Regional Tiny Stories: Using Small Models to Compare Language Learning and Tokenizer Performance

Nirvan Patil * 1 Malhar Abhay Inamdar * 1 Agnivo Gosai * 1 Guruprasad Pathak 2 Anish Joshi 2 Aryan Sagavekar 2 Anish Joshirao 2 Raj Dandekar 3 Rajat Dandekar 3 Sreedath Panat 3

Abstract

The 2023 TinyStories study developed an English dataset that allows Small Language Models (SLMs) with 1-10 million parameters to produce coherent outputs matching those of LLMs. Our research expands this framework by creating translated as well as synthetically generated datasets in Indian languages. Using this new dataset, we demonstrate that SLMs efficiently process regional languages with significantly fewer parameters than LLMs, and additionally offer a complementary framework for "inference-based evaluation" of tokenization strategies and linguistic complexity. Our analysis reveals that language-specific tokenizers outperform general-purpose ones for Indian languages. Empirical validations, supported by informationtheoretic and morphological analyses, provide insights into the superior performance of Hindi models over Marathi and Bengali. The study uncovers distinct cross-linguistic patterns: Bengali emphasizes creativity, Hindi excels in context understanding and grammar with model scaling, and Marathi requires larger models to capture its unique linguistic features. Optimal parameter allocation varies, with Hindi benefiting more from wider architectures and Bengali favoring a balanced approach. We also show that quality synthetic datasets outperform translated content for training SLMs by 15-30 \%. These findings advance both the practical application of SLMs to underserved languages and our theoretical understanding of neural language development.

1. Introduction

Recent advances in Large Language Models (LLMs) have predominantly focused on scaling architectures to multi-billion parameters (Brown et al., 2020; Chowdhery et al., 2022), driven by the generally accepted notion that increased model size directly correlates with improved performance, which relies on ever-increasing compute and data requirements (Hoffmann et al., 2022). However, Eldan & Li (2023a) challenged this paradigm through their TinyStories framework, demonstrating that Small Language Models (SLMs) with fewer than 50M parameters can achieve noteworthy performance when trained on carefully constructed but much smaller datasets. This is also observed in children who are generally exposed to no more than 100 million words by the age of 13 (Gilkerson et al., 2017), showing remarkable learning efficiency in comparison to leading LMs. By generating synthetic stories using preschool-level vocabulary through GPT-3.5 and GPT-4, the Tinystories paper established three fundamental findings: (1) coherent text generation and basic reasoning capabilities can emerge in significantly smaller architectures than previously theorized, (2) language capabilities develop hierarchically, beginning with grammatical structure and progressing through contextual consistency to creative generation, and (3) architectural choices significantly impact specific competencies, with model width correlating to knowledge retention and depth to contextual understanding. These results suggest that the field's focus on massive architectures may be unnecessary for many language modeling tasks, opening new possibilities for efficient, targeted model development.

In last two years, SLMs and modeling low-resource languages have gained traction as seen in the BabyLM challenge, proposed by Warstadt et al. (2023a), which encourages participants to focus on cognitive modeling and effective language model pre-training, keeping in mind data constraints that mirror human development. Consequently, Muckatira et al. (2024), found that smaller models trained on simplified vocabulary outperformed larger mod-

^{*}Equal contribution ¹BITS Goa, Pune Institute of Computer Technology, Independent Researcher, India ²PC COE, Pune, India ³Vizuara AI Labs, Pune, India. Correspondence to: Nirvan Patil <nirvan.ajit.patil@gmail.com>, Malhar Abhay Inamdar <malhar.inamdar.097@gmail.com>, Agnivo Gosai <agnivo2007@gmail.com>, Raj Dandekar <raj@vizuara.com>.

els trained on complete datasets at zero-shot tasks, indicating that data complexity significantly influences zero-shot capabilities in smaller models. Interestingly, Boughorbel et al. (2024), reported Tinystories inspired Arabic SLMs, where initial models trained on translated data exhibited various quality and task-specific issues, whereas further pre-training with a small amount (1%) of high-quality synthetic Arabic stories generated by GPT-4 significantly improved performance. In another recent study, Theodoropoulos et al., (2024), show that mixing high quality synthetic data with a subset of Tinystories, had modest or no improvements in the output of an LTG-BERT (Samuel et al., 2023) model when compared to outputs of GPT-Neo (Black et al., 2021) which was trained on the original Tinystories only. However, the overall performance of these LMs were capped around $\approx 70\%$, indicating a huge scope of improvement as well as illuminating the roles played by data type and quality.

While TinyStories presents compelling evidence for English language modeling with small architectures, two critical questions remain unexplored.

- First, can this modeling paradigm be extended effectively to Indian languages and is the quality of tokenizer the determining factor behind high-quality output? The development of current language models shows a significant bias towards English (Wang et al., 2024), comprising 30-60% of training data in most large-scale models. The often used BLiMP framework (Warstadt et al., 2023b) applies to linguistic knowledge evaluation for grammatical phenomenon in English.
- Second, can we leverage the TinyStories framework as a comparative tool to analyze the inherent complexities across different languages? The hypothesis being that the minimum parameter count required for effective modeling might serve as a proxy for language complexity.

To address these questions, we focus on three major Indian languages with diverse linguistic characteristics: Hindi (spoken by approximately 600-700 million people), Marathi (83-85 million speakers), and Bengali (97-100 million speakers). These languages present an ideal test case due to their significant speaker populations and distinct linguistic features. Despite their importance, there exists limited comparative analysis of their inherent complexities, particularly in the context of neural language modeling. Our work makes several key contributions:

 We demonstrate the successful adaptation of the TinyStories SLM paradigm to these three Indian languages, detailing effective pre-training steps and demonstrating substantial inference quality with significantly smaller model sizes than current state-of-the-art approaches.

- We establish a novel methodology for comparing linguistic complexity across languages using the TinyStories evaluation framework.
- We provide comprehensive analysis of tokenization efficiency across these languages, comparing standard approaches from OpenAI with specialized Indian language tokenizers like Sarvam and SUTRA.
- Consequently we provide an alternative framework for evaluating tokenizers for specific language use cases based on SLM inference quality, compared to established benchmarks like tokens/word.
- Our training data analysis shows substantial lexical differences (low BLEU ≈ 0.078) thus confirming variety essential for training, despite perceived semantic equivalence (high BERTScore ≈ 1.0), highlighting the limitation of standard metrics (ROUGE = 0) which works for English but fails for morphologically richer Indian languages.
- We show that synthetic dataset generation outperforms simple translation-based approaches, with regards to inference quality for these languages.
- Finally, we report and release our training data of $\approx 10 M$ synthetic and translated stories in three Indian languages along with the trained models for the broader community.

These findings have significant implications for both theoretical linguistics and practical applications in low-resource language modeling. Our results suggest that effective language modeling for Indian languages may not require the massive architectures currently considered standard but need task-specific quality dataset, potentially democratizing access to language technology for underrepresented languages.

2. Methodology: Data generation, training and evaluation experiments

2.1. Training data preparation

Our research extends the TinyStories framework (Eldan & Li, 2023a) to explore simple, constrained narratives in multiple Indian languages through a two-phase approach: translating the original English TinyStories dataset (Eldan & Li, 2023b) into Indian languages, followed by generating additional synthetic data using LLMs while maintaining the original methodology's constraints (Fig. 1).



Figure 1. Schematic of model pipeline. (A) Dataset prepared through machine translation as well as generation using LLM, (B) Indic tokenizers are used to preprocess the Indian language stories, (C) A decoder only transformer architecture (nanoGPT) is used to train the model in each language; Inference is evaluated by LLM on linguistic attributes

2.1.1. Translated data

Previously Doshi et al. (2024), demonstrated that machine-translated filtered data can be used to train language models for Indian languages, which can match the performance of models trained on clean native data. Likewise NLLB-3B MT has been used to translate Tiny Stories into Arabic (Boughorbel et al., 2024). Hence, using translated data for training language models is not uncommon. For dataset creation, we first translated the complete TinyStories dataset of approximately 2.0 million short stories from English to Hindi and Bengali using a combination of NLLB-200-3B and Google Translate. We verified a random subset of 1,000 stories per language using LLM to assess semantic preservation, grammatical accuracy, cultural appropriateness, and consistency in reading level.

2.1.2. SYNTHETIC DATA

We generated synthetic data by prompting GPT-4o-mini. The process began with vocabulary generation using GPT-4, creating word lists exceeding 700 entries each for nouns, verbs, and adjectives in the three languages. After filtering through GPT-3.5 to remove inappropriate content, we retained approximately 300 carefully curated words in each category. For narrative control, we generated generic features instead of specific story elements to reduce repetition patterns.

The prompt generation feature incorporated a unique identification system that combined linguistic elements systematically and prevented duplicates, successfully eliminating about 37,500 potential duplicate prompts from the 3M target dataset. After evaluating multiple models including GPT-4, LLaMA-3.1 70B, and Claude 3.5 Sonnet, we selected GPT-4o-mini based on its optimal balance of quality and generation efficiency, achieving an 8.5/10 average score based on story completeness, grammar, fluency, creativity, using GPT-4o as judge.

Quality assessment was performed using GPT-4 as the evaluation model, examining completeness, grammar, fluency, and creativity. The final implementation used complexity level 2+ prompts with expanded word limits, consistently producing the highest quality stories while maintaining generation efficiency. This approach yielded impressive evaluation scores averaging 8.73 across all metrics (details in Appendix D).

The final dataset includes 1.8M translated stories each in Hindi and Bengali, along with 2.2M new synthetic stories for Hindi, Bengali and Marathi. All content was standardized in JSON format with comprehensive metadata.

2.2. Training data evaluation

Our analysis of training data revealed significant limitations in traditional evaluation metrics when applied to Indian languages. As documented in Appendix E, ROUGE scores (Lin, 2004) consistently registered zero for semantically similar Bengali stories, highlighting a fundamental challenge in evaluating non-English text generation. Multiple metrics provided complementary insights: BERTScore (Zhang et al., 2020) values near 1.0 confirmed strong semantic equivalence between examples, while BLEU scores (Papineni et al., 2002) remained consistently low (averaging 0.078). METEOR (Banerjee & Lavie, 2005) offered middle-ground assessment (averaging 0.153) by recognizing synonyms and word variations.

This pattern—high semantic similarity (BERTScore) with low lexical overlap (BLEU/METEOR)—indicates our dataset contains diverse lexical expressions of similar concepts. Such divergence occurs because morphologically rich Indian languages permit extensive variation in expressing equivalent meanings. The apparent metric "anomaly" actually reveals a strength in our dataset: stories maintain semantic coherence while exhibiting rich linguistic variation, precisely the characteristics needed for robust language modeling. Rather than memorizing phrases, models learn to understand concepts expressed through diverse vocabulary and structures. This finding underscores both the challenge of evaluating Indian language generation and the benefit of our approach, which produces semantically coherent yet lexically diverse training examples that foster more generalizable language understanding.

2.3. Tokenizer, model and inference evaluation

The tokenizers were chosen specifically for Indian language modeling, for e.g. Sarvam (Sarvam, 2024) and SUTRA by TWO AI (Bendale et al., 2024) which incorporate tokens for language-specific elements and formatting, and results were compared with OpenAI's Tiktoken (OpenAI, 2024). For e.g., Sarvam-1's advanced tokenizer achieves near-English token fertility rates (1.4-2.1 tokens per word) for Indic scripts, significantly improving efficiency and performance compared to traditional multilingual LLMs that

struggle with high token fertility in Indian languages. Modern language model tokenizers differ in vocabulary size: OpenAI's Tiktoken (GPT2) uses 50,257 tokens, SUTRA has about 256,000 tokens, and Sarvam features 68,096 tokens with 4,096 reserved for future use.

We built on TinyStories using modified nanoGPT code (Karpathy, 2022), implementing decoder-only transformers with 8 attention heads at various parameter sizes. All models trained for 5001 epochs with 2.5% of data reserved for testing. The inference evaluation, across chosen linguistic attributes, was conducted by GPT-40, following the previously established LLM as a judge framework (Eldan & Li, 2023a; Boughorbel et al., 2024).

3. Results

3.1. Insights from our evaluation method

Model Architecture and Scaling Dynamics

Results for Hindi, Marathi, and Bengali models, trained using Sarvam tokenizer and synthetic data (Tables 1-3), reveal both systematic model scaling patterns and significant language-specific characteristics. The analysis demonstrates that increasing model size from $\approx 4.5 M$ to 153M parameters yields consistent performance improvements across all three languages, with the most substantial gains occurring in the range of $\approx 5 M$ to 73M parameters.

Our efficiency-performance analysis identifies an optimal configuration of 512 hidden units and 6 layers, totaling around 54M parameters. This architecture delivers strong performance across Hindi (8.158), Bengali (8.016), and Marathi (7.807) while keeping computational demands manageable. The relationship between layer depth and performance consistently shows optimal results at moderate depths, particularly with 6 layers across all languages.

Basic linguistic capabilities can be achieved with smaller models (4.46M-10M parameters), whereas complex story generation requires significantly larger models (41M+ parameters). This trend is consistent across the three languages, although Marathi typically needs 20-30% more parameters to match the performance levels of Hindi and Bengali. Beyond the optimal configuration, performance gains diminish relative to the quadratic growth in parameter count, highlighting important considerations for model design and deployment.

Cross-lingual Performance and Metric Analysis

Model evaluation loss consistently decreases across all languages, with Hindi dropping from about 1.4 to 0.5, indicating improved optimization in larger models. This trend is most pronounced in Hindi and Bengali, suggesting better learning. Architectural scaling impacts Hindi, Bengali,

and Marathi differently. Hindi models achieve the best performance (8.164 at 1024/7 configuration) with notable improvements in grammar (8.910) and fluency (8.580), and steady gains in overall score as model size increases. Bengali models show similar scaling characteristics (8.037 at 1024/7), hinting at structural similarities between these languages and indeed they belong to fusional languages. Conversely, Marathi which is an agglutinative language, requires still larger models for similar performance, indicating its linguistic features need more model capacity for effective processing.

