

Vemana: a 278M Telugu language model trained from scratch

UDGAM LABS VERSION 1.0

Evidence labels: [MEASURED] our own runs; [ESTIMATE] our projections; [EXTERNAL] other people's cards, papers or pages, not reproduced by us; [EXPLORATION] ideas and options. A label covers the sentence, list or table it closes or heads.

Summary

Vemana by Udgam Labs is a 278M-parameter Telugu base language model. We trained it from scratch, with its own 32,000-token Telugu tokenizer, on public Telugu web text, and release its weights, every 10% checkpoint, GGUF files and the code that scores it.

- **What it is.** 278,214,144 parameters in the Gemma 4 text layout, trained from random initialisation on 5.10B unique tokens of Telugu web text from Sangraha, FineWeb-2 and HPLT 3.0, read about 2.35 times (12.0B token positions). One H100, about 16 hours, about \$69 of GPU time, on 26-27 October 2026, live in public. [MEASURED]
- **Main result.** On 416 Telugu Wikipedia pages created between 1 September and 22 October 2026, after all but one of the compared models were released (Section 7.1), Vemana scores **0.361 bits per byte** (lower is better). The best other open base model under 1B parameters that we tested, nanochat-telugu-560M, twice its size, scores 0.383; the paired 95% interval of the gap is -0.024 to -0.021 . [MEASURED, 27 Oct 2026]
- **It holds on the checks.** Vemana is also ahead of nanochat-telugu-560M on 29 pages created 23-26 October, after every model here (-0.018), on 477 pages created in July and August (-0.023), at a 512-token window (-0.017) and without the editor who created 36% of the main pages (-0.023). [MEASURED]
- **The rules came first.** The test, the 0.02 bits-per-byte margin, the robustness checks and the choice of the final checkpoint as the release model were fixed before training. Vemana passes both pre-registered rules: best open Telugu base model under 300M, and under 1B. [MEASURED]
- **Where it falls short.** Sarvam-1 (2.5B parameters, about 9× larger) is level on the main pages (gap $+0.005$, interval -0.001 to $+0.010$) and better on the July-August pages ($+0.020$). On Telugu multiple-choice tests of reading, knowledge and inference Vemana is at or near chance, like most open models under 1B we tested. It makes things up, it repeats itself, its context is 1,024 tokens, and it is a base model with no instruction or safety tuning. [MEASURED]

1. Why a small Telugu model

Telugu has its own script, and many of its words are long: a stem with several endings. Tokenizers built for many languages at once cut those words into many small pieces. On held-out Telugu

web text, Gemma 3's tokenizer needs 2.86 tokens per word, Qwen3's 11.2 and LFM2.5's 19.3 [MEASURED, Gate 0, 23 Oct 2026]. Every extra piece costs a step of computation and a slot of the context window.

Small general-purpose models are correspondingly weak at Telugu. On our main test, Gemma-3-270m, about Vemana's size, scores 0.585 bits per byte, Qwen3-0.6B-Base 0.692 and Gemma-3-1B 0.484, against Vemana's 0.361 [MEASURED].

We wanted a Telugu model small enough to run on a laptop's CPU or inside a browser tab, open enough to fine-tune for Telugu tools, and documented well enough to study how a small model learns one language. Our release rule asks for a measured edge over the newest open models of the same size, measured in our own harness on the release date, with the claim's scope stated exactly. This report describes how Vemana was built and how that edge was measured.

The model is named after Vemana, a Telugu poet known for short, plain verses [EXTERNAL].

2. What we release

ITEM	WHAT IT IS	WHERE
Vemana-Base	The final checkpoint (step 22,888), float32, with the tokenizer; Apache-2.0	huggingface.co/udgamlabs/Vemana-Base
Vemana-Checkpoints	Twelve checkpoints: every 10% of the run, the 25% check model and its stable copy, and the final model; each with its per-page scores (<code>fresh.json</code>) and training state (<code>train_state.json</code>)	huggingface.co/udgamlabs/Vemana-Checkpoints
Vemana-Base-GGUF	Q8_0 (297 MB) and F16 (558 MB) files for llama.cpp and the browser	huggingface.co/udgamlabs/Vemana-Base-GGUF
Space	A web demo	huggingface.co/spaces/udgamlabs/Vemana
Code	The test builder, the scorer, the paired comparison, the GGUF builder and checker, the browser demo, the release-gate results	github.com/udgamlabs/vemana
Model page	Results and an in-browser demo that runs on the visitor's device	udgamlabs.com/models/vemana
Training replay	The live training page, kept as a replay	udgamlabs.com/live/vemana

The released `model.safetensors` is identical (sha256) to the final checkpoint of the training run, and the 84 checkpoint files are identical by sha256 to the run's own copies [MEASURED]. The test pages themselves are not redistributed: Telugu Wikipedia is CC BY-SA 4.0, and `build_te.py` rebuilds the sets from Wikipedia's history.

3. Data

Sources. Public Telugu web text from three open datasets [EXTERNAL licences, read at source on 27 Oct 2026]:

DATASET	PUBLISHER	LICENCE	WHAT IT IS
Sangraha, <code>verified/tel</code>	AI4Bharat	CC-BY-4.0	the human-verified part of its Telugu text (web pages and OCR'd documents)
FineWeb-2, <code>tel_Telu</code>	Hugging Face	ODC-By 1.0; use also subject to Common Crawl's terms of use	filtered Telugu from 96 Common Crawl dumps, summer 2013 to April 2024
HPLT 3.0, <code>tel_Telu</code>	HPLT project	CC0 for the packaging	Telugu from Internet Archive and Common Crawl crawls, 2012-2024

Processing (as the release facts state it). The text was filtered for Telugu script and length, near-duplicates were removed across the three sources, and it was decontaminated against the te-2026-10 test pages and the multiple-choice test sets (Belebele/FLORES, MILU, IndicXNLI, IndicCOPA). The two newer test sets were built on 27 October, after the data job; their pages were written in 2026, and the training text comes from web crawls up to 2024. [MEASURED, data job `data-1`]

What was left out. No Telugu Wikipedia dump (a 2026 dump would contain test pages) and no machine-translated text. FineWeb-2 contains older Telugu Wikipedia pages at their natural share; none of them can be a test page, because every test page was created in 2026.

Size. 5.10B unique tokens after processing, read about 2.35 times in training (12.0B token positions) [MEASURED].

