Hindi.

3 Methodology

3.1 Model Architecture used for our case

Our Hindi language model builds upon the Gemma-270M architecture with specific modifications for Hindi optimization. The architecture consists of 18 Transformer layers, reduced from typical configurations to balance performance and efficiency, with an embedding dimension of 1024. It utilizes 4 attention heads featuring grouped query attention and a single key-value head shared across query heads to reduce the KV cache. Additionally, the model employs a vocabulary size of 50,000 tokens optimized for Hindi and operates with a context window of 1024 tokens to ensure practical training efficiency.

3.1.1 Sliding-Window Attention

To reduce memory and computational bottlenecks, we implement sliding-window attention where each token attends only to the 512 tokens immediately preceding it. This mechanism:

- Dramatically reduces memory usage from $O(n^2)$ to $O(n \times w)$ where w is the window size
- Enables processing of longer sequences on limited hardware
- Behaves as full attention when sequence length < 512 tokens

The attention mask is computed as:

$$\text{mask}_{ij} = \begin{cases} 0 & \text{if } j \leq i \text{ and } i - j < 512 \\ -\infty & \text{otherwise} \end{cases}$$

3.1.2 Rotary Positional Embeddings (RoPE)

We adopt RoPE for positional encoding, which provides several advantages:

- Smooth rotational encoding based on position and dimension
- Improved stability for longer contexts

For embedding dimensions, we create pairs $(k, k+1)$ and apply rotation:

$$\begin{pmatrix} q'_k \\ q'_{k+1} \end{pmatrix} = \begin{pmatrix} \cos(m\theta_k) & -\sin(m\theta_k) \\ \sin(m\theta_k) & \cos(m\theta_k) \end{pmatrix} \begin{pmatrix} q_k \\ q_{k+1} \end{pmatrix}$$

where m is the position index and $\theta_k = 10000^{-k/d_{head}}$.

3.1.3 Feedforward Network

We employ the GeGLU (Gated Linear Unit with GELU) activation:

$$\text{GeGLU}(x) = (\text{Linear}_1(x) \cdot \text{GELU}(\text{Linear}_2(x))) \cdot \text{Linear}_3(x)$$

The feedforward dimension is 2048, providing a 2× expansion ratio from the embedding dimension.

3.2 Data Collection and Processing

3.2.1 Training Corpus

We utilize two primary datasets:

1. **Initial Training**: Hindi Wikipedia dataset containing approximately 50 million tokens
2. **Main Training**: CC100 Hindi Devanagari dataset with over 1.2 billion tokens

3.2.2 Tokenization

A custom Byte-Pair Encoding (BPE) tokenizer is trained specifically on Hindi text with:

- Vocabulary size: 50,000 tokens (reduced from 262,144)
- Optimized for Devanagari script patterns
- Better subword segmentation for Hindi morphology

3.3 Training Configuration

The model was trained with larger datasets and computational resources:

3.3.1 Final Training

- Infrastructure: 14 NVIDIA A30 GPUs
- Target iterations: 170,000 (completed: 110,000)
- Batch per GPU: 48
- Effective batch size: 1344
- Gradient accumulation: 2 steps
- Learning rate: 3×10^{-4} (3× increase)
- Warmup steps: 8,000
- Duration: 14-15 hours
- Precision: Mixed FP16/BF16

