

NVIDIA and IIT Madras announce 2.8 trillion-token Indic language corpus

Source copy kept by wheresthe.ai for the story "NVIDIA and IIT Madras announce 2.8 trillion-token Indic language corpus" (<https://wheresthe.ai/d/in-nvidia-and-iit-madras-announce-2-8-trillion-token-indic-l-2026-10/>).

Source: <https://huggingface.co/blog/nvidia/scaling-indic-pretraining>

Text saved 18 Oct 2026 (the page as published).

SHA-256 of the saved text: 69a0d5e5c68a394f70ffab703eedb753e47b8872252e075f930dcc93c51408

Documents the source offers: <https://huggingface.co/datasets/nvidia/nemotron-indic-pretraining-report/resolve/main/Scaling%20Synthetic%20Pre-training%20Data%20and%20Evaluation%20for%20Indic%20LLMs.pdf>

[Back to Articles](#) a]:hidden">

Scaling Indic Pretraining: 2.8 Trillion Tokens and an Open Data Generation Pipeline

Enterprise + Article Published October 16, 2026

Upvote 28

+22

Will Jennings WillJenningsDC Follow

nvidia

Kiran Praveen kpraveen Follow

nvidia

Sriharsha Niverty sniverty Follow

nvidia

Raviraj Joshi ravirajoshi Follow

nvidia

Markus Kliegl mkliegl-nv Follow

nvidia

Yev Meyer nv-3mei Follow

nvidia

Utkarsh Vaidya uvaidya Follow

nvidia

Niket Agarwal niketagarwal Follow

nvidia

:last-child]:mb-0">

Native-script data for 22 languages, romanized data for twelve, and a reusable GPU-accelerated pipeline for translation, quality assessment, and repair.

Authors: AI4Bharat — Mohammed Safi Ur Rahman Khan, Mitesh Khapra, Anoop Kunchukuttan. NVIDIA — Niket Agarwal, Raviraj Joshi, Anusha Kamath, Markus Kliegl, Yev Meyer, Sriharsha Niverty, Kiran Praveen, Utkarsh Vaidya.

We're excited to introduce Nemotron-pretraining-Indic-v1 approximately 2.8 trillion tokens across 22 Indian languages , releasing soon alongside the open, GPU-accelerated data generation pipeline used to create it. These are the 22 languages listed in India's Constitution: the first languages for 97% of Indians, or nearly 1.2

billion citizens at the 2011 Census. Developed by NVIDIA with IIT Madras , India's premier engineering institute, and its AI4Bharat lab, this release gives model builders a substantial new resource for Indic pretraining. The pipeline captures the full generation recipe, including prompts, LLM workflows, and quality checks, so developers can adapt it to their own source material and generate more data for their needs.

AI4Bharat, IIT Madras' Indian language research lab, builds open-source datasets and models for Indian languages. Its resources, such as Sangraha , an open pretraining corpus for 22 Indian languages, have helped lay the foundation for open Indic language-model research. Our joint open-source release represents a substantial expansion of the open synthetic Indic training data that will be available to the community. It brings a broad mix of curated English material, including mathematics, code, and question-answer content, into Indian languages, giving teams more training material for building models that can explain technical concepts, work through problems, and answer questions in those languages. Every record traces back to a curated source document, and each output is checked against its input, so the synthetic text stays grounded in real information.

Mitesh Khapra, who heads AI4Bharat and made the 2025 TIME100 AI list , has named the core problem: Indian language technology trails English because "we do not have enough data for Indian languages." We're answering that call with more data, plus an open recipe for generating more.

The corpus also accounts for how people write. It includes native-script data for all 22 languages and romanized text for twelve, spanning 34 language-script configurations. Our workflows combine document-level translation, quality assessment, and targeted repair, with checks to preserve content such as equations, tables, and code. This data complements naturally authored Indic text, which remains essential for local knowledge and cultural context.

We're sharing the open generation recipe so the community can not just reproduce the data but take this work further. Teams can use the corpus directly, adapt the prompts and quality checks, or bring their own source documents to the workflow. By opening up both the data and the process, we aim to make large-scale Indic data generation more accessible and enable more developers to contribute to stronger models across Indian languages.

Continued pretraining for Nemotron 3.5 Super VL added 10 billion Indic tokens, drawn from a twelve-language subset of this corpus and natural Indic text. Our technical report on the model covers the full story. We plan to build on this corpus in future Nemotron models.

What is in the corpus?