4. Tokenizer

Vemana has its own tokenizer: SentencePiece BPE with 32,000 tokens, byte fallback (so any text can be written), digits split one by one, line breaks kept and no Unicode normalisation. It was trained on one shard of Sangraha `verified/tel` (1.2M lines) and contains no third-party vocabulary. The released `tokenizer.json` is byte-identical to the one used in training. [MEASURED]

How many pieces a Telugu word becomes

Tokens per word, lower is better · 3,000 held-out web documents

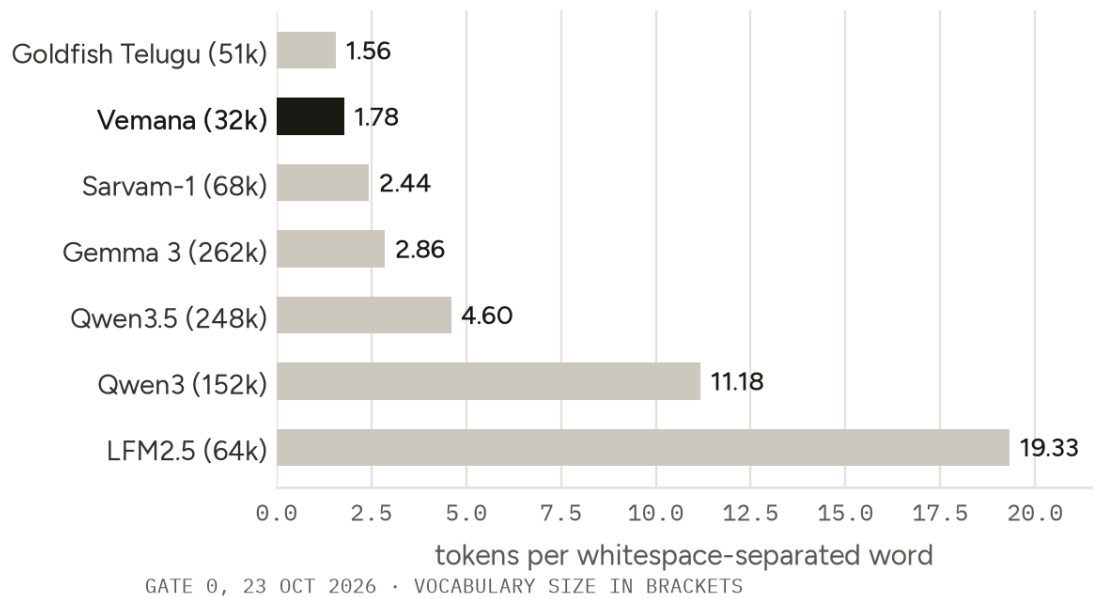


FIGURE 1. TOKENS PER TELUGU WORD ON HELD-OUT WEB TEXT. GATE 0, 23 OCT 2026.

TOKENIZER (VOCABULARY)	FLORES DEVTEST	BELEBELE PASSAGES	WIKIPEDIA (416 PAGES)	FINWEB-2 TEST	SANGRAHA SHARD 30
Vemana (32k)	1.73	1.78	1.99	1.78	1.76
Goldfish Telugu (51k)	1.47	1.49	1.57	1.56	1.65
Sarvam-1 (68k)	2.14	2.16	2.51	2.44	2.58
Gemma 3 (262k)	2.84	2.82	3.14	2.86	2.99
Qwen3.5 (248k)	4.63	4.64	5.05	4.60	4.78
Qwen3 (152k)	11.40	11.38	11.95	11.18	11.52
LFM2.5 (64k)	19.83	19.72	20.71	19.33	19.93

Tokens per whitespace-separated word; none of the five samples is in the tokenizer's training shard. [MEASURED, Gate 0]

- Vemana's tokenizer uses 36-41% fewer tokens per word than Gemma 3's across the five samples (38% on web text), and 27% fewer than Sarvam-1's on web text. Goldfish's tokenizer is more compact still. [MEASURED]

- On the 416 main test pages Vemana's tokenizer averages 10.95 bytes per token; nanochat-telugu-560M's byte-level tokenizer averages 5.34, so it needs about twice as many tokens per page. [MEASURED, release-1]
- A 32,000-token vocabulary costs 32.8M parameters at Vemana's width, 12% of the model. Gemma-3-270m spends about 170M of its 268M parameters on its 262k-token vocabulary for many languages [EXTERNAL, Google's Gemma 3 270M announcement]. And because Telugu text needs fewer tokens, a 1,024-token window holds about 1.6 times more of it than with Gemma 3's tokenizer [MEASURED ratio, web text].

5. Model

Vemana uses the Gemma 4 text layout and loads in transformers 5.19 or newer as `Gemma4ForCausalLM`. The values below are those of the released `config.json`.

ITEM	VALUE
Parameters	278,214,144: 245.4M in the transformer blocks, 32.8M in the tied input and output embedding
Layers × width	18 × 1,024; MLP width 3,584, gated GELU (tanh)
Attention	8 query heads and 2 key-value heads of dimension 128; full attention in every layer; RoPE base 1,000,000
Output	tied embeddings; final logit soft-cap 15
Vocabulary	32,000 (Section 4)
Context	1,024 tokens trained and tested, about 570 Telugu words. The config allows 4,096; nothing beyond 1,024 was trained or tested

The weights were trained from random initialisation; no pretrained weights, Gemma weights or Gemma tokenizer are involved. Training compute was about 2.3×10^{19} FLOPs for the main run (1.895 GFLOP per token position, counted from the model, × 12.0B positions), plus about 5% for the 25% check copy [MEASURED count].

6. The training run

Run. 22,888 steps, 12.0B token positions, on one H100 in three segments of at most 8 hours, about 16 hours in all, on 26-27 October 2026, for about \$69 of GPU time. Each segment saved its full resume state to cloud storage and the next one resumed from it; in a test at a tiny shape, one uninterrupted run and two segments gave identical final weights. [MEASURED]

Before the run. A learnability check at the real shape in the GPU preflight (the model must memorise four rows and decode them back exactly), resume and export-parity checks, and an independent review of the training job by a separate AI agent (12 findings, all fixed before launch). [MEASURED]

Live in public. Anyone could follow the run at udgamlabs.com/live/vemana and on an X live stream: the share of training done, the score on the fresh pages, and the model's continuations of fixed Telugu prompts at each milestone. The page stays up as a replay.

Checkpoints. The stable weights were exported and scored with the release scorer at every 10% of the run. At 25%, a copy branched off at 20% was cooled down early to give a finished model at 3.0B positions; it was measured and shown but, by a decision on 26 October, did not stop or pause the run. The release model, the final checkpoint, was fixed in advance; we did not pick a checkpoint by its score.