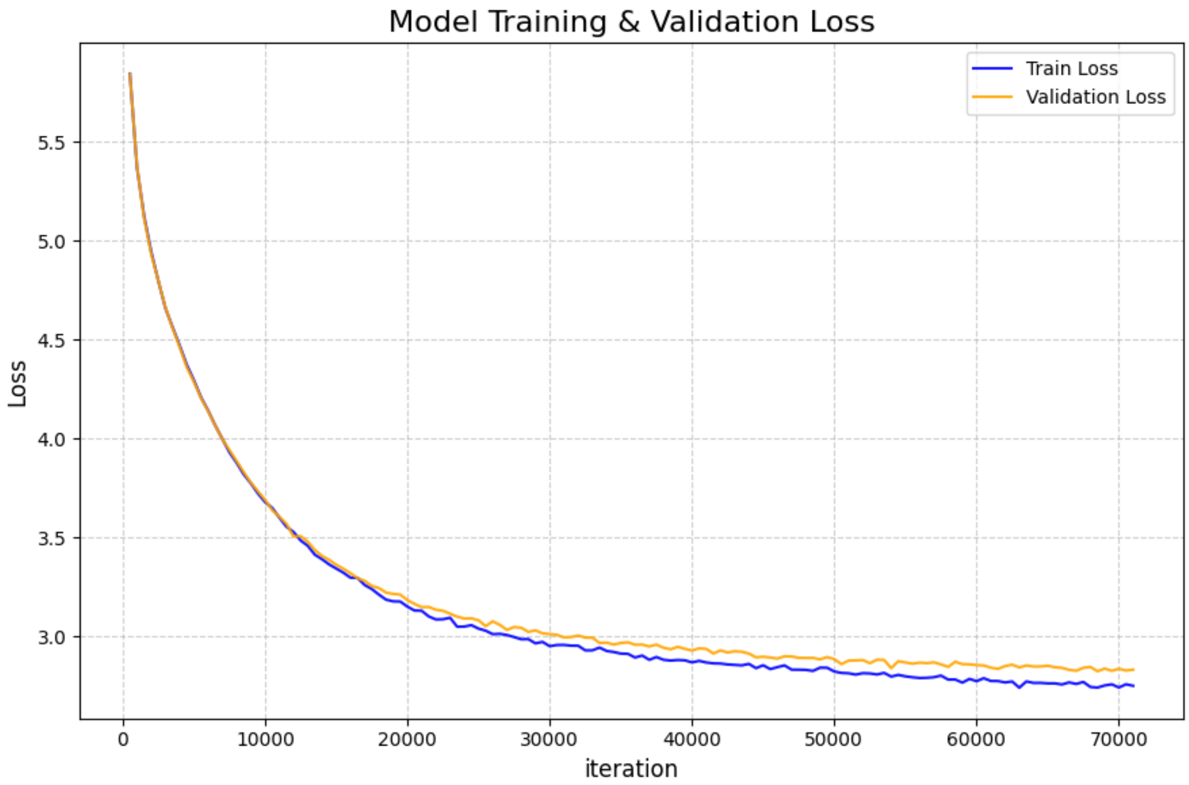

Figure 1: Loss curves

4 Results and Analysis

4.1 Training Dynamics

Due to a system crash at 110k iterations and log flushing issues, detailed loss logs after 71k iterations were unavailable. However, checkpointed models demonstrated progressive quality improvements.

4.2 Text Generation Quality

4.2.1 Early Training Issues

Initial models exhibited token repetition issues, limited contextual coherence, and difficulty maintaining narrative consistency.

4.2.2 Final Model Performance

The final model (110k iterations) demonstrated: The model demonstrates coherent, context-aware Hindi text generation with smooth narrative continuation. It maintains thematic consistency across the generated text and achieves reduced repetition through appropriate penalty tuning.

Generation parameters for optimal results: The generation parameters were configured with a temperature range of 0.7 to 0.8 and a top-p (nucleus sampling) value between 0.4 and 0.6. Additionally, a top-k of 20 was employed along with a repetition penalty of 1.2.

4.3 Inference Results

- **200 tokens generated using the earlier model (bad):**
  Prompt: भारत में बारिश कब होगी
  Generated text: । यह एक बार फिर से चीन में एक बड़ी समस्या है। चीन के एक और खेल की तरह है। भारत और ईरान के बीच भारत के साथ भारत - पाकिस्तान, चीन,जापान ओर ब्रिटेन के बीच की एक शानदार प्रदर्शन का हिस्सा , भारत के लिए सबसे बड़ा बदलाव, पाकिस्तान के साथ - साथ ऐसा किया जा रहा है कि भारत के साथ भारत में भी भारत का भविष्य निधि नहीं हुआ है। भारत ओर पाकिस्तान के बीच भी कई देशों में ही हे। भारत की सबसे बड़ी जीत, बांग्लादेश में भारत को लगा ओ यू . पी. ने कहा कि भारत के साथ काम करते हुए एक और दूसरे से बातचीत कर रहे हैं। चीन का एक नया मोड़ पर है। भारत के लिए पाकिस्तान में एक और भारतीय खिलाड़ी बनने के लिए बेहद आसान है। भारत के लिए भारत में सबसे ज्यादा मुनाफा एक ओर रिकॉर्ड बनाने के लिए सबसे ज्यादा नुकसान है। भारत ओर पाकिस्तान के बीच हे। इस समय भारतीय मूल के सबसे अमीर, चीन का भारत में सबसे अधिक है भारत की अर्थव्यवस्था के लिए सबसे ज्यादा नुकसान होगा

