Where Hindi meets English inside five open LLMs

Script, language and token cost in 1,000 conversations in eleven versions of Hindi and its neighbours

tp53 `inventcures.github.io`

18 November 2026 New edition (draft 0.5, adds four models). The first edition, draft 0.3, is unchanged.

Abstract

Hindi and Urdu are close to one spoken language written in two scripts, and Hindi and Marathi are two languages written in one. This lets script and language be varied separately, which is rarely possible. I wrote 1,000 short conversations across 24 domains in English and ten other versions (everyday, formal and romanized Hindi; Hinglish in two scripts; Urdu in two scripts; Bhojpuri; Marathi; Chinese) and read the residual stream of Qwen3-1.7B at every layer. Three results. (1) After removing each version's mean state, an English conversation's nearest Hindi state is its own translation 55% of the time at layer 13, against 0.1% chance, falling to 15% at layer 28; all versions share this rise and fall. (2) With script held fixed, the language closer to Hindi gains 0.09 in linear CKA at the last layer (Devanagari Urdu 0.66 against Marathi 0.57); with language held fixed, the shared script gains 0.10 (Devanagari Urdu against Perso-Arabic Urdu, 0.56). The two effects are about equal, so Perso-Arabic Urdu and Marathi tie. (3) Romanized Hindi is the outlier: at the last layer it scores below 0.5 with every other version, including the Devanagari text it was transliterated from. Separately, Hindi costs 4.95 times as many Qwen3 tokens as English, which factors into a byte premium of 2.87 and a tokens-per-byte premium of 1.73; across 12 tokenizers the ratio ranges from 7.9 (GPT-3) to 1.2 (Sarvam-1). All non-English text is machine-translated and unchecked by native speakers; a 12-item reading found meaning errors in half the Urdu items. I then ran the same 11,000 conversations through Qwen3.8-27B, Gemma 4 31B, gpt-oss-120b and Sarvam-105B. Every model shows the mid-depth peak in Hindi–English retrieval, from 55% (Qwen3-1.7B) to 99% (Gemma 4). But results (2) and (3) do not generalise: in the four larger models both the script and the language gain lie within 0.09 of zero at the last layer, and romanized Hindi is second-closest to Hindi in two of them. Token cost correlates with peak retrieval ($r = -0.68$, $n = 5$), but Sarvam and Gemma charge Hindi the same 1.3 tokens per English token and align it at 66% and 99%.

1 Introduction

A multilingual model has to decide, at each layer, how much of a sentence's representation is about what it means and how much is about how it is written. Work on Llama, Claude and others has converged on a three-phase picture: early layers track surface form, middle layers share a space across languages, and late layers convert back to one language's tokens [1, 3, 4, 5]. That picture has mostly been tested on language pairs that differ in script, vocabulary, grammar and training share all at once, such as English and Chinese. When they separate, one cannot say which difference did it.

Hindi offers controls that most languages do not. Urdu is nearly the same spoken language in Perso-Arabic script [8], and can also be written in Devanagari. Marathi shares Devanagari but is a different language. Romanized Hindi keeps the words and changes the script and the token count. This paper asks one question:

Inside a small decoder LLM, does a change of script move a conversation's representation more or less than a change of language?

Three hypotheses are possible. **H1, script-dominant**: versions sharing Hindi's script are closer to it whatever their language. **H2, language-dominant**: versions sharing Hindi's language are closer whatever their script. **H3, additive**: both matter, with effects of comparable size. The data favour H3.

Contributions.

1. A parallel set of 1,000 items in 11 versions across 24 domains, including three script-only transliteration pairs, with code to regenerate it (Section 3).
2. Layer-by-layer retrieval and CKA for all 55 version pairs in Qwen3-1.7B (Section 4), with two controlled comparisons that separate script from language.
3. A decomposition of the token premium into byte and vocabulary parts, and a comparison of 12 tokenizers on the same conversations (Section 6).
4. The same measurements on four larger open models, showing which Qwen3-1.7B results hold across models and which do not (Section 5).