Figure 1. The corpus contains 2,601.0 billion native-script tokens and 224.6 billion romanized tokens. Counts use the Gemma-4 tokenizer. Document records include translations and script variants of shared sources; they are not counts of unique English documents.

The corpus has two groups of language configurations:

Native-script and romanized data: Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Nepali, Odia, Punjabi, Tamil, Telugu, and Urdu.

Native-script data: Assamese, Bodo, Dogri, Kashmiri, Konkani, Maithili, Manipuri, Sanskrit, Santali, and Sindhi.

Here, native-script describes the writing system: the text is still translated from English. Romanized means Indic-language text written in Latin script, not an English translation. For each of the twelve dual-script languages, romanization covers a subset of the accepted native-script translations.

This distinction matters when constructing a training mixture. Native and romanized records offer different script exposure, but can express the same underlying content. The token counts describe the retained corpus, including accepted rewrites, rather than a training run's token budget. Counts also depend on the tokenizer used, so corpus inventory and training-token counts should be read separately.

From English source documents to Indic training data

To build this corpus, we start with documents from 15 Nemotron-CC-v2 subsets, covering web text, question-answer material, mathematics, code, and other specialized content. We use NeMo Data Designer , NVIDIA's open-source framework for generating synthetic data, to run translation, quality checks, and targeted repair. Throughout the workflow, we retain each source identifier so developers can trace an output back to its original document. Clear provenance like this is part of what makes open data practical to audit and build on. Figure 2

shows the workflow for twelve languages.

Figure 2. Four-stage Indic synthetic-data pipeline for twelve languages, implemented using NeMo Data Designer and Gemma-4-31B-it. We translate English documents, assess translation quality, repair low-scoring outputs, and convert a subset of accepted translations into Latin script. The other ten languages use the separate Indic Translate workflow described below.

The twelve-language workflow

For these twelve languages, we translate documents in full so the model can use context across paragraphs, tables, and equations. We selected Gemma-4-31B-it by comparing models on complete documents, checking translation quality and preservation of technical content. Our prompts distinguish prose to translate from content to preserve, including code, equations, URLs, and formatting.

After translation, we use the same model to assess each output against its English source. A rubric inspired by Multidimensional Quality Metrics (MQM), an open framework for classifying translation errors by type and severity, covers meaning, fluency, terminology, style, and formatting. We retain only outputs scoring at least 75 out of 100 (a rubric score, not an accuracy percentage). For lower-scoring outputs, we use the assessment feedback to guide repairs, then reassess them against the same threshold. This process recovered 438.4 billion tokens across native-script and romanized data, already included in the corpus total. Sampled human reviews and tests with deliberately introduced errors helped identify omissions and residual English, informing improvements to our prompts and checks.

Accepted native-script translations also provide the starting point for romanized data. We convert a subset into Latin script and check each result against its native-script input. This separates translation from script conversion. In our initial human comparison, this two-step approach produced fewer hallucinated additions and less residual English than generating romanized text directly from English.

The ten-language workflow

For the other ten languages, we adapt the workflow to address long-document quality and script-stability issues observed in our comparisons. We translate with Indic Translate, a document-level translation model for English and 22 Indian languages from Bodhan AI, a nonprofit center for AI in education incubated at IIT Madras. Gemma-4-31B-it then assesses each translation against its English source. For eight languages, we keep the whole-document approach. For longer Santali documents, we translate overlapping windows and reconstruct the document; for Manipuri, we translate into Bengali script before converting to Meitei Mayek, its native script. These workflows retain outputs scoring at least 75 out of 100 and produce native-script data only, without feedback-based rewriting.

What changed after Indic continued pretraining?

The Nemotron 3.5 Super VL technical report compares the checkpoint immediately before Indic continued pretraining (CPT) with the checkpoint after that stage. The configured budget was 100 billion training tokens, with 10% (10 billion tokens) allocated to a mixture of synthetic and natural Indic text. The remaining 90% maintained the preceding training mixture's relative proportions.

Scores increased on all five reported Indic knowledge and reasoning benchmarks: MILU, Indic MMLU-Pro, Indic ARC-Challenge, Indic ARC-Easy, and Indic GSM8K.

Figure 3. Nemotron 3.5 Super VL before and after the Indic CPT stage. Gains range from 4.98 to 9.82 percentage points. The comparison reports the change across a training stage that combines synthetic and natural Indic data, rather than isolating the synthetic corpus's contribution. Language coverage differs by benchmark; these are stage-level checkpoints, not a comparison of final model releases.

Source: Nemotron 3.5 Super VL technical report, Table 2. GSM8K excludes Odia because evaluation exceeded the serving context limit.