Major observations:

- Context understanding capabilities vary significantly: Hindi and Bengali demonstrate robust improvements with scale (reaching approximately 7.7 at 1024/7), while Marathi exhibits more modest gains, pointing to distinct challenges in capturing contextual relationships.
- Grammatical competence shows the most dramatic scaling improvements across all languages, though from different baselines Hindi and Bengali achieve strong performance (approximately 8.4 at 64/6) even at modest scales, while Marathi requires larger models (512/2) to reach comparable accuracy.
- Optimal parameter allocation differs by language: Hindi models keep gaining with wider architecture (more embeddings), while Bengali models prefer a balanced approach with moderate width and depth.
- Interestingly, increasing embedding dimensions beyond 512, largely stymies the percentage improvement in attributes like context, grammar and overall score. This is most evident in Marathi and Bengali but further increase in layer depth allows some improvements.

Based on 3000 stories per model inference, evaluation metrics show both universal patterns (strong correlations between creativity-quality, grammar-quality, and completeness-fluency) and language-specific relationships (Bengali emphasizes creativity, Hindi shows weaker context-completeness links, and Marathi uniquely correlates context with grammar). More details are provided in Appendix B.2-B.4.

Emergence of Linguistic Capabilities

The development of critical capabilities follows a consistent pattern across languages, revealing a clear link between model capacity and linguistic competence. Basic grammatical competence emerges in models with 64-128

hidden units (4.46M-10M parameters), achieving grammar scores of about 8.4 for Hindi and Bengali.

At 256 hidden units (19M-27M parameters), story completion consistency improves significantly, with scores rising from 6.8 to 7.6 across languages. More advanced capabilities, such as context understanding and creativity, rely more heavily on model capacity. These begin to emerge meaningfully at 512 hidden units (41M-73M parameters), with context scores exceeding 7.5 and creativity metrics nearing 7.8. This threshold is crucial for Marathi, showing marked improvements in contextual processing only at this scale. Larger models (768-1024 hidden units, 85M-153M parameters) continue to enhance creative expression and contextual coherence, albeit with diminishing returns.

The hierarchical emergence of capabilities suggests that neural story generation follows a structured developmental pattern, similar to language acquisition, where simpler grammatical competencies precede more complex narrative abilities. This view aligns with the idea that language is acquired primarily through social interactions and pattern recognition in speech, rather than being driven by an innate, specialized "language module." Language learning emerges from general cognitive skills, such as intention-reading and pattern-finding, which underlie both grammatical and narrative development (Tomasello, 2003).

3.2. Comparing inference results of our Regional TinyStories with LLMs

Comparison with reference large language models (LLMs) provides valuable insights into the current capabilities and limitations of our approach. GPT-4 and SUTRA variants establish strong baselines across all languages, consistently achieving scores between 9.0 and 9.5. While our scaled models approach these performance levels in specific metrics, particularly grammar and fluency, a notable gap persists in context understanding and creativity. SUTRA-Pro demonstrates superior performance compared to SUTRA-Light across all languages, with marked advantages in context understanding (9.60 versus 9.20), suggesting that its architectural improvements benefit contextual processing independently of the target language.

In Figs. 2 & 3, we show the stories generated from our TinyStories 5M model and that of GPT-40 for Hindi language for the same prompt. The prompt which we gave in Hindi effectively translates to:

"Once upon a time, there was a small boy. His toy..."

We then ask GPT-40 to qualitatively compare between the 2 stories. Here is the output:

एक बार एक छोटा लड़का था. उसे खिलौना बहुत पसंद था। हर सुबह, जब सूरज आसमान में चमकता, मोहन अपने खिलौने के साथ बगीचे में जाता। वह खिलौनों के साथ खेलते हुए अपने खिलौनों के साथ एक बड़ा सा खिलौना लेकर बगीचे में खो जाता। Mnएक दिन, मोहन ने सोचा कि क्यों न अपने खिलौने से एक न्या खिलौना बनाया जाए। उसने अपने खिलौनों को इकड़ा किया और खिलौनों को इधर-उधर उड़ाया। खिलौने धीरे-धीर खिलौने को उड़ने लोगे और मोहन का मन हल्का हो गया। उसने सोचा, "पयह तो बहुत मजेदार है।" Mnजब मोहन ने खिलौना आसमान में छोड़ा, तो वह उसे अपने खिलौनों के साथ ले गया। खिलौना धीरे-धीरे खिलने लगा और खिलौने उसके खिलौने चारों ओर फैल गए। मोहन ने देखा कि खिलौना अब उनके खिलौने के खिलौनों को और भी सुंदर बना रहा था। Mnजउस दिन मोहन ने सीखा कि जब हम प्यार से कुछ करते हैं, तो वह और भी खास बन जाता है। उसने अपने खिलौनों के साथ एक नई यात्रा शुरू की।

Figure 2. Regional TinyStories 5M Hindi model generated story, story after prompt is highlighted

एक बार एक छोटा लड़का था। उसे खिलोना गाड़ी बहुत पसंद थी। उसका नाम राहुल था। राहुल के पास खिलोनों की एक छोटी सी गाड़ी थी, जो उसके दादाजी ने उसे तोहफे में दी थी। वह गाड़ी लाल रंग की थी और उसके पहिए सुनहरें थी। राहुल उसे अपना सबसे अच्छा दोस्त मानता था। रोज सुबह राहुल अपनी गाड़ी लेकर बगीचे में खेलता। वह गाड़ी को घास पर दोड़ाता, कभी-कभी उसे मिट्टी के छोटे-छोटे रास्तों पर चलाता। एक दिन खेलते-खेलते उसकी गाड़ी बगीचे के एक गड़े में गिर गई। राहुल बहुत उदास हो गया और रोने लगा। तभी, बगीचे में बैठा एक छोटा खरगोश उसकी ओर आया। खरगोश ने पूछा, "राहुल हुत उदास हो गया और रोने लगा। तभी, बगीचे में बैठा एक छोटा खरगोश उसकी ओर आया। खरगोश ने पूछा, "राहुल हुत तुम करों रो रहे हो?" खरगोश ने पूछा, "राहुत हुत सुम करों हो?" खरगोश भें पाड़ी गड़े में गिर गई है।" खरगोश अरे बाला, "चिंता मत करो, में दुम्हारी मदद करना।" बरगोश ने पाड़े में एक लंबी लकड़ी तोड़ी। उसने राहुल से कहा, "आओ, मिलकर गाड़ी निकालते हैं।" राहुल ने और खरगोश ने मिलकर लकड़ी का इस्तेमाल किया और राहुल के कहाने खुर हो जह स्तेमाल किया और उसे गाजर खाने को दी।उस दिन के बाद राहुल और खरगोश बहुत अच्छे दोस्त बन गए। अब वे हर दिन साथ खेलते और बगीचे में नई-नई जगहें खोजते। राहुल ने सीखा कि दोस्ती और मदद करना सबसे बड़ा खजाना है।इस तरह, राहुल और बरगोश की दोसती की कहानी पूर गाँव में मशहर हो गई।

Figure 3. GPT40 generated Hindi story

Both stories follow a similar narrative structure, focusing on a young boy who loves playing with toys. Both stories conclude with a positive resolution, emphasizing companionship and the joy of forming new relationships. Both are written in simple, accessible Hindi, making them suitable for children.

This demonstrates that despite having a parameter count nearly 1,000,000 times lower than GPT-40, we successfully generate coherent and fluent stories with clear messages. Appendix C includes SLM vs LLM generated Marathi and Bengali story results.

3.3. Regional tokenizers perform better

Table 4 compares three tokenizers—Sarvam, SUTRA, and Tiktoken—across Hindi, Marathi, and Bengali using both quantitative and qualitative metrics based on stories generated by our 54M parameter models. Our analysis reveals a striking pattern: Tiktoken consistently achieves the lowest evaluation loss across all languages (Hindi: 0.149, Marathi: 0.167, Bengali: 0.135), suggesting superior perplexity minimization. However, this advantage doesn't translate to generation quality, where Tiktoken underperforms on all subjective dimensions. Indian language-specific tokenizers demonstrate superior performance in generation quality. Sarvam achieves the highest overall scores for all languages (Hindi: 8.158, Marathi: 7.807, Bengali: 8.016), particularly excelling in context understanding and narrative completeness. SUTRA follows closely, with strengths in grammatical accuracy.

The performance gap is most pronounced in context awareness (+0.56 points average for Sarvam over Tiktoken) and

| Hidden Size | Layer | Model Size | Eval Loss | Context | Completeness | Creativity | Fluency | Grammar | Overall |
|-------------|-------|------------|-----------|---------|--------------|------------|---------|---------|---------|
| 64 | 2 | 4.46 | 1.408 | 5.665 | 6.826 | 7.217 | 7.472 | 7.969 | 7.030 |
| 64 | 6 | 4.65 | 1.182 | 6.412 | 7.122 | 7.314 | 7.901 | 8.446 | 7.439 |
| 64 | 12 | 5.00 | 1.057 | 6.374 | 7.227 | 7.390 | 7.959 | 8.450 | 7.480 |
| 512 | 2 | 41.00 | 0.654 | 7.054 | 7.661 | 7.705 | 8.427 | 8.746 | 7.919 |
| 512 | 6 | 54.00 | 0.518 | 7.734 | 7.783 | 7.806 | 8.554 | 8.912 | 8.158 |
| 512 | 12 | 73.00 | 0.519 | 7.572 | 7.659 | 7.718 | 8.458 | 8.862 | 8.054 |
| 1024 | 2 | 94.00 | 0.581 | 7.344 | 7.798 | 7.829 | 8.516 | 8.825 | 8.062 |
| 1024 | 7 | 153.00 | 0.513 | 7.695 | 7.806 | 7.830 | 8.580 | 8.910 | 8.164 |

Table 1. **Hindi** - This table illustrates the hyperparameter configurations and evaluation results for Hindi Stories. The color coding is such that, the lighter the color, the better the performance. No. of attention heads = 8, tokenizer vocab size = 68096 (Sarvam). Mean scores across 3000 samples are reported for each model configuration.

| Hidden Size | Layer | Model Size | Eval Loss | Context | Completeness | Creativity | Fluency | Grammar | Overall |
|-------------|-------|------------|-----------|---------|--------------|------------|---------|---------|---------|
| 64 | 2 | 4.46 | 3.7298 | 5.618 | 6.615 | 7.525 | 6.823 | 7.411 | 6.799 |
| 64 | 6 | 4.65 | 2.843 | 6.171 | 6.974 | 7.435 | 7.390 | 8.103 | 7.215 |
| 64 | 12 | 5.00 | 2.6244 | 6.249 | 7.009 | 7.288 | 7.471 | 8.184 | 7.240 |
| 512 | 2 | 41.00 | 2.3330 | 6.934 | 7.396 | 7.521 | 8.002 | 8.603 | 7.691 |
| 512 | 6 | 54.00 | 2.0761 | 7.245 | 7.407 | 7.553 | 8.106 | 8.723 | 7.807 |
| 512 | 12 | 73.00 | 1.8117 | 7.281 | 7.565 | 7.664 | 8.156 | 8.739 | 7.881 |
| 1024 | 2 | 94.00 | 0.680 | 6.728 | 7.184 | 7.484 | 7.687 | 8.295 | 7.476 |
| 1024 | 7 | 153.00 | 0.619 | 7.275 | 7.152 | 7.540 | 7.896 | 8.625 | 7.698 |

Table 2. Marathi - This table illustrates the hyperparameter configurations and evaluation results for Marathi Stories. The color coding is such that, the lighter the color, the better the performance. No. of attention heads = 8, tokenizer vocab size = 68096 (Sarvam). Mean scores for 3000 samples are reported for each model configuration.

fluency (+0.63 points average). This suggests regionally specialized tokenizers better capture semantic cohesion, idiomatic expressions, and structural nuances. These findings align with research showing general-purpose tokenizers introduce significant biases in non-English languages, requiring up to 15 times more tokens for equivalent content (Petrov et al., 2023). The superior performance of language-specific tokenizers can be attributed to several factors: (1) more efficient subword segmentation aligned with morphological boundaries, (2) better handling of script-specific features in Devanagari and Bengali scripts, and (3) vocabulary coverage optimized for the linguistic distributions of these languages. These advantages are particularly evident in the grammar scores, where both Sarvam and SUTRA demonstrate robust handling of morphological and syntactic features specific to Indian languages.

3.4. Models trained on translated data have lower inference evaluations

The 54M model with Sarvam tokenizer was chosen for inference evaluations on translated datasets and compared with those trained on synthetic data. We observe that these models have higher evaluation loss for same training duration and have poorer inference evaluation scores, suggest-

ing that translated data quality is lower compared to synthetic. For e.g., the Hindi model has an overall score of 6.39 against 8.16 observed for model trained with synthetic data (Appendix F). Reduced performance, upon using translated data, was also noticed in one previous report (Boughorbel et al., 2024). This could be explained using the following viewpoints: (1) Cultural biases: Source data culture transfers to target languages, including elements like foreign names that prevent models from generating culturally appropriate content (Holmström et al., 2023). (2) Grammatical and style issues: Languages express similar concepts with different strucutures and conventions, and often translations fail to adapt these nuances, producing unnatural text in the target language (Zhang & Toral, 2019). (3) Lastly, higher evaluation losses suggest difficulty in next token prediction which could stem from the noise introduced through the translation process, that complicates the task compared to working with original text (Boughorbel et al., 2024).