Fresh-text score during training

Bits per byte on the 416 Sep-Oct pages, every 10% of the run · lower is better

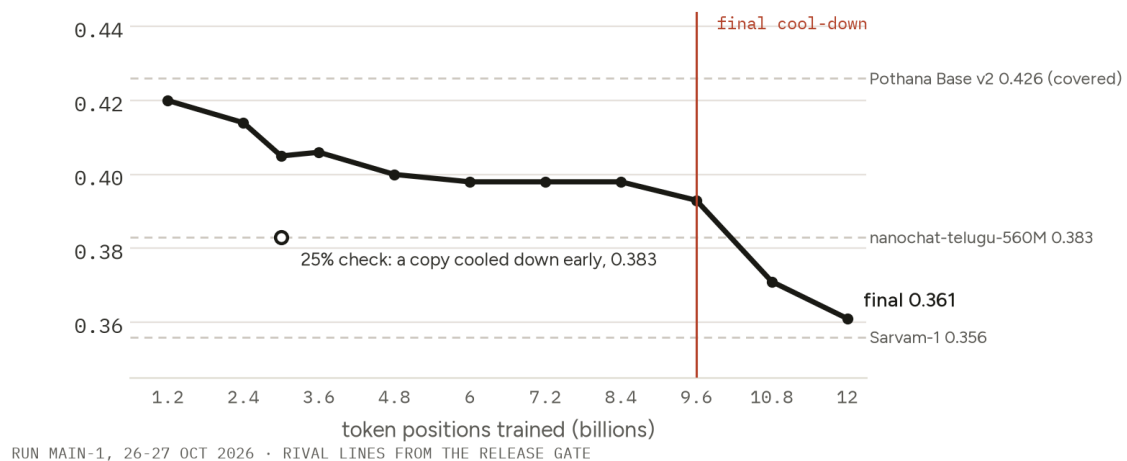


FIGURE 2. BITS PER BYTE ON THE 416 MAIN TEST PAGES AT EVERY CHECKPOINT. RUN MAIN-1, 26-27 OCT 2026.

CHECKPOINT	STEP	TOKEN POSITIONS	BITS PER BYTE (416 PAGES)
10%	2,289	1.2B	0.420
20%	4,578	2.4B	0.414
25%, stable	5,722	3.0B	0.405
25% check (cooled down from 20%)	5,722	3.0B	0.383
30%	6,866	3.6B	0.406
40%	9,155	4.8B	0.400
50%	11,444	6.0B	0.398
60%	13,733	7.2B	0.398
70%	16,022	8.4B	0.398
80% (start of the final cool-down)	18,310	9.6B	0.393
90%	20,599	10.8B	0.371
Final = Vemana-Base	22,888	12.0B	0.361

[MEASURED, run main-1; same scorer and pages as the release gate]

- Most of the gain comes in the cool-down. The stable checkpoints level off near 0.398 between 40% and 70% of the run, and the last 20% brings the score to 0.361.
- The 25% check model, cooled down early, already scored 0.383 at a quarter of the tokens, level with nanochat-telugu-560M. Its cool-down was worth 0.022; at the halfway point, the pre-registered formula (stable score at 50% minus that drop) projected 0.376 for the final model [ESTIMATE]. The final model came in at 0.361.

7. How we test

7.1 Fresh text

A score means little if the model has seen the test. We test on Telugu Wikipedia pages created after the compared models were released. One script (`build_te.py`) and one rule set build every set: new mainspace pages only; no redirects, recreated pages, lists or bot-created pages; at least 120 words of prose; at least 85% of letters in Telugu script; every page re-screened for deletion. No page is in two sets.

SET	PAGES CREATED	PAGES	BYTES	EDITORS (TOP EDITOR'S PAGES)	ROLE
te-2026-10	1 Sep-22 Oct 2026	416	4,476,833	47 (150)	main test, built 23 Oct (Gate 0)
te-2026-10b	23-26 Oct 2026	29	285,496	9 (16)	the newer set, defined before training, built 27 Oct
te-2026-summer	1 Jul-31 Aug 2026	477	6,001,285	33 (210)	supplementary, larger, built 27 Oct

[MEASURED]

Every compared model was released before these pages were written, except Dhee-Pravega-OG Atom: it was released on 8 October 2026 and does not state its data sources or cut-off, so the July-August and early-September pages could be in its training data, which could only help it. The 23-26 October pages are after every model. [EXTERNAL release dates]

7.2 Bits per byte

For each page, the scorer adds up the bits the model needs for every token, $-\log_2 p(\text{token} \mid \text{preceding text})$, and the totals over all pages are divided by the pages' UTF-8 bytes:

bits per byte = $\sum \text{over pages and tokens of } -\log_2 p(\text{token} \mid \text{context}) \div \sum \text{over pages of UTF-8 bytes}$

Bytes do not depend on the tokenizer, so models with different tokenizers can be compared; lower is better. The same unchanged scorer (`score.py`) ran for every model: float32 weights, TF32 matrix multiplies, a 1,024-token window (Goldfish 512, its limit) sliding by half a window, the model's beginning-of-sequence (BOS) token first when it has one, every token scored exactly once.

7.3 Paired page bootstrap

Intervals are 95% percentile intervals from 4,000 resamples of the pages, with replacement (seed 0). For a gap between two models, each resample draws one set of pages and scores **both** models on it, so the interval reflects only how the two models differ on the same text. A gap is Vemana minus the other model: negative means Vemana is better. For every gap in the claim, Vemana was better in all 4,000 resamples. [MEASURED]

7.4 Rules fixed before the result

We wrote two win rules into the run plan before training; both are tested here. A rival's score for the bar is its better score where it has two (the covered-text score in Section 7.6), which favours the rival.

RULE	WRITTEN	WIN CONDITION
A. Best open Telugu base model under 300M	25-26 Oct, before training	Vemana $\leq$ 0.406 on te-2026-10, and the paired 95% interval of the gap to Pothana Base v2 (its better score, 0.426) below zero
B. Best open Telugu base model under 1B	25 Oct, before the Step 0 rival scoring and before training	Vemana at least 0.02 below B, the best open base model under 1B in the release-day scan, with the interval below zero; and the interval also below zero at a 512-token window, without the top editor's pages, and on pages created after 22 October

The order of events. Rule B was written first. Step 0 then scored the 2026 Telugu-only models (Section 7.5) and found nanochat-telugu-560M at 0.383. A win under rule B then needed about 0.363, and our projections put every affordable model size at 0.37-0.41 [ESTIMATE]. So the run plan aimed at rule A, "under 300M", with Pothana Base v2's 0.426 as the bar, fixed before training. The final model passed both. Because the scope we state was chosen after the result, between two rules written before it, we report both. The stated claim uses rule B, which covers more rivals and has the stricter conditions.