These gains support the combined synthetic-and-natural Indic CPT recipe, while leaving the choice of mixture for other models and languages to further study.

The companion Indic report examines source mixtures, synthetic-data pool size, and script coverage using first-pass accepted data for the twelve-language group. Evaluating the full corpus, including recovered rewrites and the other ten languages, remains a next step for those controlled studies.

Broader Indic capabilities in Nemotron 3.5 Super VL

Beyond the CPT-stage results, how does the released model compare with the previous generation? Figure 4 compares Nemotron 3.5 Super VL with Nemotron 3 Super across Indic task performance and language quality. These results reflect the released model's full training pipeline, including Indic CPT and post-training.

Nemotron 3.5 Super VL scores higher on five of the six task benchmarks shown, spanning knowledge, reasoning, mathematics, code, and translation. Instruction following on IFEval 1.5 dipped 2.04 points. It also improves across all four IndiVibe dimensions: usefulness, fluency, language and script fidelity, and appropriate verbosity. IndiVibe is Sarvam AI's open benchmark of assistant-style prompts in Indian languages. Together, these evaluations examine both what the model can do and how well it communicates in Indian languages.

Figure 4. Indic task performance and language quality across Nemotron Super generations. Left: six task benchmarks. Right: preliminary IndiVibe conditional preference scores, not direct head-to-head win rates. Labels show changes in percentage points, except WMT24++, which uses chrF points. See the companion Indic technical report for benchmark and evaluation details.

Start building

Load the corpus

Start with the languages and scripts your model needs, inspect a sample, and review the dataset card and license before training. Configurations use language, script, and region tags: for example, hi-Deva-IN denotes Hindi in Devanagari, while hi-Latn-IN denotes romanized Hindi.

The example below asks for the repository ID from the dataset card, then discovers the available splits and streams a sample.

```
from datasets import get_dataset_split_names, load_dataset

dataset_id = input ( "Repository ID from the dataset card: " ).strip() config = "hi-Deva-IN"

splits = get_dataset_split_names(dataset_id, config_name=config) print ( "Available splits:" , splits)

data = load_dataset( dataset_id, name=config, split=splits[ 0 ] , streaming= True ) print ( next ( iter (data)))
```

Hugging Face Datasets supports streaming so readers can inspect records without downloading the entire corpus. Once the release is available, pin a dataset revision for reproducible experiments.

For a first training study , compare mixtures at a fixed token budget and record the natural/synthetic and native/romanized shares. Track per-language results alongside English capability retention and language/script quality. When creating held-out splits, account for records derived from the same source across languages and scripts.

To build a fuller recipe, pair the corpus with Nemotron post-training data, such as the Chat & Instruction or Post-Training Blends collections. For India-specific synthetic data, seed NeMo Data Designer with Nemotron-Personas-India , synthetic personas in English and Hindi grounded in real-world distributions for India.

Adapt the generation recipe

The open-source Nemotron Bharat translation plugin for NeMo Data Designer provides translation and evaluation prompts, optional feedback-based rewriting, document windowing, and script-remapping support. Use its configuration reference to choose models and languages, adapt prompts to your domain, and set rewrite parameters. The accompanying developer note explains the workflow and engineering choices.

Start with a small sample of your own source documents. Inspect translation quality and preservation of code, mathematics, and formatting, then refine the prompts and checks with native-language reviewers before scaling up. Treat generation and corpus acceptance as separate steps: the plugin returns candidates and scores, while your filtering policy determines which records enter the training corpus.

Scope and limitations

Translation broadens access to the source material, but does not replace text originally written in Indian languages or guarantee coverage of local knowledge and culture. We recommend combining this corpus with appropriately sourced natural Indic text and evaluating the resulting mixture for the intended use case.

Quality assessment is primarily automated, complemented by sampled human review whose coverage varies across languages. For the twelve-language workflow (Figure 2), the translator also serves as the judge. Retained text can still contain errors, and the ten-language workflow needs broader native-speaker validation. Source-ID checks remove processing duplicates; they are not semantic deduplication of similar content.

Alongside the corpus, AI4Bharat at IIT Madras, with support from NVIDIA, is building Indic-LLM-Eval-Suite, an open and evolving collection of benchmarks for Indic model capabilities. It helps model builders assess both task performance and language quality. See the companion blog, [Better Models Need Better Benchmarks: Introducing Indic-LLM-Eval-Suite](#).

Explore the language configurations, test a training mixture, or adapt the open recipe to a new domain. Sharing per-language results and improvements to the generation workflow can help the community build on this work.