3.5. Tokenizer analysis with focus on language complexity

We use a dual-perspective approach to quantify the linguistic complexity of Hindi, Bengali, and Marathi, revealing their intrinsic features for evaluating tokenization strate-

| Hidden Size | Layer | Model Size | Eval Loss | Context | Completeness | Creativity | Fluency | Grammar | Overall |
|-------------|-------|------------|-----------|---------|--------------|------------|---------|---------|---------|
| 64 | 2 | 4.46 | 1.514 | 6.663 | 7.097 | 7.469 | 7.797 | 8.424 | 7.490 |
| 64 | 6 | 4.65 | 1.245 | 6.533 | 7.225 | 7.482 | 7.975 | 8.454 | 7.534 |
| 64 | 12 | 5.00 | 1.136 | 6.760 | 7.289 | 7.563 | 7.968 | 8.507 | 7.617 |
| 512 | 2 | 41.00 | 0.693 | 7.373 | 7.494 | 7.644 | 8.314 | 8.782 | 7.922 |
| 512 | 6 | 54.00 | 0.569 | 7.507 | 7.645 | 7.693 | 8.420 | 8.816 | 8.016 |
| 512 | 12 | 73.00 | 0.544 | 7.525 | 7.718 | 7.743 | 8.450 | 8.836 | 8.054 |
| 1024 | 2 | 95.00 | 0.609 | 7.407 | 7.470 | 7.626 | 8.293 | 8.786 | 7.916 |
| 1024 | 7 | 157.00 | 0.557 | 7.567 | 7.639 | 7.740 | 8.409 | 8.832 | 8.037 |

Table 3. **Bengali** - This table illustrates the hyperparameter configurations and evaluation results for Bengali Stories. The color coding is such that, the lighter the color, the better the performance. No. of attention heads = 8, tokenizer vocab size = 68096 (Sarvam). Mean scores for 3000 samples are reported for each model configuration.

| Tokenizer Name | Eval Loss | Context | Completeness | Creativity | Fluency | Grammar | Overall |
|----------------|-----------|---------|--------------|------------|---------|---------|---------|
| Hindi | | | | | | | |
| Sarvam | 0.518 | 7.734 | 7.783 | 7.806 | 8.554 | 8.912 | 8.158 |
| SUTRA | 0.522 | 7.548 | 7.449 | 7.584 | 8.292 | 8.875 | 7.950 |
| Tiktoken | 0.149 | 6.974 | 7.106 | 7.360 | 7.889 | 8.681 | 7.602 |
| Marathi | | | | | | | |
| Sarvam | 0.645 | 7.245 | 7.407 | 7.553 | 8.106 | 8.723 | 7.807 |
| SUTRA | 0.627 | 7.523 | 7.162 | 7.483 | 8.012 | 8.724 | 7.781 |
| Tiktoken | 0.167 | 7.014 | 6.742 | 7.137 | 7.524 | 8.451 | 7.374 |
| Bengali | | | | | | | |
| Sarvam | 0.569 | 7.507 | 7.645 | 7.693 | 8.420 | 8.816 | 8.016 |
| SUTRA | 0.608 | 7.614 | 7.374 | 7.595 | 8.212 | 8.845 | 7.928 |
| Tiktoken | 0.135 | 7.118 | 6.989 | 7.358 | 7.778 | 8.614 | 7.572 |

Table 4. Comparison of tokenizers across Hindi, Marathi, and Bengali for model with 6 layers, 8 attention heads, 512 hidden embeddings.

gies.

3.5.1. Information-Theoretic Analysis

To evaluate tokenization quality and language complexity, we computed Rényi entropy (Zouhar et al., 2023) information-theoretic measure of uncertainty and diversity in the tokenized distributions across the training corpora for each of the three languages using both Sarvam and SUTRA tokenizers. Rényi entropy provides a parameterized framework for quantifying information content in tokenized distributions, with parameter α controlling the sensitivity to rare versus common tokens. Table 5 presents our findings:

| Tokenizer | Hindi | Bengali | Marathi |
|-----------|--------|---------|---------|
| Sarvam | 6.2852 | 6.3579 | 6.5449 |
| SUTRA | 7.1530 | 7.4135 | 7.7620 |

Table 5. Rényi entropy ($\alpha=2.5$) for Hindi, Bengali, and Marathi using Sarvam and SUTRA tokenizers.

Our analysis reveals consistent patterns across tokenizers and languages. Marathi consistently exhibits the highest entropy values, which suggests that Marathi may possess a more complex morphological structure or greater variability in token-level patterns, necessitating a more diverse set of tokens for accurate representation. This could be the reason behind Marathi's overall lower evaluation scores.

The choice of tokenizer significantly influences entropy distributions. With Sarvam, we observe lower entropy values across all languages, indicating more concentrated probability mass in token distributions. This suggests Sarvam's design achieves more compact tokenization by capturing efficient subword structures for Indian languages. Conversely, SUTRA's higher entropy values point to a more diverse tokenization strategy, potentially offering richer representational capacity at the cost of increased vocabulary complexity. This could explain why models using Sarvam consistently outperform those with SUTRA.

To examine how entropy varies with the α parameter, we computed values at different levels (α = 0.5, 1.0, 2.0). At α = 0.5, which emphasizes rare tokens, Marathi showed the highest entropy (SUTRA: 10.69, Sarvam: 11.06). At α = 1.0 (Shannon entropy), the languages demonstrated moderate convergence, though Marathi maintained higher values. These consistent patterns across α values confirm robust differences in tokenization complexity among these languages.

3.5.2. MORPH SCORE

To complement our information-theoretic approach, we evaluated morphological fidelity, on the words used for analyzing Rényi entropy, using MorphScore, which quantifies alignment between tokenizer outputs and linguistic morphemes. A 'morpheme' is the smallest unit of language with meaning, serving as a basic building block for words. Following the methodology established by Arnett & Bergen (2024), we constructed morphologically-annotated evaluation sets for each language. The MorphScore results

| Language | SUTRA | Sarvam | | |
|----------|--------|--------|--|--|
| Hindi | 0.7268 | 0.7276 | | |
| Bengali | 0.3002 | 0.3194 | | |
| Marathi | 0.6671 | 0.6620 | | |

Table 6. MorphScore evaluation results comparing SUTRA and Sarvam tokenizers across three Indic languages. Higher scores indicate better alignment with morphological boundaries.

in Table 6 reveal several interesting patterns. First, Sarvam achieves marginally higher MorphScore values for Hindi and Bengali, over SUTRA, suggesting better preservation of morphological boundaries for these languages in its tokenization strategy. Second, we observe substantial variation in absolute MorphScore values across languages, with Bengali showing markedly lower scores (by 50 %) compared to Hindi and Marathi. This stark difference suggests current tokenization approaches may not optimally capture Bengali's morphological structures. According to Arnett & Bergen (2024)'s analysis a higher MorphScore indicates a better aligned tokenizer and consequently better model performance, which in our case is broken by the Bengali models outperforming Marathi. Hence, other mechanistic factors like Rényi entropy may have greater language specific impact, necessitating more research in this area.

3.5.3. Cross-Linguistic Complexity Analysis

The above analyses reveal interesting insights into language complexity. While Marathi shows higher complexity in terms of token distribution and processing requirements (highest Rényi entropy values), Bengali presents unique challenges in morphological boundary recognition (lowest MorphScore). Hindi consistently shows moderate values across both metrics, potentially explaining its relatively efficient tokenization and consequently the best performance in our model evaluations. Also, Sarvam shows consistently lower entropy and higher morphological alignment (except for Marathi) and thus better performance which is again corroborated by the inference scores.

Bafna & Žabokrtský (2022) have previously discussed Marathi being more agglutinative than Hindi, thus allow-

ing suffix stacking with boundary changes. They show that a Marathi token might combine verb, nominalizing morpheme and case marker, while Hindi often separates these into individual tokens. This aligns with our empirical observations of model performance, where Marathi models typically required 20-30 % more parameters to achieve comparable performance levels to Hindi and Bengali models. The consistency between theoretical complexity measures and empirical model behavior provides measurable validation for our hypothesis of using SLMs to evaluate language complexity.

It is crucial to note that these complexity measures are conditional on the chosen tokenization strategy. As demonstrated previosuly (Arnett & Bergen, 2024), no language is inherently harder or easier for a language model to learn based solely on its morphological topology or tokenization strategy. The variations we observe in model performance likely stem from the interaction between intrinsic linguistic properties and practical factors such as dataset size and quality.

4. Conclusion

We demonstrated the TinyStories paradigm's effectiveness for Indian languages, showing SLMs with just 5-50 million parameters can generate coherent outputs. Our largest model ($\approx 150M$ parameters) achieves 90 % of GPT-4o's performance (≈ 8 vs ≈ 9 on overall evaluations), despite being 10^6 times smaller.

Further Regional TinyStories provides a novel approach for comparing language complexities and tokenizer efficiencies based on our inference evaluation framework. We find that Hindi models using Sarvam perform the best. Its observed that Bengali favors creativity, Hindi has slightly weaker context-completeness links, and Marathi uniquely associates context with grammar, suggesting distinct learning capabilities. Through comprehensive evaluation, we established that Sarvam and SUTRA tokenizers outperform alternatives like Tiktoken —the first study comparing regional tokenizers through inference evaluations rather than token fertility metrics. Our mechanistic analysis points toward higher entropy in tokenization behind the lower overall performance for Marathi compared to Hindi and Bengali, suggesting potential differences in how tokenizers deal with language complexity.

Our work demonstrates that effective Indian language modeling may not require massive architectures but rather quality, focused datasets, potentially democratizing language technology for underrepresented languages.

5. Limitation

- Due to computational resource constraint we could not run the same model several times for gathering more statistics, even though for each model run, 3000 stories were generated for inference evaluations.imilarly we could not translate English to Marathi.
- We do not incorporate human-in-the-loop evaluations of the SLM generated stories. The use of LLM as judge is still an exploratory topic. Potential biases and reliability of evaluation is an active field of research (Chen et al., 2024).
- 3. While our 54M-parameter models achieve decent scores across Hindi, Marathi, and Bengali, we observed consistent patterns where context scores (7.2-7.7) lag behind fluency scores (8.1-8.6). This may be explained by previous research (Peng et al., 2022; 2023) about neural models' challenges with entity-relationship tracking and deeper causal consistency.
- 4. WeightWatcher analysis (Appendix G) revealed under-training in our models, suggesting potential benefits from additional training epochs and targeted regularization.
- 5. We did not visualize and analyze the attention and activation maps of the models, and show how they relate to the generation process and the story content for different languages. This may shed more light on the observed variety in learning different linguistic attributes in case of Hindi or Marathi models.
- 6. Future work should explore hybrid architectures combining our models' strong fluency with explicit entity-relationship tracking mechanisms, potentially bridging the gap between statistical pattern recognition and human-like narrative understanding in Indian language story generation.

Impact Statement

Our work in generating children's stories in Indian regional languages presents both significant opportunities and challenges for educational accessibility and cultural preservation. We intend to release this as open weight models for deployment across diverse environments from edge devices to cloud infrastructure. While this technology could democratize access to children's literature in underserved languages and support early childhood literacy in resource-constrained environments, it raises important considerations about cultural authenticity and content quality. The system's ability to generate low-cost, scalable content could help address the scarcity of children's literature in

many languages, particularly beneficial for rural areas with limited publishing infrastructure. However, this advancement necessitates careful consideration of cultural nuances, content moderation, and the preservation of regional storytelling traditions. To ensure responsible deployment, we recommend implementing robust review mechanisms involving language experts, establishing clear guidelines for cultural appropriateness, and developing metrics to measure educational impact. The technology should complement rather than replace traditional storytelling, working in concert with local educators and cultural experts to maintain authenticity while leveraging the benefits of AI-assisted content generation.

Acknowledgement

We gratefully acknowledge the generous support from TensorDock (https://tensordock.com), whose \$1,500 research grant enabled the computational experiments in this work. Their cloud computing infrastructure was instrumental in training and evaluating our regional language models.

Code and model repository

We publicly release all our codes and datasets used for the experiments described in this work as well as selected trained models, to allow for external validation and reproducibility:

- 1. Code : https://github.com/VizuaraAI/
 Tiny-Stories-Regional
- 2. Code : https://github.com/nirvan840/
 Vizuara-TinyStories-Regional
- 3. Dataset and models : https://huggingface.
 co/TinyStories-Regional

References

- Arnett, C. and Bergen, B. K. Why do language models perform worse for morphologically complex languages?, 2024. URL https://arxiv.org/abs/2411.14198.
- Bafna, N. and Žabokrtský, Z. Subword-based crosslingual transfer of embeddings from Hindi to Marathi and Nepali. In Nicolai, G. and Chodroff, E. (eds.), Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pp. 61–71, Seattle, Washington, July 2022. Association for Computational Linguistics. doi:10.18653/v1/2022.sigmorphon-1.7. URL https://aclanthology.org/2022.sigmorphon-1.7/.
- Banerjee, S. and Lavie, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Goldstein, J., Lavie, A., Lin, C.-Y., and Voss, C. (eds.), *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL https://aclanthology.org/W05-0909/.
- Bendale, A., Sapienza, M., Ripplinger, S., Gibbs, S., Lee, J., and Mistry, P. Sutra: Scalable multilingual language model architecture, 2024. URL https://arxiv.org/abs/2405.06694.
- Black, S., Gao, L., Wang, P., Leahy, C., and Biderman, S. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow. 2021. URL https://api.semanticscholar.org/CorpusID:245758737.
- Boughorbel, S., Parvez, M. R., and Hawasly, M. Improving language models trained on translated data with continual pre-training and dictionary learning analysis, 2024. URL https://arxiv.org/abs/2405.14277.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.
- Chen, G. H., Chen, S., Liu, Z., Jiang, F., and Wang, B. Humans or LLMs as the judge? a study on judgement bias. In Al-Onaizan, Y., Bansal,