7.5 Which rivals

Three scans, each in our own harness:

- **Gate 0 (23 Oct):** the open multilingual and Telugu models under 1B we could find, and Sarvam-1 (2.5B) as a larger reference.
- **Step 0 (25 Oct):** five 2026 Telugu-only base models that Gate 0 had missed: nanochat-telugu-560M and four Pothana models.
- **Release-day scan (27 Oct):** Hugging Face searches by name and language, every repository changed since 20 October in the Telugu and Indic labs' organisations, every text model created since 20 October by about 90 major labs, two web searches, and a search of X posts. It found no new Telugu base model since 25 October and one model both earlier scans had missed: Dhee-Pravega-OG Atom (645.6M parameters, 18 languages, trained from scratch, released 8 October). [EXTERNAL, scan of 27 Oct 2026]

Left out, with reasons: Jaddangi-Ai-Telugu-1B (no published tokenizer, so it cannot be scored; also 0.7% over 1B); telugu-llama-small (64M; no published tokenizer); telugu-gpt-moe-sangraha-10k (298M, a mixture-of-experts model that needs its own code, trained on 0.33B tokens; not scored); instruction-tuned variants (we compare base models); Gemma-4-E2B (5.1B parameters in total); a romanised-Telugu model, toy models of a few million parameters, repositories without weights, and models that are not causal text base models (a sequence-to-sequence rewriter, a vision-language model, a speculative-decoding head). [EXTERNAL]

7.6 Loading rivals fairly

Scoring someone else's model is easy to get subtly wrong, and most mistakes make a model look worse than it is. We checked every model's tokenizer round trip (text to tokens and back) on all 416 pages, and loaded the weights into standard transformers classes rather than running repositories' own model code.

MODEL	BYTES PER TOKEN	PAGES THAT ROUND-TRIP EXACTLY	NOTE
Vemana	10.95	416 of 416	
nanochat-telugu-560M	5.34	416 of 416	BOS id fixed (below)
Dhee-Pravega-OG Atom	9.20	0 (116 ignoring whitespace)	NFKC normalisation (below)
Pothana Base v2	12.55	0	3.6% of bytes hidden in <code><unk></code> (below)
Goldfish Telugu 1000mb	13.93	0 (83 ignoring whitespace)	
Sarvam-1	8.69	416 of 416	
Gemma-3-1B	6.96	416 of 416	

On te-2026-10. [MEASURED, release-1]

A tokenizer that drops or merges characters never has to predict them, which can only flatter its model; every lossy one here is still behind Vemana.

- **Pothana's `<unk>`**. Pothana's published tokenizer turns words outside its vocabulary into `<unk>`: 3.6% of the main test's bytes (4.1% and 3.9% on the other sets), mostly ordinary Telugu words. The model spends about 13 bits on each `<unk>`, so this hurts it. We therefore also compute a **covered-text** score (bits minus the bits spent on `<unk>` targets, divided by bytes minus the hidden bytes) and use the better of the two for the bar. Pothana base-300M, whose published pipeline hides 25.2% of the bytes, is scored (0.555) but not treated as comparable. Pothana's model class computes exactly what transformers' `Qwen3ForCausalLM` does (logit difference 0.0 on random weights), so its weights load without its code. [MEASURED, Step 0]
- **Dhee's normalisation**. Dhee-Pravega-OG Atom's tokenizer (`nmt_nfkc`) turns every zero-width non-joiner (U+200C, common in Telugu) and every line break into a space, so it never predicts them: none of the 416 pages decodes back exactly. We also scored Vemana on a copy normalised the same way: Vemana 0.360, Dhee 0.394, gap -0.034 [-0.036, -0.033], so Vemana's lead holds either way. Its weights load into transformers' `01mo2ForCausalLM` with per-head Q/K norms; logits equal those of its own code exactly. We scored the revision released under Apache-2.0 (`ad71ca4a70`). [MEASURED]

- **nanochat's BOS token.** nanochat-telugu-560M's config names token 0 as its BOS, but the model was trained with `<|bos|>` (id 65527), and its tokenizer adds no BOS by itself. We put `<|bos|>` first, for the fresh text and for the multiple-choice tests; on all 416 pages the ids equal `[65527]` followed by the repository tokenizer's own ids. [MEASURED]
- **Goldfish's window.** Goldfish models accept 512 tokens, so Goldfish is scored at 512, and rule B also checks Vemana at 512.

7.7 Control

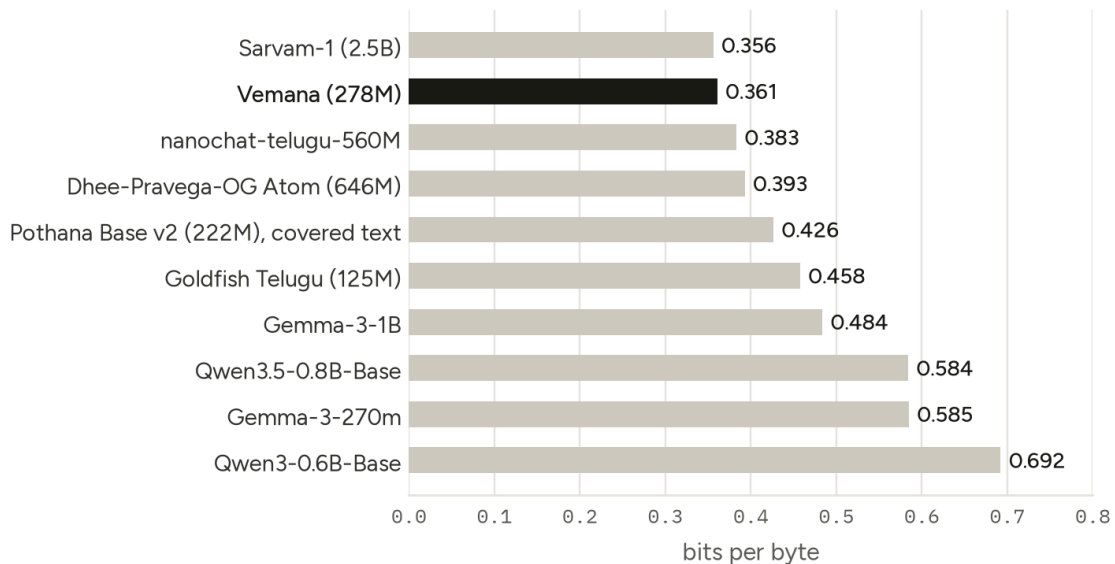
In the release run, every model that had been scored before and was scored again (five rivals and Vemana) had its per-page result on te-2026-10 compared with the earlier one: the five rivals reproduced exactly (identical tokens, largest relative difference in a page's bits 0.0), and Vemana's per-page bits were within a relative 6.7×10^{-5} of its scores on the training machine. Gate 0 had already shown bit-identical scores on shared pages from two different GPU machines. [MEASURED, release-1]

8. Results

8.1 The main test

Predicting new Telugu text

Bits per byte, lower is better · 416 Telugu Wikipedia pages, Sep-Oct 2026



UDGAM LABS, 27 OCT 2026 · SAME SCORER AND PAGES FOR EVERY MODEL

FIGURE 3. BITS PER BYTE ON THE 416 MAIN TEST PAGES. POTHANA BASE V2 AT ITS COVERED-TEXT SCORE; POTHANA STAGE-A-PLUS, POTHANA SP-BASE-300M, GPT-2-TELUGU1 AND LFM2.5-230M ARE IN THE TABLE, NOT DRAWN. RELEASE GATE, 27 OCT 2026.