Further reading and acknowledgments

The companion reports, [Scaling Synthetic Pre-training Data and Evaluation for Indic LLMs](#) and the [Nemotron 3.5 Super VL and Lightning technical report](#), provide the corpus accounting, generation methods, experiments, and evaluation details. Evaluation is the other half of this work. For the benchmark side, see the companion blog from [AI4Bharat and NVIDIA](#), [Better Models Need Better Benchmarks: Introducing Indic-LLM-Eval-Suite](#).

This work brings together NVIDIA, IIT Madras, AI4Bharat, and Bodhan AI. We also acknowledge the model and tooling contributions used in the pipeline: Google's Gemma, Bodhan AI's Indic Translate, and NeMo Data Designer. Additional NVIDIA data resources are available in the Nemotron Pre-Training Datasets collection .

Models mentioned in this article 4

Datasets mentioned in this article 5

Collections mentioned in this article 2

More from this author

Fine-Tune ASR for Your Domain with NVIDIA Speech Agent Skills

12 October 13, 2026

One Model Family, Two Gold-Level Results: Fine-Tuning Nemotron for IOI and IMO

24 October 7, 2026

Community sepertinya menarik, aku pasti akan mencobanya</p>

p>\n","updatedAt":"2026-10-16T02:29:42.860Z","author":{"_id":"6ad182b0f8ca93986a17c5d5","avatarUrl":"https://cdn-avatars.huggingface.co/v1/production/uploads/noauth/EZ8NWzEv3llbD7YTXH_se.jpeg","fullName":"Jay Krist","name":"BawiLewu","type":"user","isPro":false,"isHf":false,"isHfAdmin":false,"isMod":false,"isUserFollowing":false,"language":"id","probability":0.8887125253677368},"editors":["BawiLewu"],"editorAvatarUrls":["https://cdn-avatars.huggingface.co/v1/production/uploads/noauth/EZ8NWzEv3llbD7YTXH_se.jpeg"],"reactions":[],"isReport":false}},{"id":"6ad4693d8a17f270f4ba17c3","author":{"_id":"69ad99f8a52f9c0bb9ccae81","avatarUrl":"https://cdn-avatars.huggingface.co/v1/production/uploads/69ad99f8a52f9c0bb9ccae81/dgpXpb6wpH4kTEao0pth3.jpeg","fullName":"Vikrant Maharshi","name":"vikrantmaharshi","type":"user","isPro":false,"isHf":false,"isHfAdmin":false,"isMod":false,"followerCount":0,"edited":false,"hidden":false,"latest":{"raw":"sdk","html":"<p>sdk</p>"},"updatedAt":"2026-10-18T06:37:49.025Z","author":{"_id":"69ad99f8a52f9c0bb9ccae81","avatarUrl":"https://cdn-avatars.huggingface.co/v1/production/uploads/69ad99f8a52f9c0bb9ccae81/dgpXpb6wpH4kTEao0pth3.jpeg","fullName":"Vikrant Maharshi","name":"vikrantmaharshi","type":"user","isPro":false,"isHf":false,"isHfAdmin":false,"isMod":false,"followerCount":0,"language":"en","probability":0.5783829092979431},"editors":["vikrantmaharshi"],"editorAvatarUrls":["https://cdn-avatars.huggingface.co/v1/production/uploads/69ad99f8a52f9c0bb9ccae81/dgpXpb6wpH4kTEao0pth3.jpeg"],"reactions":[],"isReport":false}}],"status":"open","isReport":false,"pinned":false,"locked":false,"collection":"community_blogs"},"correlation":{"WillJenningsDC","kipraveen","sniverty","ravarajoshi","mkliegl-nv","nv-3mei"},"uvaidya","niketagarwal"},"primaryEmailConfirmed":false,"discussionRole":0,"acceptLanguages":["en"],"withThread":true,"cardDisplay":false,"repoDiscussionsLocked":false,"hideComments":true}>

BawiLewu 3 days ago

sepertinya menarik, aku pasti akan mencobanya

[See translation](#)

[Reply](#)

vikrantmaharshi about 9 hours ago

sdk

[Reply](#)

[Edit Preview](#)

Upload images, audio, and videos by dragging in the text input, pasting, or clicking here . Tap or paste here to upload images

[Comment](#) · [Sign up or log in to comment](#)

[Upvote](#) 28

+16

[Models mentioned in this article](#) 4

[Datasets mentioned in this article](#) 5

[Collections mentioned in this article](#) 2