- M., and Chen, Y.-N. (eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 8301–8327, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.emnlpmain.474. URL https://aclanthology.org/2024.emnlp-main.474/.
- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. Palm: Scaling language modeling with pathways, 2022. URL https://arxiv.org/abs/2204.02311.
- Dabre, R., Shrotriya, H., Kunchukuttan, A., Puduppully, R., Khapra, M. M., and Kumar, P. Indicbart: A pretrained model for indic natural language generation. *arXiv* preprint arXiv:2109.02903, 2021.
- Dolamic, L. and Savoy, J. Comparative study of indexing and search strategies for the hindi, marathi, and bengali languages. *ACM Transactions on Asian Language Information Processing*, 9(3), September 2010. ISSN 1530-0226. doi:10.1145/1838745.1838748. URL https://doi.org/10.1145/1838745.1838748.
- Doshi, M., Dabre, R., and Bhattacharyya, P. Pretraining language models using translationese. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5843–5862, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.emnlpmain.334. URL https://aclanthology.org/2024.emnlp-main.334/.
- Eldan, R. and Li, Y. roneneldan/tinystories. https://huggingface.co/datasets/roneneldan/TinyStories, 2023a.
- Eldan, R. and Li, Y. Tinystories: How small can language models be and still speak coherent english?, 2023b. URL https://arxiv.org/abs/2305.07759.
- Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Celebi, O., Wenzek, G., Chaud-

- hary, V., et al. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48, 2021.
- Gala, J., Chitale, P. A., AK, R., Gumma, V., Doddapaneni, S., Kumar, A., Nawale, J., Sujatha, A., Puduppully, R., Raghavan, V., et al. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. arXiv preprint arXiv:2305.16307, 2023.
- Gilkerson, J., Richards, J. A., Warren, S. F., Montgomery, J. K., Greenwood, C. R., Oller, D. K., Hansen, J. H. L., and Paul, T. D. Mapping the early language environment using all-day recordings and automated analysis. *American journal of speech-language pathology*, 26 2:248–265, 2017. URL https://api.semanticscholar.org/CorpusID:4447742.
- Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.
- Holmström, O., Kunz, J., and Kuhlmann, M. Bridging the resource gap: Exploring the efficacy of English and multilingual LLMs for Swedish. In Ilinykh, N., Morger, F., Dannélls, D., Dobnik, S., Megyesi, B., and Nivre, J. (eds.), *Proceedings of the Second Workshop on Resources and Representations for Under-Resourced Languages and Domains (RESOURCEFUL-2023)*, pp. 92–110, Tórshavn, the Faroe Islands, May 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.resourceful-1.13/.
- Jiao, W., Wang, W., tse Huang, J., Wang, X., Shi, S., and Tu, Z. Is chatgpt a good translator? yes with gpt-4 as the engine, 2023. URL https://arxiv.org/abs/ 2301.08745.
- Karpathy, A. nanogpt, 2022. URL https://github.com/karpathy/nanoGPT.
- Kocmi, T. and Federmann, C. Large language models are state-of-the-art evaluators of translation quality, 2023. URL https://arxiv.org/abs/2302.14520.
- Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

- Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020.
- Martin, C. H., Peng, T. S., and Mahoney, M. W. Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data. *Nature Communications*, 12(1):4122, Jul 2021. ISSN 2041-1723. doi:10.1038/s41467-021-24025-8. URL https://doi.org/10.1038/s41467-021-24025-8.
- McCarthy, P. M. and Jarvis, S. Mtld, vocd-d, and hdd: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods*, 42(2):381–392, May 2010. ISSN 1554-3528. doi:10.3758/BRM.42.2.381. URL https://doi.org/10.3758/BRM.42.2.381.
- Muckatira, S., Deshpande, V., Lialin, V., and Rumshisky, A. Emergent abilities in reduced-scale generative language models, 2024. URL https://arxiv.org/abs/2404.02204.
- OpenAI. Tiktoken, 2024. URL https://github.com/openai/tiktoken.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pp. 311–318, USA, 2002. Association for Computational Linguistics. doi:10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.
- Peng, X., Xie, K., Alabdulkarim, A., Kayam, H., Dani, S., and Riedl, M. O. Guiding neural story generation with reader models, 2022. URL https://arxiv.org/abs/2112.08596.
- Peng, X., Li, S., Wiegreffe, S., and Riedl, M. Inferring the reader: Guiding automated story generation with commonsense reasoning, 2023. URL https://arxiv.org/abs/2105.01311.
- Petrov, A., Malfa, E. L., Torr, P. H. S., and Bibi, A. Language model tokenizers introduce unfairness between languages, 2023. URL https://arxiv.org/abs/2305.15425.
- Samuel, D., Kutuzov, A., Øvrelid, L., and Velldal, E. Trained on 100 million words and still in shape: BERT meets British National Corpus. In Vlachos, A. and Augenstein, I. (eds.), Findings of the Association for Computational Linguistics: EACL 2023, pp. 1954–1974,

- Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi:10.18653/v1/2023.findings-eacl.146. URL https://aclanthology.org/2023.findings-eacl.146/.
- Sarvam. Sarvam 1: The first indian language llm, 2024. URL https://www.sarvam.ai/blogs/ sarvam-1.
- Team, N., Costa-jussà, M. R., Cross, J., Çelebi, O., Elbayad, M., Heafield, K., Heffernan, K., Kalbassi, E., Lam, J., Licht, D., Maillard, J., Sun, A., Wang, S., Wenzek, G., Youngblood, A., Akula, B., Barrault, L., Gonzalez, G. M., Hansanti, P., Hoffman, J., Jarrett, S., Sadagopan, K. R., Rowe, D., Spruit, S., Tran, C., Andrews, P., Ayan, N. F., Bhosale, S., Edunov, S., Fan, A., Gao, C., Goswami, V., Guzmán, F., Koehn, P., Mourachko, A., Ropers, C., Saleem, S., Schwenk, H., and Wang, J. No language left behind: Scaling humancentered machine translation, 2022. URL https://arxiv.org/abs/2207.04672.
- Theodoropoulos, N., Filandrianos, G., Lyberatos, V., Lymperaiou, M., and Stamou, G. Berttime stories: Investigating the role of synthetic story data in language pretraining, 2024. URL https://arxiv.org/abs/2410.15365.
- Tiedemann, J., Aulamo, M., Bakshandaeva, D., Boggia, M., Grönroos, S.-A., Nieminen, T., Raganato A., Scherrer, Y., Vazquez, R., and Virpioja, S. Democratizing neural machine translation with OPUS-MT. *Language Resources and Evaluation*, (58):713–755, 2023. ISSN 1574-0218. doi:10.1007/s10579-023-09704-w.
- Tomasello, M. *Toward a Psychology of Language Acquisition*, pp. 323–328. Harvard University Press, 2003. ISBN 9780674010307. URL http://www.jstor.org/stable/j.ctv26070v8.11.
- Wang, F., Zhang, Z., Zhang, X., Wu, Z., Mo, T., Lu, Q., Wang, W., Li, R., Xu, J., Tang, X., He, Q., Ma, Y., Huang, M., and Wang, S. A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness, 2024. URL https://arxiv.org/abs/2411.03350.
- Warstadt, A., Choshen, L., Mueller, A., Williams, A., Wilcox, E., and Zhuang, C. Call for papers the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus, 2023a. URL https://arxiv.org/abs/2301.11796.
- Warstadt, A., Parrish, A., Liu, H., Mohananey, A., Peng, W., Wang, S.-F., and Bowman, S. R. Blimp: The benchmark of linguistic minimal pairs for english, 2023b. URL https://arxiv.org/abs/1912.00582.

- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. mt5: A massively multilingual pre-trained text-to-text transformer. arXiv preprint arXiv:2010.11934, 2020.
- Zhang, M. and Toral, A. The effect of translationese in machine translation test sets. In Bojar, O., Chatterjee, R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., Martins, A., Monz, C., Negri, M., Névéol, A., Neves, M., Post, M., Turchi, M., and Verspoor, K. (eds.), *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pp. 73–81, Florence, Italy, August 2019. Association for Computational Linguistics. doi:10.18653/v1/W19-5208. URL https://aclanthology.org/W19-5208/.
- Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. Bertscore: Evaluating text generation with bert, 2020. URL https://arxiv.org/abs/1904.09675.
- Zouhar, V., Meister, C., Gastaldi, J., Du, L., Sachan, M., and Cotterell, R. Tokenization and the noiseless channel. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5184–5207, Toronto, Canada, July 2023. Association for Computational Linguistics. doi:10.18653/v1/2023.acl-long.284. URL https://aclanthology.org/2023.acl-long.284/.

Appendix

A. Renyi Entropy of Hindi, Marathi, and Bengali with SUTRA and Sarvam Tokenizers

| Language | Tokenizer | $\alpha = 0.5$ | $\alpha = 1.0$ | $\alpha = 2.0$ |
|----------|-----------|----------------|----------------|----------------|
| Hindi | SUTRA | 9.77 | 8.67 | 7.49 |
| Hindi | Sarvam | 10.07 | 8.51 | 6.75 |
| Marathi | SUTRA | 10.69 | 9.37 | 8.15 |
| Marathi | Sarvam | 11.06 | 9.40 | 7.21 |
| Bengali | SUTRA | 9.82 | 8.79 | 7.72 |
| Bengali | Sarvam | 9.91 | 8.63 | 6.91 |

Table 7. Rényi entropy of Hindi, Marathi, and Bengali with SUTRA and Sarvam tokenizers

When examining these metrics at different values of α (0.5, 1.0, 2.0), we observe consistent patterns that illuminate different aspects of language complexity:

- 1. At $\alpha = 0.5$, emphasizing rare tokens, Marathi shows the highest entropy (SUTRA: 10.69, Sarvam: 11.06), suggesting greater diversity in its rare token distributions.
- 2. At $\alpha = 1.0$ (Shannon entropy), all languages show moderate convergence, though Marathi maintains higher values (SUTRA: 9.37, Sarvam: 9.40).
- 3. At $\alpha=2.0$, emphasizing common tokens, the differences between languages become less pronounced, though the relative ordering remains consistent.

According to Arnett & Bergen (2024), agglutinative languages have higher Rényi entropy compared to fusional languages. A study comparing Hindi, Marathi, and Bengali notes that Marathi's agglutinative structure creates more complex inflectional patterns, requiring distinct stemming strategies for information retrieval tasks. Bengali's simpler fusional morphology contrasts with Marathi's suffix-heavy word formation (Dolamic & Savoy, 2010).

B. Statistical Analysis of Inference Results

More details are provided for the inference evaluations for the 54M model that used Sarvam tokenizer.

B.O.1. DISTRIBUTION ANALYSIS OF EVALUATION METRICS

We conducted a comprehensive statistical analysis of the 3,000 stories generated by our Small Language Models (SLMs) for each of the three target languages: Hindi, Bengali, and Marathi, using Sarvam tokenizer. This analysis provides deeper insights into the performance characteristics of our models across various evaluation dimensions.

B.0.2. DISTRIBUTIONAL CHARACTERISTICS

The evaluation scores for all three languages exhibit distinct distributional patterns that reveal important aspects of model behavior (Figures 4-9). Below, we summarize key statistical properties observed across languages and metrics.

B.0.3. Cross-Linguistic Performance Patterns

Our analysis reveals several significant cross-linguistic patterns that provide insights into both model behavior and inherent language characteristics:

1. **Hierarchical Emergence of Capabilities**: Across all three languages, we observe a consistent hierarchy in performance metrics, with grammar consistently achieving the highest scores (Hindi: 8.91, Marathi: 8.72, Bengali: 8.82),

Table 8. Statistical Properties of Evaluation Metrics Across Languages for 54M models

| Language | Metric | Mean | Median | Std Dev |
|----------|-------------------|------|--------|---------|
| Hindi | Context Awareness | 7.73 | 8.00 | 1.01 |
| | Completeness | 7.78 | 8.00 | 0.86 |
| | Grammar | 8.91 | 9.00 | 0.34 |
| | Fluency | 8.55 | 9.00 | 0.56 |
| | Creativity | 7.81 | 8.00 | 0.58 |
| | Overall | 7.79 | 8.00 | 0.52 |
| Marathi | Context Awareness | 7.25 | 8.00 | 1.18 |
| | Completeness | 7.41 | 7.00 | 0.87 |
| | Grammar | 8.72 | 9.00 | 0.50 |
| | Fluency | 8.11 | 8.00 | 0.64 |
| | Creativity | 7.55 | 8.00 | 0.69 |
| | Overall | 7.50 | 8.00 | 0.67 |
| Bengali | Context Awareness | 7.51 | 8.00 | 1.11 |
| | Completeness | 7.64 | 7.00 | 0.85 |
| | Grammar | 8.82 | 9.00 | 0.42 |
| | Fluency | 8.42 | 8.00 | 0.59 |
| | Creativity | 7.69 | 8.00 | 0.59 |
| | Overall | 7.68 | 8.00 | 0.57 |

followed by fluency (Hindi: 8.55, Marathi: 8.11, Bengali: 8.42), completeness (Hindi: 7.78, Marathi: 7.41, Bengali: 7.64), and context awareness (Hindi: 7.73, Marathi: 7.25, Bengali: 7.51). This pattern aligns with the developmental progression observed in the original TinyStories research, suggesting that grammatical competence emerges earlier than contextual understanding, regardless of language.

- 2. **Bimodal Distribution of Context Scores**: The violin plots reveal a distinctive bimodal distribution for context awareness scores across all three languages, with concentration of scores around 7 and 8-9 ranges. This bimodality suggests that stories tend to either achieve strong contextual coherence or struggle with maintaining context throughout the narrative, with relatively few stories falling in the intermediate range. This pattern is evident across all three languages but varies in intensity.
- 3. **Consistency in Grammar Scores**: Grammar scores exhibit the lowest standard deviation across all languages (Hindi: 0.34, Marathi: 0.50, Bengali: 0.42), indicating that once basic grammatical competence is achieved, it remains relatively stable across generated stories. The narrow distribution of grammar scores visible in the violin plots demonstrates the models' tendency to consistently produce grammatically correct text.
- 4. Language-Specific Performance Differences: Hindi outperforms both Bengali and Marathi across most metrics, with the most substantial advantage in grammar (Hindi: 8.91 vs. Marathi: 8.72) and fluency (Hindi: 8.55 vs. Marathi: 8.11). Marathi consistently shows lower performance across all metrics. This finding is particularly noteworthy given the relationship with Rényi entropy values discussed in the paper, suggesting that languages with higher entropy measures may present greater challenges for coherent text generation.