MODEL	PARAMETERS	BITS PER BYTE [95% CI]	VEMANA MINUS MODEL [95% CI]	WITHOUT THE TOP EDITOR (266 PAGES): GAP
Sarvam-1	2.5B	0.356 [0.351, 0.361]	+0.005 [-0.001, +0.010]	+0.012 [+0.004, +0.019]
Vemana	278M	0.361 [0.353, 0.367]	—	(Vemana 0.373)
nanochat- telugu-560M	561M	0.383 [0.376, 0.390]	-0.022 [-0.024, -0.021]	-0.023 [-0.026, -0.021]
Dhee- Pravega-OG Atom	646M	0.393 [0.385, 0.400]	-0.032 [-0.034, -0.031]	-0.035 [-0.037, -0.033]
Pothana Base v2	222M unique (370M with its blocks run twice)	0.440 [0.433, 0.448]; covered text 0.426	-0.080 [-0.081, -0.078]; vs covered -0.066 [-0.067, -0.064]	-0.083 [-0.085, -0.081]
Pothana stage-a-plus	230M unique	0.450; covered 0.436	-0.090 [-0.092, -0.088]	-0.093 [-0.095, -0.091]
Goldfish Telugu 1000mb	125M	0.458 [0.450, 0.465]	-0.097 [-0.099, -0.095]	-0.096 [-0.100, -0.093]
Pothana sp- base-300M	387M unique	0.462	-0.102 [-0.105, -0.099]	-0.106 [-0.111, -0.101]
Gemma-3-1B	1.0B	0.484 [0.474, 0.493]	-0.123 [-0.127, -0.119]	-0.126 [-0.131, -0.121]
Qwen3.5- 0.8B-Base	0.87B	0.584	-0.223 [-0.228, -0.219]	-0.220 [-0.227, -0.215]
Gemma-3- 270m	268M	0.585	-0.224 [-0.228, -0.220]	-0.229 [-0.235, -0.224]
Qwen3-0.6B- Base	0.6B	0.692	-0.332 [-0.336, -0.327]	-0.328 [-0.334, -0.322]
gpt-2-telugu1	98M	0.786	-0.425 [-0.436, -0.414]	-0.402 [-0.416, -0.389]
LFM2.5- 230M-Base	230M	1.709	-1.348 [-1.355, -1.341]	-1.335 [-1.342, -1.327]

[MEASURED. Vemana, nanochat-telugu-560M, Dhee, Pothana Base v2, Goldfish, Gemma-3-1B and Sarvam-1 scored in the release run on 27 Oct; the others scored 23-25 Oct in the same

harness on the same pages. Paramanu-Telugu (208M) scored 2.52 in Gate 0, or 1.58 on a copy without punctuation, the form it was trained on.]

Both rules pass. Rule A: 0.361 against Pothana Base v2's covered-text 0.426, gap -0.066 [-0.067, -0.064], well under the 0.406 bar. Rule B: B is nanochat-telugu-560M at 0.383; the margin is 0.022, at least the 0.02 required, with the interval below zero, and every robustness part holds (Sections 8.2 and 8.3). [MEASURED]

The claim. Vemana (278M) predicts new Telugu text better than any other open base model under 1 billion parameters that we tested. On 416 Telugu Wikipedia pages created in September and October 2026, it scores 0.361 bits per byte, against 0.383 for the next best, nanochat-telugu-560M, twice its size (paired 95% interval of the gap -0.024 to -0.021). It also leads on pages created in July-August 2026 and on pages created 23-26 October. Same scoring script and pages for every model; measured by Udgam Labs on 27 October 2026.

8.2 Newer and older pages

MODEL	23-26 OCT (29 PAGES)	VEMANA MINUS MODEL	JUL-AUG (477 PAGES)	VEMANA MINUS MODEL
Sarvam-1	0.337 [0.313, 0.359]	+0.009 [-0.009, +0.022]	0.365 [0.360, 0.370]	+0.020 [+0.017, +0.024]
Vemana	0.346 [0.311, 0.372]	—	0.385 [0.379, 0.392]	—
nanochat-telugu-560M	0.364 [0.330, 0.388]	-0.018 [-0.021, -0.015]	0.408 [0.402, 0.414]	-0.023 [-0.024, -0.022]
Dhee-Pravega-OG Atom	0.371 [0.336, 0.397]	-0.025 [-0.030, -0.022]	0.414 [0.408, 0.420]	-0.029 [-0.030, -0.028]
Pothana Base v2	0.425; covered 0.410	-0.079 [-0.088, -0.069]; vs covered -0.065 [-0.074, -0.055]	0.456; covered 0.444	-0.071 [-0.073, -0.069]; vs covered -0.058 [-0.060, -0.057]
Goldfish Telugu 1000mb	0.435	-0.090 [-0.097, -0.083]	0.480	-0.095 [-0.096, -0.093]
Gemma-3-1B	0.441	-0.095 [-0.108, -0.080]	0.502	-0.117 [-0.120, -0.113]

[MEASURED, release-1, 27 Oct 2026]

Vemana against the three closest models

Paired 95% intervals, 4,000 page resamples · red line: no difference

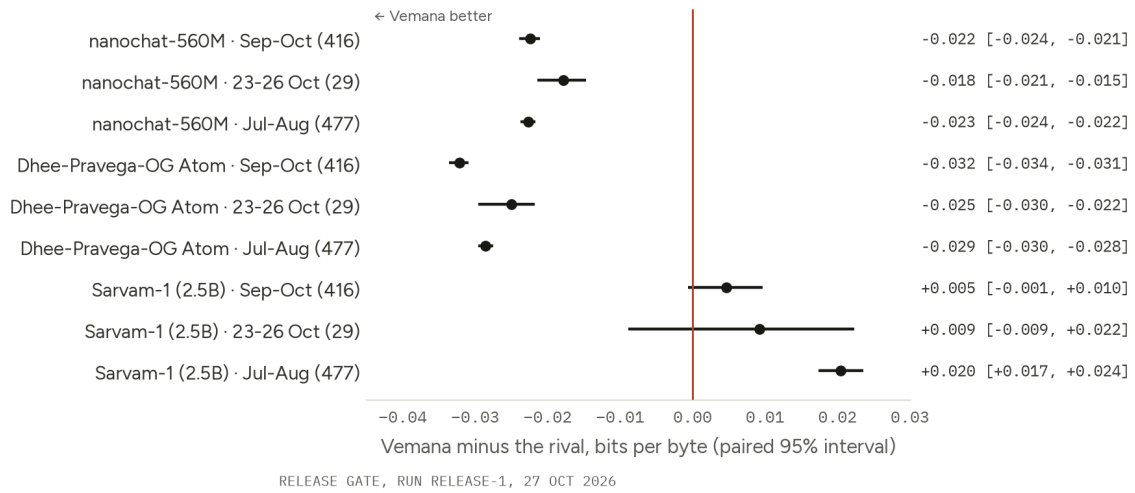


FIGURE 4. VEMANA MINUS EACH OF THE THREE CLOSEST MODELS ON ALL THREE SETS; SARVAM-1 IS ABOUT 9× LARGER. RELEASE GATE, 27 OCT 2026.