- **200 tokens generated using the better model (able to continue the context of the text):**
  Prompt: भारत में बारिश कब होगी।
  Generated text: इसकी संभावना कम ही है। यहां का मौसम शुष्क रहता है। यहां की नमी से लोग परेशान रहते हैं। मोसम शुष्क रहने के कारण यहां का अधिकतम तापमान 35 डिग्री तक पहुंच रहा है। इसके चलते यहां का अधिकतम तापमान 32 डिग्री से नीचे पहुंच गया है। यहां का न्यूनतम तापमान 1.5 डिग्री से नीचे दर्ज किया गया। यहां का न्यूनतम तापमान 21 डिग्री से नीचे दर्ज किया गया। वहीं अधिकतम तापमान 35 डिग्री से नीचे दर्ज किया गया है। - मोसम विभाग ने कहा - अगले 24 घंटे में कुछ बूदा बां दी हो सकती है। - मोसम विभाग ने कहा है कि अगले 24 घंटों में यहां बारिश होने की संभावना है। - वहीं, बारिश के चलते यहां की स्थिति खराब हो सकती है। - वहीं, मोसम विभाग के अनुसार अगले 24 घंटों में यहां का अधिकतम तापमान 38 डिग्री से ऊपर रहने की संभावना है। - अगर अभी मौसम साफ होता है तो, अगले 24 घंटों में यहां तापमान में बढ़ोतरी हो सकती है। - वहीं, अगले 24 घंटों में मौसम साफ रहेगा। बारिश की संभावना

4.4 Limitations

Several limitations were identified:

- **Factual accuracy**: The model struggles with factual recall, requiring additional training or fine-tuning
- **Dataset quality**: Limited preprocessing compared to commercial models
- **Context window**: Reduced from 32k to 1024 tokens limits long-form generation

5 Conclusion

This project successfully demonstrates that specialized small language models can achieve meaningful improvements for specific languages. Our compact Hindi language model, based on Gemma-270M architecture, shows fluency and contextual understanding through:

- Language-specific tokenization optimized for Devanagari script
- Curated training on over 1 billion Hindi tokens
- Efficient architecture with sliding-window attention and RoPE
- Mixed-precision training enabling resource-efficient development

With only 270 million parameters, the model maintains computational efficiency while delivering improved Hindi language understanding. This validates the viability of SLMs for language-specific applications, particularly in low-resource settings, edge deployment, and real-time inference scenarios.

The project highlights the importance of specialization in language model development and provides a foundation for future work in Hindi NLP. As small language models continue to evolve, language-specific optimization will play a crucial role in democratizing AI capabilities across diverse linguistic communities.

Acknowledgments

I would like to express my gratitude to my supervisor, Dr. Bedartha Goswami, for his guidance and support throughout this project. I also thank our institute IISER Pune for providing computational resources. Special thanks to the open-source community for making datasets and tools available for research.

References

[1] Google DeepMind, *Gemma 3 Technical Report*, arXiv:2503.19786, 2025. `https://arxiv.org/pdf/2503.19786`

[2] Google DeepMind, *Gemma Models*, `https://deepmind.google/models/gemma/`

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I., *Attention Is All You Need*, Advances in Neural Information Processing Systems 30, 2017. `https://arxiv.org/pdf/1706.03762`

[4] *Small Language Models for Agentic Systems: A Survey of Architectures, Capabilities, and Deployment Trade-offs*, arXiv:2510.03847, 2025. `https://arxiv.org/pdf/2510.03847`

[5] NVIDIA Research, *Small Language Models are the Future of Agentic AI*, arXiv:2506.02153, 2025. `https://arxiv.org/pdf/2506.02153`

[6] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, É., Ott, M., Zettlemoyer, L., and Stoyanov, V., *Unsupervised Cross-lingual Representation Learning at Scale*, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020.

[7] Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y., *RoFormer: Enhanced Transformer with Rotary Position Embedding*, arXiv:2104.09864, 2021.