B.1. Relationship Between Model Architecture and Evaluation Metrics

To understand how different architectural choices affect specific linguistic capabilities, we conducted correlation analyses between model parameters and evaluation metrics.

B.1.1. PARAMETER EFFICIENCY ACROSS LANGUAGES

Tables 1-3 in the main text illustrate the relationship between model parameter count and evaluation metrics for each language. Several key observations emerge:

- Divergent Scaling Patterns: While all languages benefit from increased model size, the marginal improvements from scaling differ significantly. The comparable performance of similarly-sized models (54M parameters) across the three languages suggests that architectural scaling properties may be relatively consistent, though the absolute performance levels differ. Hindi demonstrates the strongest performance at this parameter range, followed by Bengali and then Marathi.
- 2. **Optimal Parameter Allocation**: The inflection point in the performance-parameter curve occurs consistently around 54M parameters across all three languages, with minimal improvements beyond this threshold. However, the specific distribution of these parameters (between embedding dimension and layer depth) that yields optimal performance varies by language. Bengali achieves optimal performance with a balanced 512/6 configuration, while Hindi benefits more from increased width (1024/7) than depth.
- 3. **Parameter Elasticity by Metric**: Different evaluation metrics show varying sensitivity to parameter scaling. Grammar scores demonstrate the lowest elasticity (average 12% improvement from 4.46M to 153M parameters across languages), while context awareness shows the highest (average 33% improvement). This supports our hypothesis regarding the hierarchical emergence of capabilities, with grammatical competence requiring less model capacity than contextual understanding.

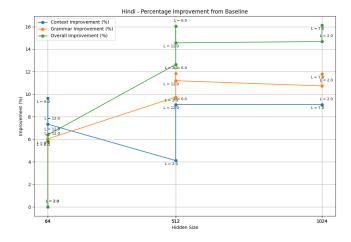


Figure 4. Percentage Improvement in Contextual Comprehension, Grammatical Accuracy, and Overall Performance from Baseline Scores across Hindi models detailed in Table 1

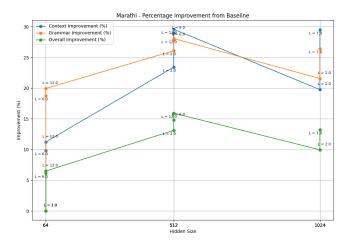


Figure 5. Percentage Improvement in Contextual Comprehension, Grammatical Accuracy, and Overall Performance from Baseline Scores across Marathi models detailed in Table 2

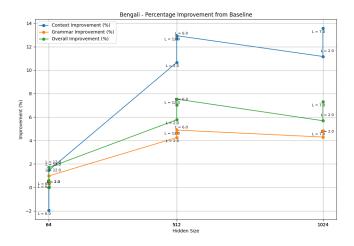


Figure 6. Percentage Improvement in Contextual Comprehension, Grammatical Accuracy, and Overall Performance from Baseline Scores across Bengali models detailed in Table 3

B.2. Correlation Analysis Between Metrics

Hindi The evaluation metrics in Hindi short stories demonstrated significant inter-correlations. The strongest association was observed between creativity and overall quality assessment (r = 0.73, t(2998) = 58.48, p < .001), indicating that creative elements substantially influenced holistic quality perceptions. Grammar and overall quality demonstrated a robust positive relationship (r = 0.69, t(2998) = 52.20, p < .001), suggesting grammatical accuracy significantly contributed to quality judgments. Notably, completeness and fluency exhibited a strong correlation (r = 0.72, t(2998) = 56.81, p < .001), indicating narrative completeness typically accompanied smooth readability. The weakest relationship was identified between context awareness and completeness (r = 0.35, t(2998) = 20.46, p < .001), suggesting these constructs captured distinct dimensions of narrative quality.

Bengali Analysis of Bengali short stories revealed similar correlation patterns, with creativity and overall quality showing the highest correlation coefficient (r = 0.80, t(2998) = 73.01, p < .001). This suggests that creative expression was the predominant factor in quality assessment. Grammar and overall quality maintained a strong positive relationship (r = 0.71, t(2998) = 55.20, p < .001), highlighting the importance of grammatical precision. Completeness and fluency demonstrated substantial correlation (r = 0.77, t(2998) = 66.08, p < .001), reinforcing the connection between narrative coherence and reading experience observed across languages. Context awareness and completeness displayed a moderate correlation (r = 0.39, t(2998) = 23.19, p < .001), indicating these metrics evaluated partially distinct aspects of narrative construction.

Marathi In contrast to Hindi and Bengali, Marathi short stories exhibited their strongest correlation between context awareness and grammar (r = 0.78, t(2998) = 68.25, p < .001), suggesting a language-specific relationship between contextual appropriateness and grammatical structure. Creativity and overall quality maintained equivalent correlation strength (r = 0.78, t(2998) = 68.25, p < .001), consistent with patterns observed in the other languages. Completeness and fluency correlation remained robust (r = 0.77, t(2998) = 66.08, p < .001), indicating a consistent relationship across all three languages. The weakest association was observed between completeness and creativity (r = 0.49, t(2998) = 30.78, p < .001), suggesting these dimensions function more independently in Marathi narratives compared to Hindi and Bengali.

These findings reveal both cross-linguistic patterns and language-specific relationships between evaluation metrics, with implications for understanding quality assessment in Indic-language short stories. Given the large sample size (n = 3000 per language), all correlations were statistically significant at p < .001, with the critical value for significance at this level being r = 0.060.

B.3. Comparative Analysis of Score Distributions

B.3.1. DISTRIBUTION VARIATION ANALYSIS

Examining the standard deviations across metrics and languages provides insight into the consistency of model performance:

| <i>Table 9.</i> Standard | Deviation Comparison | Across Me | errics and | Languages |
|--------------------------|----------------------|-----------|------------|-----------|
| | | | | |

| Metric | Hindi | Marathi | Bengali | Average |
|-------------------|-------|---------|---------|---------|
| Context Awareness | 1.01 | 1.18 | 1.11 | 1.10 |
| Completeness | 0.86 | 0.87 | 0.85 | 0.86 |
| Grammar | 0.34 | 0.50 | 0.42 | 0.42 |
| Fluency | 0.56 | 0.64 | 0.59 | 0.60 |
| Creativity | 0.58 | 0.69 | 0.59 | 0.62 |
| Overall | 0.52 | 0.67 | 0.57 | 0.59 |
| Average | 0.65 | 0.76 | 0.69 | 0.70 |
| | | | | |

Notable patterns include:

1. **Consistent Hierarchy of Variability**: Across all languages, Context Awareness shows the highest standard deviation (average: 1.10), indicating greater variability in the model's ability to maintain contextual coherence. Grammar consistently shows the lowest standard deviation (average: 0.42), suggesting that grammatical competence is more uniformly achieved once the model reaches sufficient capacity.

- 2. **Language-Specific Consistency Patterns**: Marathi shows higher standard deviations across all metrics (average: 0.76) compared to Hindi (0.65) and Bengali (0.69), suggesting greater variability in performance. This is particularly evident in context awareness (Marathi: 1.18 vs. Hindi: 1.01) and overall scores (Marathi: 0.67 vs. Hindi: 0.52).
- 3. **Form vs. Content Metrics**: Metrics related to linguistic form (grammar, fluency) consistently show lower standard deviations (0.42, 0.60) than those related to content (context, completeness, creativity) (1.10, 0.86, 0.62). This pattern suggests that form-related capabilities may develop more uniformly compared to content-related capabilities.

B.3.2. Consistency-Performance Relationship

Examining the relationship between metric means and standard deviations reveals important patterns:

- 1. **Inverse Relationship**: Across all languages, we observe an inverse relationship between mean scores and standard deviations. Metrics with higher means (like grammar) tend to have lower standard deviations, while metrics with lower means (like context awareness) show higher standard deviations. This pattern suggests that as performance on a particular aspect improves, consistency also increases.
- 2. **Language-Specific Consistency**: All three languages show a moderate to strong negative correlation between means and standard deviations, with Marathi having the strongest inverse relationship (r = 0.77), while both Bengali and Hindi show identical correlations (r = 0.70). This negative correlation indicates that metrics with higher mean scores tend to have lower variability across all three languages, suggesting more consistent performance in areas where the models score higher.
- 3. **Metric-Specific Patterns**: Grammar shows both the highest means and lowest standard deviations across all languages, suggesting that grammatical competence represents a "foundational" capability that is both strong and consistent once achieved. Context awareness, by contrast, shows lower means and higher standard deviations, indicating it may represent a more advanced capability that remains challenging even as models improve.

B.4. Performance Gap Analysis

To better understand the relative strengths and weaknesses of the models across different languages, we analyze the gaps between different evaluation metrics:

| Metric Pair | Hindi | Marathi | Bengali | Average |
|-----------------------------|-------|---------|---------|---------|
| Grammar - Context Awareness | 1.18 | 1.47 | 1.31 | 1.32 |
| Grammar - Completeness | 1.13 | 1.31 | 1.18 | 1.21 |
| Grammar - Creativity | 1.10 | 1.17 | 1.13 | 1.13 |
| Grammar - Fluency | 0.36 | 0.61 | 0.40 | 0.46 |
| Fluency - Context Awareness | 0.82 | 0.86 | 0.91 | 0.86 |
| Fluency - Completeness | 0.77 | 0.70 | 0.78 | 0.75 |
| Fluency - Creativity | 0.74 | 0.56 | 0.73 | 0.68 |
| Context - Completeness | -0.05 | -0.16 | -0.13 | -0.11 |

Table 10. Performance Gaps Between Metrics (Difference in Mean Scores)

Key patterns include:

- 1. **Consistent Gap Hierarchy**: Across all languages, the largest performance gap is between grammar and context awareness (average: 1.32), while the smallest gap among the major metric pairs is between context awareness and completeness (average: -0.11, with context scores actually lower than completeness in all languages). This consistent pattern suggests a universal hierarchy in how different linguistic capabilities develop in these models.
- 2. **Language-Specific Gap Patterns**: Marathi shows notably larger gaps between grammar and other metrics (Grammar-Context: 1.47, Grammar-Completeness: 1.31) compared to Hindi and Bengali. This suggests that while Marathi models achieve reasonable grammar scores, they struggle more with contextual coherence and narrative completeness compared to models in other languages.

- 3. **Form-Content Divide**: The substantial gaps between form-related metrics (grammar, fluency) and content-related metrics (context, completeness, creativity) highlight the models' stronger capabilities in producing structurally correct text versus semantically coherent narratives. This divide is most pronounced in Marathi and least evident in Hindi.
- 4. **Grammar-Fluency Relationship**: The gap between grammar and fluency scores is significantly smaller (average: 0.46) than between grammar and other metrics, suggesting these capabilities may develop in tandem. This pattern holds across all three languages, though Marathi shows a larger grammar-fluency gap (0.61) compared to Hindi (0.36) and Bengali (0.40).

B.5. Statistical Significance Analysis

To assess the significance of observed cross-linguistic differences, we analyzed the overall performance scores across the three languages:

| Language | Mean | Standard Deviation |
|----------|------|--------------------|
| Hindi | 7.79 | 0.52 |
| Marathi | 7.50 | 0.67 |
| Bengali | 7.68 | 0.57 |

These results suggest that:

- 1. The performance differences between languages appear meaningful, with Hindi outperforming both Bengali and Marathi, and Marathi showing the lowest overall performance.
- 2. The standard deviations indicate different levels of consistency across languages, with Hindi showing the most consistent performance (SD = 0.52) and Marathi showing the greatest variability (SD = 0.67).
- 3. These performance differences align with the entropy analysis presented in the main paper, where Marathi exhibited higher Rényi entropy values (7.76) compared to Hindi (7.15) and Bengali (7.41), suggesting a potential relationship between tokenization complexity and generation performance.

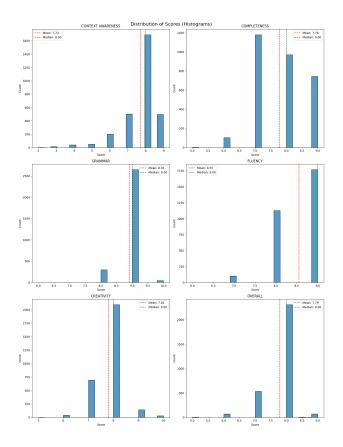
B.6. Final comments

Our statistical analysis reveals complex relationships between tokenization strategies, linguistic properties, and generation performance across Hindi, Marathi, and Bengali. The consistent hierarchy of capabilities (grammar > fluency > completeness > context) across all three languages suggests universal aspects of language model development, while significant cross-linguistic differences in absolute performance point to the importance of language-specific optimization.

The performance metrics for comparable model architectures (53-54M parameters) across the three languages show Hindi achieving the strongest overall results (7.79), followed by Bengali (7.68) and Marathi (7.50). This aligns with the Rényi entropy analysis presented in the main paper, suggesting that languages with higher entropy values may present greater challenges for coherent text generation.

These findings underscore the value of the Regional TinyStories framework as both a practical approach to developing efficient language models for Indian languages and as an analytical tool for understanding comparative linguistic complexity. Future work should focus on exploring the relationship between tokenization strategies, morphological characteristics, and model performance to develop more comprehensive metrics for predicting language modeling difficulty across typologically diverse languages.