The order of the models is the same on all three sets. The 29 newest pages give wide intervals for each model's own score, but the paired gaps are much narrower, because both models are scored on the same resampled pages.

8.3 Checks on the main test

Rule B, part by part

Black: parts of the rule · grey: reported only · red line: no difference

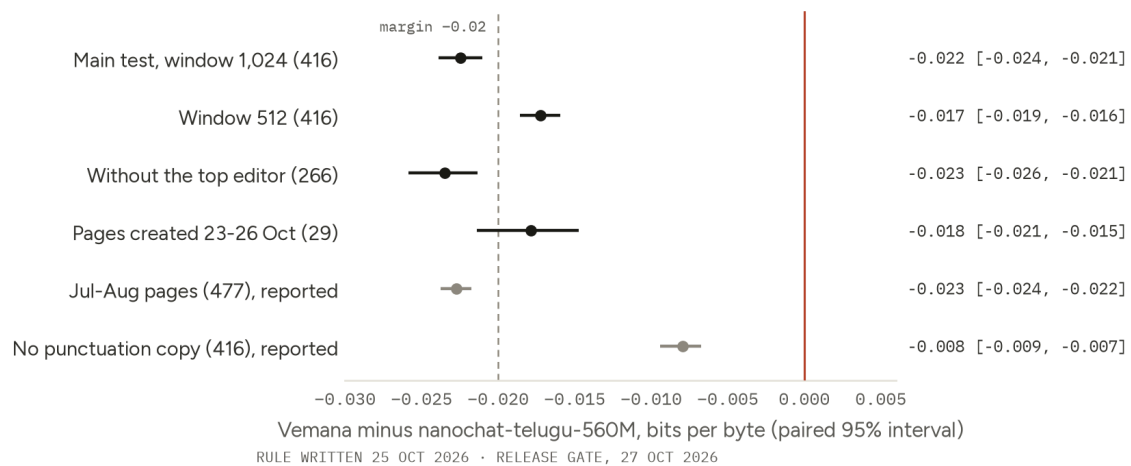


FIGURE 5. VEMANA MINUS NANOCHAT-TELUGU-560M, PART BY PART. RELEASE GATE, 27 OCT 2026.

- **Window 512** (Vemana at 512 tokens, rivals at their own windows): Vemana 0.366; gap to nanochat -0.017 [-0.019, -0.016], to Dhee -0.027 [-0.029, -0.026], to Sarvam-1 +0.010

[+0.004, +0.015]. [MEASURED]

- **Without the top editor** (one editor created 150 of the 416 pages): Vemana 0.373; gap to nanochat -0.023 [-0.026, -0.021]. [MEASURED]
- **Without ASCII punctuation or line breaks** (1.1% of the bytes; a diagnostic copy kept since Gate 0, reported only): Vemana 0.373, nanochat 0.381, gap -0.008 [-0.009, -0.007]. The lead narrows: nanochat is relatively better on punctuation-free text. Pothana sp-base-300M 0.401 (gap -0.028), Dhee 0.402 (-0.030), Sarvam-1 0.373 (-0.000 [-0.006, +0.005]). [MEASURED]
- **Dhee's normalisation**: on the copy normalised as Dhee's tokenizer does, Vemana 0.360 and Dhee 0.394, gap -0.034 [-0.036, -0.033]. [MEASURED]

8.4 Multiple-choice tests

The Gate 0 harness: lm-evaluation-harness 0.4.13, zero-shot, each model's own BOS token first, float32 with TF32. "Letter" forms score the answer letter (accuracy); "cloze" forms score the answer text (length-normalised accuracy). 95% intervals from an item bootstrap.

TASK (ITEMS; CHANCE)	VEMANA	NANOCHAT- 560M	DHEE	GOLDFISH	GEMMA- 3-1B	QWEN3.5- 0.8B	SARVAM-1 (2.5B)
Belebele-te, letter (900; 25%)	25.1 [22.3, 28.0]	23.0	22.4	22.7	24.9	35.6	54.0
Belebele-te, cloze (900; 25%)	26.1 [23.3, 29.0]	26.8	27.0	24.2	25.9	26.8	37.6
MILU-te, letter (7,304; 25%)	25.3 [24.3, 26.3]	25.3	24.8	25.2	25.0	31.1	40.1
MILU-te, cloze (7,304; 25%)	30.2 [29.2, 31.3]	29.3	27.2	27.0	27.1	26.3	36.9
IndicXNLI-te (5,010; 33.3%)	35.0 [33.7, 36.3]	34.9	36.7	32.9	35.5	34.9	46.7
IndicCOPA-te (500; 50%, floor about 55%)	58.4 [54.2, 62.8]	60.4	59.0	56.6	57.4	55.6	63.6

Accuracy, %. On IndicCOPA, picking the shorter answer scores 54% and a random model scored 55.8%. [MEASURED, release-1 and Gate 0, same harness]

- Vemana is at chance on Belebele and MILU (letter form) and near chance on IndicXNLI (35.0% against 33.3%), like the other open models under 1B, except the Qwen models on the letter forms: Qwen3.5-0.8B scores 35.6% on Belebele and 31.1% on MILU (gaps -10.4 [-14.4, -6.3] and -5.8 [-7.3, -4.4] points). Both Qwen models are weak at predicting Telugu text (Section 8.1) and most likely carry reading-test skills over from their large multilingual training [EXPLORATION].
- On MILU's cloze form Vemana scores 30.2% (chance 25%). We report it and make no claim from it.
- Sarvam-1 is far above chance, so the tests work; they need a larger model, or a Telugu test built to separate small ones.
- Dhee's tokenizer copy for lm-eval matches its SentencePiece ids on 70,177 of 70,186 task strings. [MEASURED]

8.5 What it writes

Two continuations from the model card (sampling with temperature 0.8 and top-p 0.95; the prompt in bold; the English translations are ours):

వర్షాకాలంలో సీజనల్ వ్యాధులు ఎక్కువగా వస్తాయి. అందువల్ల ఆరోగ్యంగా ఉండటం చాలా ముఖ్యం. మీరు వర్షాకాలంలో డెంగ్యూ, మలేరియా, టైఫాయిడ్, వైరల్ ఇన్ఫెక్షన్ల వంటి సీజనల్ వ్యాధుల బారిన పడే ప్రమాదం ఉంది. ...