Figures 4-9 show the histograms and violin plots for the Hindi, Marathi and Bengali languages for 3000 samples for the following metrics: context awareness, completeness, grammar, fluency, creativity and overall score. For each of the languages, the mean and the median scores for the different evaluation categories are provided by the red and the green dashed lines respectively on the subplots.



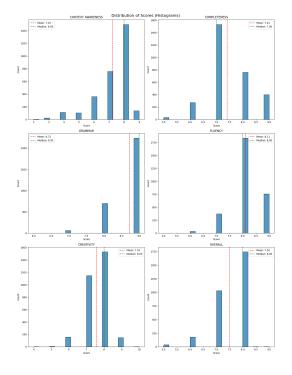
CONTEXT AWARENESS

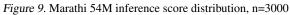
COMPLETENESS

COMPLETENES

Figure 7. Hindi 54M inference score distribution, n = 3000

Figure 8. Hindi 54M inference score violin plots, n = 3000





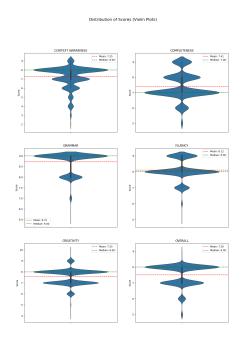
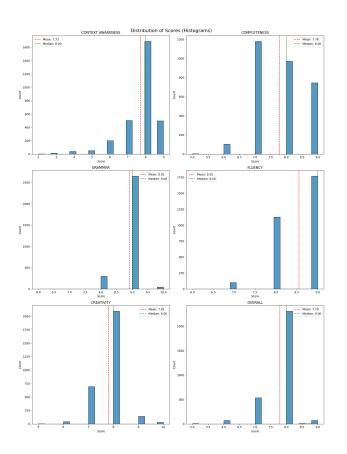


Figure 10. Marathi 54M inference score violin plots, n=3000



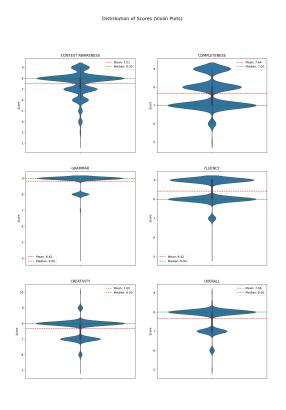


Figure 11. Bengali 54M inference score distribution, n = 3000

Figure 12. Bengali 54M inference score violin plots, n = 3000

C. Generated story samples for Marathi and Bengali

Figures 10-13 show the stories generated from our TinyStories 54M model and that of GPT-40 for Marathi and Bengali language for the same prompt. The prompt which we provided effectively translates to:

"Once upon a time, there was a small boy. His toy..."

एकादा एक छोटा मुलगा होता. त्याला खेळणी खूप आवडत होती. एक दिवस, तो आपल्या खेळण्यांसोबत बाहेर खेळायला गेला. त्याने एक सुंदर गडदशेतील बॉल घेतला आणि त्याला फेकायला सुरुवात केली. "हॅ बॉल किती मजेदार आहे!" तो आनंदाने म्हणाला. त्याच्या खेळार एक छोटा कोबडा आला. कोबडायने विचारले, "तू बॉल का फेकतोस?" मग तो कारण तो लांब गेल्या! मुलाचा उत्तरला. कोबडायने हमून म्हटले, "माझ्यासोबत खेळ, मी तुला एक गुपित सांगतो. "मुलाने आणि कोबडायने एका खेळले. पण अचानक, बॉल एका झाडाच्या फांदीवर अडकला. मुलाचा चिंतेत झाला. "आता तो कसा काढू?" तो विचार करत होता. कोबडायने त्याला मदत करायचे ठरवले. एका झाडावर चढणे खूप कठीण होते. अखेर, मुलाने आणि कोबडायने त्याला बॉल काढू शकले नाही. मुलाचा उदास झाला. त्याच्या खेळण्यातून त्याला आनंद मिळाला होता, पण आता तो बॉल काढू शकला नाही. त्या दिवशी मुलाला शिकवले की काही गोष्टी गमावल्यावर त्यांना पुन्हा मिळवणे कठीण असते. पण मित्रांसोबत खेळणे आणि आनंद घेणे महत्वाचे आहे.

Figure 13. Regional TinyStories 54M Marathi model generated story

এক সময় একটা ছোট ছেলে ছিল। তার খেলনা গাড়িটি নিয়ে খেলছিল। রাছলের গাড়ি নিয়ে থামের চারপাশে ঘুরছিল। সে ভাবছিল, আজ কি হবে? \n\nরাছলের মা বললেন, \"বাছল, তুমি কি গাড়িটি নিয়ে থাতে পারো?\" রাছল মাথা নাড়ল এবং বলল, \"হাঁ, মা।\" মা বললেন, \"তুমি মি দাড়িটি বানাতে পারো, তাহলে আমি রোছন মাথা নাড়ল এবং বলল, \"হাঁ, মা।\" মা বললেন, \"তুমি মি দাড়িটি বানাতে পারো, তাহলে আমি তোমাকৈ একটি সুন্দর গাড়ি বানাব।\" মা বললেন, \"ত্বমাই, রাছল!\" \n\nরাছল এবং তার মা পারে গালেন। তারা গাড়িব ছবি আঁকতে শুক্র করল। সে দেখল, তার বন্ধুরা কত আনন্দিত। তাই রাছল বুঝতে পারল, সে শুধু গাড়িটি খুব ভালো কাজ করেনি। \n\nশেষে, রাছল তার মায়ের কাছে গিয়ে বলল, \"মা, আমি আজ একটি সুন্দর গাড়ি বানিয়েছি।" মা হাসলেন এবং বললেন, \"তুমি যদি চেষ্টা করো, তাহলে তুমি সবকিছু করতে পারবে।" বাছলের মনে একটি উজ্জ্বল অনুভূতি গড়ে উঠল, এবং সে জানল যে চেষ্টা করলে সবকিছু সম্ভব।

Figure 15. Regional TinyStories 54M Bengali model generated story

एकादा एक छोटा मुलगा होता. त्याला खेळणी खू<mark>प आवडत होती. एक दिवस, त्याने आपल्या खेळण्यातमधून</mark> एक खास खेळणी हरवले. तो चिंतेत होता. "माझी खेळणी कुठे गेली?" तो विचार करत होता. त्याला त्याच्या मित्रांना विचारचं ठरवले. त्यांने एक चिंपाजीला विचारले, "गुला माझी खेळणी सापडली दिसते का?" विंपाजीने उत्तर दिले, "मी पाहिले नाही, पण मी तुला मदत करीन." त्यांनंतर, त्यांनी एकत्रितपणे जंगलात शोध घेण्यास सुरुवात केली. झाडांच्या सखोल भागात त्यांनी शोध घेतला. अचानक, एक छोटा ससा त्यांच्या समोर आला. "ससा माझे खेळणी शोधत आहे," तसाला विचारले. मुलाने सांगितले, "मी ही पाहिले, ती त्या मोठ्या झाडाच्या मुळाशी आहे," मुलाने आणि चिंपाजीन धावत त्या झाडाकडे गेले. खरंत, त्यांना खेळणी तिथे सापडली. "धंन्यावाद, ससा!" मुलाने आनंदाने म्हणाले. सर्वांनी एकत्र येऊन खेळणी खेळण्यात मजा आली. कधी कधी, मित्रांची मदत चेतत्यास आपल्या आपल्याला गोष्टी मिळवता येतात.

Figure 14. GPT 40 generated Marathi story

এক সময় একটা ছোট ছেলে ছিল। তার খেলনা ছিল একখানা ছোট্ট কাঠের ঘোড়া। ছেলেটির নাম ছিল শুভ।
শুভ সেই কাঠের ঘোড়াটিকে খুব ভালোবাসত। সে ঘোড়াটিকে সাথে নিয়ে সারা বাড়ি দৌড়াতো, মনে হাতা যেন সে
রাজপুর আর তার ঘোড়াটি রাজার সবচেয়ে ফ্রুতগামী ঘোড়া! একদিন শুভ তার ঘোড়াটিকে নিয়ে রাগানে
খেলছিল। ঠাং এক বাতাদের রাগদীয় ঘোড়াটি উড়ে গিয়ে ঝোপের মধ্যে পড়ে গেল। শুভ অনেক খুঁজেও তার
প্রিয় খেলনাটি পেল না। সে খুব দুঃখ পেল এবং কাঁদতে লাগল।ঠিক তখনই একটা ছোট্ট খরগোন্দ লাফিয়ে এল।
খরগোন্দটি বলল, "শুভ, তুমি কাঁদছো কেন?" শুভ বলল, "আমার প্রিয় ঘোড়াটি হারিয়ে গেছে" খরগোন্দটি তার
লহা কান নাড়িয়ে বলল, "আমি তোমাকে খুঁজে দিতে পারি!" তারপর খরগোন্দ আর শুভ মিলে বাগানের কোণা
কোণা খুঁজতে লাগল। একসময় তারা ঝোপের নিচে ছোট্ট কাঠের ঘোড়াটিকে পেয়ে গেলা শুভ খুব খুন্দি হয়ে বলল,
"ধন্যবাদ, ছোট্ট খরগোন্দা" খরগোন্দটি হাসল এবং বলল, "বন্ধুর সাহায্য করাই তো আনন্দের বাগারা" সেদিন
থেকে শুভ ও খরগোন্দা খুব ভালো বন্ধু হয়ে গেল। শুভ তার কাঠের ঘোড়া আর খরগোন্দক সাথে নিয়ে বাগানে
খলতে লাগল। তারা একসাথে দৌড়াত, লুকোচুরি খেলত, আর দারুণ মজা করত। শুভ বুঞ্চত পারল, খেলনা
মূল্যবান ঠিকট, কিন্ত বন্ধুত্ব তার চয়েও বড় সম্পদ।

Figure 16. GPT 40 generated Bengali story

We then ask GPT-40 to qualitatively compare between the 2 stories. Here is the output for Marathi language:

Overall, the Text A on the left offers more playful dialogues and scene details around a lost ball, while Text B on the right provides a succinct adventure-style narrative about finding a missing pen. Both convey simple morals about seeking and giving help, but they differ in their degree of description, the nature of supporting characters, and the pacing of the action.

Here is the output for Bengali language:

Overall, Text A on the left (story by 54M model) revolves around a boy named Rahul and his toy car, focusing on everyday interactions and the excitement he feels learning to help and explore. Text B on the right (story by GPT 4.0), on the other hand, highlights a boy's wooden horse and leans into imaginative, almost fairy-tale elements where the child envisions himself as a prince. Both narratives depict a child's sense of wonder and creativity, yet Text A remains more grounded in everyday life, while Text B draws on a more whimsical, dream-like tone to convey its central theme of playful discovery.

D. Synthetic dataset generation through LLM prompting

D.1. Dataset Generation Strategy

The prompt generation process began with creating comprehensive lexical resources for each target language: Hindi, Bengali, and Marathi. We compiled vocabulary lists consisting of approximately 300 nouns, 300 verbs, and 300 adjectives appropriate for children aged 5-7 years for each of the languages. These were stored in language-specific text files.

Additionally, we developed "features" lists in both English and the target languages. These features represented narrative elements, themes, or tones to guide story generation (e.g., learning values, friendship themes, acts of kindness). These resources were consolidated into a structured JSON format for each language.

D.2. Unique Prompt Generation Algorithm

To ensure maximum diversity in the dataset while preventing duplicates, we implemented Algorithm 1.

Algorithm 1 Unique Prompt Generation

```
Input: N_{words} (nouns), V_{words} (verbs), A_{words} (adjectives), F_{words} (features), TargetCount (prompt count)
Output: P (unique prompts)
 1: UsedIDs \leftarrow \emptyset
 2: UsedTriplets \leftarrow \emptyset
 3: P \leftarrow \emptyset
 4: DuplicateCount \leftarrow 0
 5: while |P| < TargetCount do
       n \leftarrow \text{Select random element from } N_{words}
       v \leftarrow \text{Select random element from } V_{words}
 7:
       a \leftarrow \text{Select random element from } A_{words}
       f \leftarrow \text{Select random element from } F_{words}
 9:
10:
       ID \leftarrow ConcatenateIndices(n, v, a, f)
       TripletID \leftarrow ConcatenateIndices(n, v, a)
11:
       if ID \notin UsedIDs and TripletID \notin UsedTriplets then
12:
          UsedIDs \leftarrow UsedIDs \cup \{ID\}
13:
          UsedTriplets \leftarrow UsedTriplets \cup \{TripletID\}
14:
15:
          prompt \leftarrow FormatTemplate(n, v, a, f)
          P \leftarrow P \cup \{prompt\}
16:
17:
       else
          DuplicateCount \leftarrow DuplicateCount + 1
18:
       end if
19:
20: end while
21: return P, DuplicateCount
```

This approach effectively prevented repetition patterns in the dataset, eliminating approximately 37,500 potential duplicate prompts from the target 3M dataset per language. The tracking of both quadruplet and triplet identifiers ensured maximum lexical diversity in the stories.

D.3. Prompt Complexity Evolution

We systematically explored different prompt complexity levels to identify the optimal configuration for generating high-quality children's stories. Five distinct complexity levels were developed, with increasing sophistication:

- Level 1: Basic structure (TinyStories baseline) with minimal guidance
- Level 2: Enhanced structure with explicit narrative guidance (beginning/middle/end) and tone constraints
- Level 2+: Extended word limit (350-500 words) while maintaining structural guidance

- Level 3: Addition of dialogue elements (maximum three exchanges) and thematic guidance
- Level 4: Incorporation of cultural references (e.g., Panchatantra, Tenali Raman stories)
- Level 4+/5: Extension with supporting characters and natural elements

Through comparative evaluation using GPT-4 as the assessment model, complexity level 2+ was determined to provide the optimal balance of quality and generation efficiency. This template consistently yielded stories that achieved evaluation scores averaging 8.73 across all metrics (completeness, grammar, fluency, creativity).

D.4. Optimal Prompt Template

The Level 2+ template that produced the best results across languages followed this structure:

Optimal Prompt Template (Level 2+):

Write a short story in {language} suitable for 5-to-7-year-old children.

Use simple, easy-to-understand words and limit the story to 3-4 short paragraphs (around 350-500 words).

The story should feature a clear beginning, middle, and end.

Incorporate the verb "{verb}", the noun "{noun}", and the adjective "{adjective}" naturally into the story.

The story should also integrate the conclusion/tone "{feature}" through actions and outcomes, without directly stating the tone.

Remember to use only simple words and keep the story appropriate for the target age group.

Return the output as a JSON dictionary: { "story": "your_generated_story" }

Figure 17. Template used for story generation prompts across all three languages

This template's effectiveness stems from several critical elements:

- 1. It specifies a clear target audience and language
- 2. It provides explicit structural guidance (3-4 paragraphs, clear beginning/middle/end)
- 3. It incorporates lexical constraints (verb, noun, adjective) to guide vocabulary usage
- 4. It requests thematic integration (feature/tone) through narrative rather than explicit statements
- 5. It maintains appropriate word count constraints (350-500 words)
- 6. It specifies the return format (JSON) for consistent processing

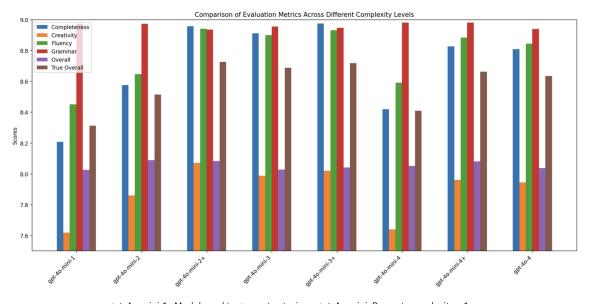
D.5. Implementation and Data Generation

The prompt generation process was designed for scalability. For each language, 3 million unique prompts were generated and stored in JSON files. The implementation included progress tracking and efficient JSON-based storage.

For the actual story generation, a parallel processing approach with multiple concurrent API sessions was employed. We could configure 4 concurrent sessions, each with 16 threads, to achieve an approximate generation rate of 100 stories per minute with GPT-40-mini as the generation model.

After evaluating multiple models (GPT-4o, LLaMA-3.1 70B, Claude 3.5 Sonnet), GPT-4o-mini was selected (Fig. 15) based on its optimal balance of quality and generation efficiency, consistently achieving an 8.5/10 average score across evaluation metrics.

The final dataset included 2.2 million synthetic stories each for Hindi, Bengali, and Marathi, all generated using this systematic approach. This data generation methodology ensured both diversity and quality in the Regional TinyStories dataset, enabling effective training of Small Language Models for these languages.



gpt-4o-mini-1: Model used to generate stories = gpt-4o-mini. Prompt complexity = 1 gpt-4o-mini-2+: Model used to generate stories = gpt-4o-mini. Prompt complexity = 2+ (where + = increased word limit of 350-500 words)

Figure 18. Comparison of Evaluation metrics across different complexity levels

E. Training Data Analysis: Linguistic Diversity and Evaluation Metric Performance

E.1. The Zero-ROUGE Phenomenon in Cross-Lingual Evaluation

Our experiments revealed a striking phenomenon when applying traditional n-gram based evaluation metric like ROUGE(Lin, 2004) to non-English text generation. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics designed to evaluate automatic summarization and machine translation by comparing generated text to reference texts. ROUGE-1 and ROUGE-2 measure the overlap of unigrams (single words) and bigrams (word pairs) respectively between the candidate and reference texts, while ROUGE-L uses the longest common subsequence to assess sentence-level structural similarity. We wanted to utilize ROUGE to analyze the diversity / quality of the synthetically generated training dataset for our SLMs. Although evaluating text generation quality for English has established benchmarks, we observed significant challenges when applying the same metrics to, e.g., LLM generated Bengali training stories from the TinyStories-Regional dataset.

E.1.1. CONTRASTING ROUGE PERFORMANCE BETWEEN LANGUAGES

When applied to the English TinyStories dataset (Eldan & Li, 2023b), ROUGE metrics provided nuanced scores reflecting different degrees of lexical overlap:

```
Average ROUGE Scores (English):
Average ROUGE-1 F1: 0.2916
Average ROUGE-2 F1: 0.0553
Average ROUGE-L F1: 0.1700
```

Individual story scores exhibited a normal distribution of values, matching the reports from the Tinystories paper:

Table 12. English TinyStories ROUGE scores sample

| story_idx | rouge1_f1 | rouge2_f1 | rougeL_f1 |
|-----------|-----------|-----------|-----------|
| 0 | 0.272727 | 0.054054 | 0.124579 |
| 1 | 0.258503 | 0.006849 | 0.102041 |
| 2 | 0.375000 | 0.094488 | 0.218750 |
| | | | |
| 9 | 0.266160 | 0.061303 | 0.152091 |

However, when the same methodology was applied to the Bengali TinyStories dataset (TinyStories-Regional/beng-generated_4o-mini_2M), ROUGE uniformly produced zero values:

```
Average ROUGE Scores (Bengali):
Average ROUGE-1 F1: 0.0000
Average ROUGE-2 F1: 0.0000
Average ROUGE-L F1: 0.0000
```

This striking result persisted across all ten pairs of evaluated stories, with precision, recall, and F1 scores uniformly at zero for all ROUGE variants.

E.1.2. CONTEXTUAL ANALYSIS OF EVALUATION METRIC PERFORMANCE

To better understand this phenomenon, we conducted a comprehensive comparative analysis using other evaluation metrics. BLEU (Bilingual Evaluation Understudy) (Papineni et al., 2002) is an algorithm for evaluating machine translation quality based on n-gram precision, measuring how many generated phrases match reference translations. BLEU scores range from 0 to 1, with higher values indicating closer alignment to human references, though the metric tends to favor shorter texts and often fails to capture semantic equivalence.

BERTScore (Zhang et al., 2020) leverages contextual embeddings from pre-trained language models to compute similarity between generated and reference texts at a semantic level rather than exact word matches. This approach allows BERTScore to recognize paraphrases and synonyms as similar, making it more robust for evaluating text generation quality in morphologically rich languages where lexical variation is common.

METEOR (Metric for Evaluation of Translation with Explicit ORdering) (Banerjee & Lavie, 2005) evaluates translation quality by calculating precision and recall weighted by importance, while also accounting for word order, stemming, and synonymy. METEOR typically correlates better with human judgments than BLEU by considering linguistic elements beyond n-gram matching, making it particularly useful for evaluating text in languages with flexible word order and rich morphology.

BLEU Score Analysis. BLEU scores for Bengali stories exhibited considerable variation yet remained consistently low. The mean BLEU score was 0.078 ($\sigma = 0.126$), with values ranging from 0.003 to 0.421. Story index 5 demonstrated a notably higher BLEU score (0.421), suggesting some lexical alignment with its reference. The generally low BLEU scores corroborated our ROUGE findings, confirming significant lexical divergence (Fig. 15).

BERTScore Analysis. In stark contrast to lexical metrics, BERTScore values were remarkably high across all Bengali story pairs. The mean BERTScore was 0.967 ($\sigma=0.012$), with scores ranging from 0.944 to 0.982. This dramatic difference between BLEU and BERTScore revealed a fundamental characteristic of the generated stories: while they utilize different vocabulary and phrasing from the references, they maintain high semantic fidelity (Fig. 15).

METEOR Score Analysis. METEOR scores occupied a middle ground between BLEU and BERTScore, with a mean of 0.153 ($\sigma = 0.046$) and range of 0.071 to 0.231. For the sample in Fig. 16, story index 2 achieved the highest METEOR score (0.231), while story index 9 received the lowest (0.071). The intermediate nature of METEOR scores reflects its design as a balanced metric that considers both lexical and semantic similarities.

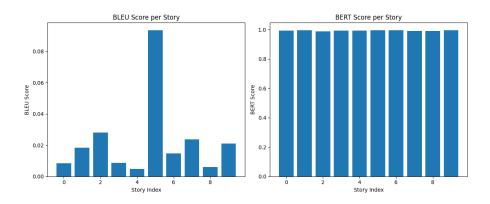


Figure 19. BLEU and BERT scores for 10 randomly selected stories from the synthetic Bengali dataset

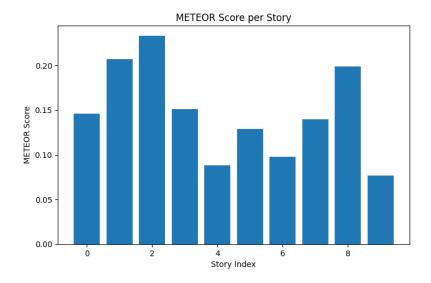


Figure 20. METEOR scores for 10 randomly selected stories from the synthetic Bengali dataset

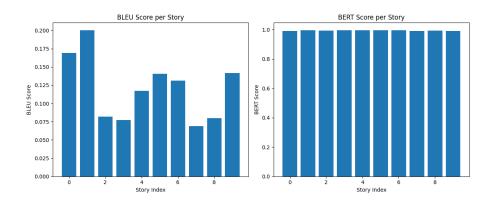


Figure 21. BLEU and BERT scores for 10 randomly selected stories from the synthetic Hindi dataset

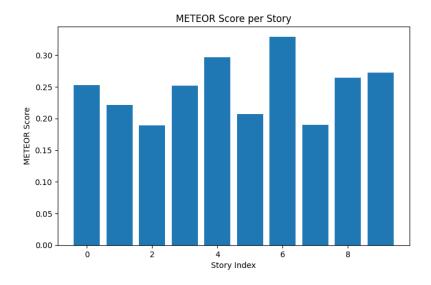


Figure 22. METEOR scores for 10 randomly selected stories from the synthetic Hindi dataset

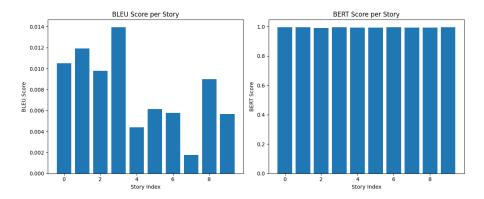


Figure 23. BLEU and BERT scores for 10 randomly selected stories from the synthetic Marathi dataset

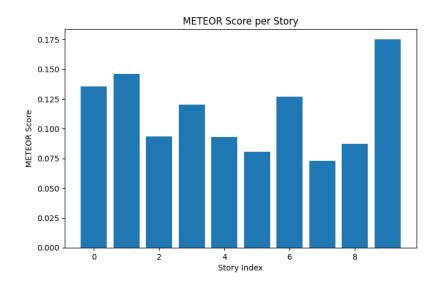


Figure 24. METEOR scores for 10 randomly selected stories from the synthetic Marathi dataset

Table 13. Pearson Correlation Coefficients Between Metrics

| Tuble 15: I carson correlation coefficients Between tite | | | | | |
|--|--|--|--|--|--|
| Metric Pair | Correlation Coefficient | | | | |
| BLEU-BERTScore | 0.29 | | | | |
| BLEU-METEOR | 0.63 | | | | |
| BERTScore-METEOR | 0.51 | | | | |
| | Metric Pair BLEU-BERTScore BLEU-METEOR | | | | |

The moderate correlation between BLEU and METEOR (r=0.63) suggests that despite METEOR's consideration of synonymy, it still maintains sensitivity to lexical overlap. The weaker correlation between BLEU and BERTScore (r=0.29) confirms that these metrics capture fundamentally different aspects of text similarity. Qualitatively similar observations hold true for randomly sampled synthetic training data in Hindi and Marathi as observed in Figs. 17-20.

E.2. Linguistic Factors Contributing to the Zero-ROUGE Phenomenon

The zero-ROUGE phenomenon observed in Bengali text evaluation can be attributed to several linguistic factors:

- 1. **Morphological Richness:** Bengali possesses a complex morphological structure with numerous inflectional and derivational forms, increasing the likelihood of lexical variation even when expressing identical concepts.
- 2. **Word Formation Patterns:** The agglutinative tendencies in Bengali create fewer opportunities for exact n-gram matches compared to English.
- 3. **Syntactic Flexibility:** Bengali permits greater variation in word order while preserving meaning, reducing the likelihood of matching n-grams even in semantically equivalent sentences.
- 4. **Training Methodologies:** Modern language models with multiple decoding paths may naturally produce diverse lexical realizations of similar semantic content, especially when the target language permits such variation.

This finding represents an extreme manifestation of the limitations of lexical metrics, where the absence of exact n-gram overlap, as evidenced by zero ROUGE scores, suggests that text generation systems employ sophisticated paraphrasing mechanisms while maintaining semantic coherence.

E.3. Implications for Multi-Lingual Text Generation Evaluation

Our analysis suggests that robust evaluation of text generation requires a multi-metric, language-aware approach. Based on our findings, we propose:

E.3.1. LANGUAGE-SPECIFIC CONSIDERATIONS

- 1. **Metric Selection:** Researchers must carefully select evaluation metrics appropriate to the target language, considering morphological complexity and typical paraphrasing patterns.
- 2. **Benchmark Calibration:** Distinct performance benchmarks should be established for each language rather than applying universal thresholds derived from English.
- 3. **Reference Design:** Evaluation datasets for morphologically rich languages should include multiple reference texts to better capture acceptable lexical variation.

E.3.2. MULTI-DIMENSIONAL EVALUATION FRAMEWORK

For comprehensive assessment of generated text quality across languages, we recommend an integrated approach:

- 1. **Semantic Fidelity Assessment:** Using embedding-based metrics like BERTScore with language-specific models to verify preservation of core meaning.
- 2. **Structural Evaluation:** Employing METEOR with language-appropriate resources for stemming and synonymy to assess whether narrative structure and word order are maintained within language-specific constraints.