In the rainy season, seasonal illnesses are more common, so staying healthy matters a lot. In the rainy season you are at risk of seasonal illnesses such as dengue, malaria, typhoid and viral infections. ...

కంప్యూటర్ అనేది మనకు అత్యంత అవసరమైన వస్తువులలో ఒకటి. కంప్యూటర్లు మనం చేసే చాలా పనులను చేస్తాయి, ... ఈ సాధనాలలో ఒకటి CPU. ...

A computer is one of the things we need most. Computers do many of the tasks we do, ... One of these tools is the CPU. ...

9. Where it falls short

- **A larger model is still better.** Sarvam-1 (2.5B parameters, about 9× larger) is level with Vemana on the main pages (+0.005 [-0.001, +0.010]) and on the 23-26 October pages (+0.009 [-0.009, +0.022]), and better without the top editor (+0.012 [+0.004, +0.019]), at a 512-token window (+0.010 [+0.004, +0.015]) and on the July-August pages (+0.020 [+0.017, +0.024]). [MEASURED]
- **Knowledge and reasoning tests are at or near chance** (Section 8.4); Qwen3.5-0.8B is better on the letter forms of Belebele and MILU. [MEASURED]

- Without punctuation, the lead over nanochat-telugu-560M narrows to 0.008 bits per byte. [MEASURED]
- It makes things up. Sampled on a CPU on 27 October (temperature 0.8, top-p 0.95), it wrote fluently that Telugu Language Day falls on 21 September; it is 29 August [EXTERNAL]. [MEASURED]
- It repeats itself. A short-story prompt ("ఒక రోజు ఒక రైతు", one day a farmer) turned into the same sentence about a farmer and a cow, repeated with small changes. With greedy decoding it often repeats a phrase; the training-state files in Vemana-Checkpoints show this on several of the fixed prompts. In the release run's greedy samples it repeated less than some rivals: on "తెలుగు భాష", Pothana, Goldfish and Dhee repeat a phrase within 40 tokens. [MEASURED]
- Base model only. It continues text; it does not answer questions or follow requests, and it has no safety tuning. It can produce false, offensive or harmful text.
- 1,024-token context, about 570 Telugu words.

10. Running Vemana: CPU, GGUF and the browser

transformers on a CPU. With the card's quick-start code in float32 on a Mac mini's CPU (four threads), Vemana writes about 65-68 new tokens per second after the first call. [MEASURED, 27 Oct]

GGUF for llama.cpp. llama.cpp's Gemma 4 code expects at least one sliding-window attention layer, and Vemana uses full attention in every layer, so a plain conversion aborts when loaded. Our build makes two metadata changes and leaves the weights untouched:

1. Layer 0 is marked as sliding-window with a window of 4,096 tokens, the file's whole context, and the same rotary and head settings as the other layers, so the computation is unchanged.
2. The tokenizer is stored as Vemana's SentencePiece model (same ids as the released `tokenizer.json`), so llama.cpp splits text exactly as transformers does.

CHECK AGAINST TRANSFORMERS (FLOAT32), 27 OCT 2026	F16 (558 MB)	Q8_0 (297 MB)
Tokenization, 40 Telugu Wikipedia pages	identical on 40	identical on 40
Perplexity on the same 4,080 tokens (transformers: 15.764)	15.764	15.765
Greedy continuations, 4 prompts × 40 tokens	identical on 4	identical on 3; one differs after a few words (8-bit rounding)

[MEASURED, `check_gguif.py`]

In the browser. The model page runs the Q8_0 file inside the browser with wllama 3.6.1, llama.cpp compiled to WebAssembly: the file downloads once and the text never leaves the device. In that engine, prompt token counts and two greedy continuations were identical to the original model's. On a Mac mini in Chromium it writes about 100 tokens per second. Ollama and LM Studio were not tested. [MEASURED]

11. Limitations and next steps

Limits of the test.

- The test covers one kind of Telugu: new Wikipedia pages, written in an encyclopedic register. News, conversation, older literature and dialects are not measured.
- One editor created 36% of the main pages; we report every result without that editor's pages as well.
- Pages written with machine translation or a language model cannot be told apart reliably; none of the main test's pages carried Wikipedia's content-translation tag.
- A new page can reuse older text from the web that is in some model's training data. That would affect any model, not only ours, and we did not measure it.
- Dhee-Pravega-OG Atom's training-data cut-off is not stated (Section 7.1).

Limits of the model. Section 9.

Next steps. [EXPLORATION]

- **A chat model** that answers questions and follows requests in Telugu needs Telugu instruction data under a licence that allows training and release. Our count found only about 61,000 such rows, about 43,000 of them open-ended chat [MEASURED, Gate 0]; most larger Telugu instruction sets come from models or licences that bar this use [EXTERNAL]. Paths: translating permissively licensed English instruction data with a permissively licensed translator, or generating data with a permissively licensed model and keeping a provenance tag on every row. It would be its own project, with its own measured bar before any money is spent.
- **A Telugu test that separates small models.** The multiple-choice tests sit at or near chance for most models below 1B, so they cannot show progress at this size.
- **Use Vemana as a base:** fine-tuning for Telugu tools (summarisation, classification), continued pretraining, and studies of how a small model learns a language from its checkpoints.

12. Licences

Vemana. Weights and tokenizer under Apache-2.0, copyright 2026 Udgam Labs LLP, with a `NOTICE` that credits the training data. The model is the Gemma 4 text layout as implemented in transformers (Apache-2.0); no Gemma weights, tokenizer or Gemma terms are involved.

Training data [EXTERNAL, read at source on 27 Oct 2026]: Sangraha by AI4Bharat (CC-BY-4.0); FineWeb-2 by Hugging Face (ODC-By 1.0; use also subject to Common Crawl's terms of use); HPLT 3.0 by the HPLT project (CC0 for the packaging; the HPLT project states it does not own the underlying text, and takedown requests go to the project). None of the three restricts training models or releasing their weights; two ask for attribution, which the card and `NOTICE` give. The web text itself keeps its own copyright, as for every model trained on web text.

Test pages. Telugu Wikipedia (CC BY-SA 4.0), used as evaluation data only and not redistributed; `build_te.py` rebuilds the sets.

Rivals were scored at pinned revisions under their own licences: nanochat-telugu-560M (AGPL-3.0, `dd616ef5`), Pothana Base v2 (Apache-2.0, `e7123f27`), Goldfish Telugu 1000mb (Apache-2.0, `c4929bac`), Dhee-Pravega-OG Atom (Apache-2.0 at `ad71ca4a70`; the same weights carry the Dhee Research-Only Licence from a later revision, whose terms keep earlier revisions under Apache-2.0), Gemma-3-1B (Gemma Terms of Use, `fcf18a2a`), Sarvam-1 (Sarvam AI Research License, `e9607337`), and the Qwen, Gemma-3-270m and LFM2.5 models under their own licences.

13. Citation

```
@misc{udgamlabs2026vemana,
  title = {Vemana: a 278M Telugu language model trained from scratch},
  author = {Prasad, Jitesh K},
  year = {2026},
  howpublished = {Udgam Labs},
  url = {https://huggingface.co/udgamlabs/Vemana-Base}
}
```

Thank you to AI4Bharat, Hugging Face and the HPLT project for the open data Vemana stands on, and to the Telugu Wikipedia editors whose new pages made a fair test possible.

14. References

1. AI4Bharat. Sangraha: a large Indic-language pretraining corpus (`ai4bharat/sangraha`), 2024.
2. Hugging Face. FineWeb-2 (`HuggingFaceFW/fineweb-2`), 2024.

3. HPLT project. HPLT datasets v3.0 (hplt-project.org/datasets/v3.0), 2025.
4. Google. Gemma 3 technical report, 2025; Introducing Gemma 3 270M, 2025; Gemma 4 model family, 2026.
5. A. Karpathy. nanochat. github.com/karpathy/nanochat, 2025.
6. T. Kudo and J. Richardson. SentencePiece: a simple and language independent subword tokenizer and detokenizer for neural text processing, 2018.
7. Goldfish: monolingual language models for 350 languages ([goldfish-models](#)), 2024.
8. Model cards of the rivals at the pinned revisions above: viswamaicoe/nanochat-telugu-560M, dvitvaai/pothana-base-v2-225M, dheeyantra/dhee-pravega-og-atom, sarvamai/sarvam-1, google/gemma-3-1b-pt, google/gemma-3-270m, Qwen/Qwen3.5-0.8B-Base, Qwen/Qwen3-0.6B-Base, LiquidAI/LFM2.5-230M-Base.
9. L. Bandarkar et al. The Belebele benchmark, 2024. MILU (AI4Bharat), 2024. IndicXNLI, 2022. IndicCOPA (IndicXTREME), 2023.
10. EleutherAI. lm-evaluation-harness, version 0.4.13.
11. ggml-org. llama.cpp; ngxson. wllama.

Appendix A. Training recipe and data mix

The recipe behind the run, for anyone who wants to reproduce or build on it.

A.1 Data mix and processing

SOURCE	UNIQUE TOKENS AFTER PROCESSING	SHARE	NOTES
Sangraha verified/tel	2.90B	57%	all shards except the validation shard data-30
HPLT 3.0 tel_Telu	1.33B	26%	quality bins 10-7 (bins 5 and 6 left out)
FineWeb-2 tel_Telu	0.86B	17%	
Total	5.10B		read about 2.35 times

[MEASURED, data job data-1 ; shares rounded]

- **Filters:** at least 70% of letters in Telugu script and at least 30 words per document.
- **Near-duplicates across sources:** a document is removed when at least half of its long paragraphs, or its whole text, already appeared in an earlier document, in the order validation documents, Sangraha, FineWeb-2, HPLT.
- **Decontamination:** the same index as our earlier Telugu proxy runs, covering the te-2026-10 pages, Belebele, FLORES+, MILU-te, IndicXNLI-te and IndicCOPA-te.
- **No re-weighting:** every kept document counts once per pass, with a new shuffle each pass.
- **Source check before training:** a 44M proxy model scored held-out documents at 0.413 bits per byte for Sangraha, 0.485 for FineWeb-2 and 0.440 for HPLT; 20 random kept documents per source showed no junk. [MEASURED]

A.2 Optimiser and schedule

SETTING	VALUE
Code	our nanochat fork, Gemma 4 port, exported to Hugging Face <code>Gemma4ForCausalLM</code> with a logit-parity check
Optimiser	Muon for matrices, AdamW for embeddings and scalars
Batch	524,288 tokens per step (512 rows × 1,024)
Steps	22,888 (12.0B token positions)
Learning rate	warm-up over 500 steps (0.26B tokens), a stable phase, then a cool-down over the last 20% of steps (from step 18,310)
Weight decay	0.125 by the fork's scaling rule, used at half (0.0625), held constant
Muon momentum	0.85 → 0.97 over the first 400 steps, → 0.90 over each cool-down
Precision	bf16 compute with fp32 weights; no gradient clipping
Data reading	rows read contiguously from the shuffled token stream (no document-aligned packing)
Not used	token superposition (no gain in a proxy test while reading 1.75× the data), weight averaging (EMA), FP8

[MEASURED, run plan and the run's training-state files]

A.3 How the run was sized

- **Gate 0 (23 Oct):** the bar then was Goldfish's 0.458; two projections put a 150-300M model at 0.38-0.44 [ESTIMATE].
- **Proxy-1 (24 Oct):** a 44M model on 0.5B tokens of Sangraha with a plain recipe (Llama layout, AdamW) reached 0.512 [MEASURED].
- **Step 0 (25 Oct):** nanochat-telugu-560M at 0.383 [MEASURED] put the under-1B win level at about 0.363, beyond the 0.37-0.41 projected for every affordable model size [ESTIMATE].
- **Proxy-2 (25 Oct):** the same 44M model, data and tokens on our training stack reached 0.493, 0.019 better than the plain recipe ([−0.020, −0.018]), at 1.8× the training speed; token superposition added nothing (−0.0003, interval spanning zero) [MEASURED].
- **Projection for 280M × 12B positions:** 0.389 by the central method, with a bad case of 0.418; the other method projected 0.333 [ESTIMATE]. The final model scored 0.361, better than the central projection and its bad case. [MEASURED]

A.4 Stop rules inside the run (as planned, then set aside)

The plan stopped the run at 25% if the check model scored 0.426 or more, paused it from 0.421 to 0.426, and paused at 50% if the projected final score was above 0.406. On 26 October we decided to train to the end whatever these readings said; they were still measured and shown. The 25% check model scored 0.383 and the halfway projection was 0.376, so neither would have stopped the run. [MEASURED]

About Udgam Labs. Udgam Labs is an independent AI research lab. We work from first principles: before we build, we understand why things work, whether we are training a model, fine-tuning one or applying AI to a new problem. Udgam (उद्गम) is Sanskrit for origin. udgamlabs.com