- 3. **Lexical Diversity Measurement:** Calculating type-token ratios or using metrics like MTLD (McCarthy & Jarvis, 2010) to quantify lexical richness relative to language norms.
- 4. **Reference-Free Quality Assessment:** Incorporating fluency and coherence metrics calibrated to the specific language being evaluated.

E.4. Case Study: Qualitative Analysis of Bengali Story Pairs

To illustrate the disconnect between lexical overlap and semantic similarity, we present a representative Bengali story pair from our dataset, alongwith the English translations in Figs. 21 and 22:

Sample Story:

একদিন, ছোট্ট রাহুল তার বন্ধুদের নিয়ে একটি ঐশ্বর্যময় ঝর্ণার কাছে গেল। ঝর্ণার পানি ঝরঝরে এবং সাদা ফেনায় ভবা ছিল...

English Translation:

One day, little Rahul went to a glorious waterfall with his friends.

The water of the waterfall was sparkling and full of white foam...

Figure 25. A sample Bengali story from the synthetic dataset

Reference Story:

একদিন, রৌদ্রজ্জ্বল সকালে, ছোট্ট রিমি তার খাতা নিয়ে বাগানে বসেছিল। সে খাতায় ছবি আঁকছিল। রিমির খুব ভাল...

English Translation:

One day, in the bright morning, little Rimi sat in the garden with her notebook.

She was drawing pictures. Rimi was very good...

Figure 26. A story for reference from the same dataset

Despite sharing the theme of a child's experience outdoors, these stories use entirely different vocabulary, characters, and settings. ROUGE metrics registered zero overlap, yet BERTScore identified high semantic similarity (0.961), recognizing the shared narrative elements and emotional tone.

E.5. Final comments

Our discovery of the zero-ROUGE phenomenon highlights the need for better evaluation frameworks for non-English languages, particularly as text generation systems prioritize semantic preservation over lexical copying.

Analysis across Hindi, Bengali, and Marathi reveals consistent patterns:

• BLEU scores remain low (<0.2)

- BERTScore values approach near-perfect (>0.95)
- METEOR scores provide a middle ground (0.07-0.33)

This contrast demonstrates how traditional lexical metrics fail to capture semantic equivalence in morphologically rich Indian languages. The pattern confirms our generation approach produces semantically coherent content with lexical diversity, rather than relying on exact phrase repetition.

Future research directions should include:

- Developing specialized metrics balancing plot preservation with stylistic variation
- Establishing multilingual benchmark datasets with multiple reference texts
- Investigating human-metric correlations for generative tasks
- Exploring reference-free evaluation approaches

F. Details on translated dataset and inference evaluations

F.1. Translation method

The original TinyStories dataset in English (Eldan & Li, 2023b) has a train split that contains 2,119,719 stories, with 320,470 duplicates, reducing unique stories to 1,799,249 (15.12% duplicates). This may impact model training, potentially skewing results.

Examples of stories appearing three times include:

- "Lily and Ben were playing in the park. They saw a fox hiding in the bushes. The fox had red fur and ..."
- "Tom is a boy who likes to play with his lab. His lab is a big, black dog who can do amazing tricks. ..."
- "Sara loved her pet cat, Lily. Lily was soft and fluffy and liked to sleep on Sara's bed. Sara liked ..."
- "Sam was sick. He had a bad cough and a sore throat. He did not like being sick. He wanted to play wi..."

Validation split has 21,990 unique stories without duplicates, ensuring reliable evaluation. However, cross-split analysis shows 6,601 stories appear in both splits, which could inflate model performance metrics. Addressing these duplicates is crucial for accurate model comparison between translated and synthetic data, enhancing the robustness of our findings in regional language modeling. Hence, we removed the duplicates and merged the two datasets. Additionally, due to resource constraints we could only translate for Hindi and Bengali. Apart from the previously mentioned issue with duplicates, we observed some data loss in translation due to issues with the freely available Google Translate API, resulting in approximately 1.9 M unique stories in Hindi and Bengali. Due to severe resource constraint, we could not translate to Marathi.

It was decided to use GPT4o for rating the translation quality based on recent reports (Kocmi & Federmann, 2023; Jiao et al., 2023) of GPT4 matching or outperforming Google Translate in case of non-English languages. In the context of evaluating translation models for Hindi and Bengali, using GPT4o, several multilingual models were assessed to identify the most effective translation method. The models include mBART (Liu et al., 2020), IndicTrans2 (Gala et al., 2023), mT5 (Xue et al., 2020), Helsinki-NLP OPUS MT (Tiedemann et al., 2023), IndicBART (Dabre et al., 2021), NLLB (Team et al., 2022), and M2M100 (Fan et al., 2021). Each model offers distinct capabilities, particularly in handling translations between English and Indic languages. GPT4o was given original English text and asked to translate to the target Indic language. Following this, it was provided with one of the machine translations and asked to rate it out of 10, using the following prompt:

I have used a machine translation model to translate the original story. On a scale of 1-10 evaluate the translation quality of the story, with respect to your translation, 1 being very bad and 10 being of the same quality as yours. Remember that "quality" here does not mean the same words but meaningfully retaining the same context and fluency. Also, point out each instance where there is a mistake in translation.

For Bengali translation, IndicTrans2 achieved an average score of 7/10 on 100 stories. Google Translate, although widely used, sometimes produced translations with contextual inaccuracies and verb errors. Still it was rated at an average score of 8 by GPT-4o. Hence it was chosen for Bengali. On the other hand, the NLLB model received an average score of 7/10. GPT-4o noted that while the NLLB translations were generally good, they lacked the fluency and natural phrasing found in Google Translate. Specific issues were identified with verb tenses and word choices, leading to minor awkwardness in sentence structures.

We used a Python library called DeepTranslate to perform Google Translate API calls. This ran on CPU hardware but we performed several optimizations to improve translation speed, namely batch processing of stories, rate limiting at 5 requests per second, and robust fault tolerance measures. These measures included automatic retries with exponential backoff, an LRU cache for successful translations, and regular intermediate saves. Quality control involved clear story demarcation, UTF-8 encoding, and progress tracking.

However, for Hindi translations both NLLB and Google Translate produced scores around 8.5. Hence, the total Hindi translation was split between these two, primarily due to resource planning regarding API calls for Google Translate and GPU time for NLLB.

Overall, these evaluations demonstrate the strengths and weaknesses of different models in translating between English and Indic languages, highlighting the importance of choosing a model that balances semantic accuracy with natural language fluency. This analysis is critical for understanding the challenges and opportunities in model translation quality, particularly in academic contexts where precise language use is essential.

F.2. Inference scores

| Dataset used | Eval Loss | Context | Completeness | Creativity | Fluency | Grammar | Overall |
|-----------------------------------|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Hindi | | | | | | | |
| Translated data Synthetic data | 1.3849 0.518 | 6.178 7.734 | 5.806 7.783 | 5.938 7.806 | 6.597 8.554 | 7.438 8.912 | 6.391 8.158 |
| Bengali | | | | | | | |
| Translated data Synthetic data | 1.4939 0.569 | 6.879 7.507 | 6.598 7.645 | 6.462 7.693 | 7.339 8.420 | 8.122 8.816 | 7.080 8.016 |

Table 14. Inference score evaluations for Hindi and Bengali in case of models trained with translated vs synthetic dataset. Model configuration: 6 layers, 8 attention heads, 512 hidden embeddings. 3000 stories were evaluated for each model. Its observed that models trained with translated data have reduced performance.

Boughorbel et al. (2024) explored how machine-translated data can be used to train language models for generating stories in Arabic, and pinpointed problems related to linguistic and cultural biases present in the translated data. They suggested additional pre-training of the models using a limited amount of high-quality synthetic data and examined the impact of this approach using Dictionary Learning techniques. They further reported that the resulting Sparse Auto-Encoders demonstrated a change in the features they learn, showing improved linguistic characteristics and less cultural bias. In any event, we have shown that training on fully synthetic data that has been generated with high quality vetting process can indeed result in significant performance improvements as shared in Table 14. The drop in quality of inference for Hindi compared to Bengali could be due to two separate translation means employed for generating Hindi translated data. We do not have the resource at present to thoroughly evaluate this.

G. WeightWatcher Analysis of Bengali Short Story Generation Model

G.1. Overview

This appendix presents a quantitative analysis of the Bengali short story generation model using WeightWatcher (WW) (Martin et al., 2021), a tool designed to assess the quality and stability of neural network weights. WW analyzes statistical properties of weight matrices to identify potential issues with training and generalization. It analyzes the quality of deep learning model layers by computing layer-specific metrics, including the α metric. This metric is based on the Heavy-Tailed Spectral Random (HTSR) theory, which suggests that well-trained layers exhibit a specific spectral density shape. In the context of WeightWatcher, a good α value for a well-trained layer is generally considered to be between 2 and 6.

G.2. Model Architecture Summary

The analyzed model is a transformer-based architecture with approximately 157M parameters, consisting of 7 transformer blocks, each with attention and feed-forward components. The model uses a 1024-dimensional embedding space and features dense linear projections throughout its architecture. In terms of overall score, this was the best performing Bengali model (Table 3). The distribution of layer α values are shown in Fig. 23 and Table 14 provides a summary of the model architecture post-training.

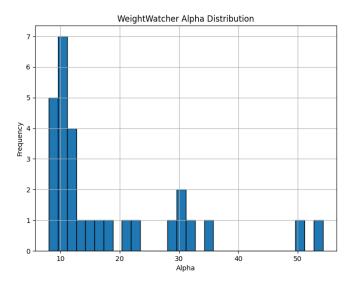


Figure 27. Distribution of α values of layers of the 157M parameter Bengali model

Table 15. Model Architecture Summary

| Metric | Value |
|----------------------|--------------|
| Under-trained Layers | 29/29 (100%) |
| Mean α | 22.51 |
| α Range | 6.76 - 89.34 |
| Mean Entropy | 0.9655 |
| Mean Spectral Norm | 1.7654 |
| Mean Stable Rank | 392.25 |

G.3. Layer Type Analysis

The model consists of one embedding layer and 28 dense layers. The dense layers include attention query, key, and value projections, attention output projections, and feed-forward network components. Table 16 presents metrics across different layer types.

Table 16. Metrics by Layer Type

| Layer Type | Count | Mean α | Mean Spectral Norm | Mean Entropy |
|------------|-------|---------------|--------------------|--------------|
| Embedding | 1 | 6.76 | 1.6231 | 0.9280 |
| Dense | 28 | 23.07 | 1.7705 | 0.9669 |

G.4. Notable Observations

G.4.1. ALPHA VALUES (α)

The Power-Law (PL) exponent α is a key metric in WeightWatcher's analysis, with higher values potentially indicating instability or over-parameterization. Our model shows generally high α values (mean: 22.51), with several extreme outliers in deeper layers of the network. Three layers exhibit particularly high values:

Table 17. Layers with Outlier Alpha Values

| Layer | α | Spectral Norm |
|----------------------------|----------|---------------|
| transformer.h.2.mlp.c_fc | 70.94 | 3.6742 |
| transformer.h.4.mlp.c_proj | 89.34 | 0.2627 |
| transformer.h.6.mlp.c_proj | 79.97 | 0.2652 |

These extreme values, particularly in later transformer blocks, suggest potential instability in the model's deeper layers. Notably, the projection layers in MLP blocks exhibit the highest α values, indicating they may be problematic components in the training process.

G.4.2. SPECTRAL NORMS

The distribution of spectral norms demonstrates a clear pattern where attention query and MLP feed-forward layers have consistently higher spectral norms (mean \approx 3.0-3.7), while attention projection and MLP projection layers have much lower values (mean \approx 0.1-0.3). This dichotomy reflects the architectural design of transformer networks where projection layers typically compress information.

G.4.3. Entropy Values

Entropy values are consistently high across all layers (mean: 0.9655), suggesting good information propagation through the network. Dense layers show slightly higher entropy values compared to the embedding layer, which is expected given their role in transforming and processing the information flow through the model.

G.5. Analysis of Transformer Blocks

Examining metrics across the 7 transformer blocks reveals interesting patterns:

- 1. Blocks 2, 4, and 6 show notably higher mean α values, suggesting potential instability in these specific blocks.
- 2. Spectral norms remain relatively consistent across blocks, indicating a stable architectural design throughout the model depth.
- 3. The number of PL spikes (indicating heavy-tailed eigenvalue distributions) decreases in deeper layers, which may indicate diminishing expressivity in later layers.

G.6. Recommendations for Model Improvement

Based on the WeightWatcher analysis, we recommend the following steps to potentially enhance model performance:

1. **Extended Training:** All layers show signs of under-training, suggesting the model could benefit from additional training epochs.

- 2. **Layer-specific Learning Rates:** Apply differential learning rates for problematic layers with extremely high α values, particularly the MLP projection layers in transformer blocks 2, 4, and 6.
- 3. **Regularization Strategies:** Consider layer-specific regularization techniques for components with outlier metrics to stabilize their behavior.
- 4. **Architecture Refinement:** Potential benefit from architectural modifications to blocks with extreme α values, such as adjusting the hidden dimensions or introducing additional normalization.

G.7. Final comments

The WeightWatcher analysis provides valuable insights into the Bengali short story generation model's internal characteristics. The model demonstrates high entropy values (mean: 0.9655), indicating effective information propagation, although several layers in transformer blocks 2, 4, and 6 exhibit extremely high alpha values (70.94-89.34), which may indicate instability in these components. We observed a clear dichotomy in spectral norms between attention query and MLP feedforward layers (3.0-3.7) versus projection layers (0.1-0.3), reflecting the architectural design of information compression in transformer networks. Based on these findings, we recommend targeted regularization strategies for outlier layers and potentially adjusting learning rates for problematic MLP projection components to enhance model stability and generative performance.