2 Related work

Phases across depth. Language-specific neurons concentrate in the first and last layers [1, 2]. The logit lens shows Llama-2's middle layers decoding concepts in English for non-English prompts [3], and attribution graphs in Claude 3.5 Haiku show language-agnostic middle features whose shared fraction grows with scale [4]. Patching middle-layer concepts across languages carries meaning while keeping the target language [5].

Language as an offset. In multilingual encoders each language occupies a shifted copy of a shared space [6], and removing per-language means raises translation retrieval while making language identification harder [7]. Mean-centred retrieval below follows that idea.

Script against language. Transliterating Indic languages into one script raises their representational similarity, measured with CKA on parallel sentences [9]. Liu et al. measured Hindi–Urdu similarity in encoders and showed that transliteration raises similarity even for unrelated pairs, so a matched-minus-shuffled comparison is needed [10]. A 2026 study reports that multilingual models encode script over linguistic structure [11], and Serbian in two scripts gives a parallel case [12]. In decoder LLMs, Devanagari Hindi generation passes through romanized subwords in late layers [13], and a single direction can flip output script [14]. I found no matched study of Hindi and Urdu inside a decoder LLM.

Tokenizer premiums. Petrov et al. define the premium of language A over B as the ratio of token counts for parallel text and trace its effects on cost, latency and context [16]. Ahia et al. compare API cost across languages grouped by script [17], and Arnett et al. separate out a byte premium, listing Hindi at 2.37 over English on their corpus [18]. Deng et al. measure the energy cost directly [19].

Similarity measures. Linear CKA [20] compares the inner-product structure of two sets of paired representations and stays informative when the number of points is below the dimension, as here (n = 1,000, d = 2,048). Its reliability has limits [21], and alternatives tailored to cross-lingual comparison exist [22].

3 Data and method

3.1 Conversations

Qwen3-32B (vLLM 0.9.2, seed 0, thinking off) wrote 240 topics across 24 domains, five rooted in everyday life in India and five universal per domain, and five English exchanges per topic, one per question type: explain, how-to, feedback on a plan, choose between options, and a personal problem. User turns have 15 to 35 words and replies 40 to 70. The same model translated each exchange into seven versions and transliterated three more from those, so that each pair differs only in script (Table 1). An item was kept only if every version passed a script check (at least 85% of letters in the expected script, 60% for Chinese) and a length check (0.25 to 3 times the English length, 0.12 for Chinese). Of 1,200 items, 1,080 passed in every version, and 1,000 were kept with 39 to 42 per domain (559 universal, 441 Indian).

Table 1: The eleven versions. Rows marked † are transliterations of the row named in brackets.

Code	Version	Tokens / English
`en`	English, the source	1.00
`hi-deva`	everyday Hindi, Devanagari	4.95
`hi-shuddh`	formal, Sanskritized Hindi	5.14
`hi-latn`†	everyday Hindi in Roman letters [`hi-deva`]	2.13
`hien-latn`	Hinglish, Roman	1.67
`hien-deva`†	Hinglish, Devanagari [`hien-latn`]	5.11
`ur-arab`	Urdu, Perso-Arabic	3.64
`ur-deva`†	Urdu, Devanagari [`ur-arab`]	5.09
`bho-deva`	Bhojpuri	4.66
`mr-deva`	Marathi	4.99
`zh`	Simplified Chinese	0.99

Translation quality. I read the user turn of 12 randomly drawn items in every version (Table 2). This is a reading by a model, not a native-speaker check, and 12 items give only a rough rate. Errors are mostly single words; script and register were right in every item read.

Table 2: Items with a meaning error in a 12-item reading. Six items were read for the transliterated versions and Chinese.

Version	Errors	Example
Chinese	0 of 6	none
Hindi, everyday	1 of 12	Ayushman Bharat rendered as अयोग्य भारत, “unfit India”
Hindi, formal	1 of 12	the same
Hinglish, Roman	2 of 12	one question changed meaning; one item left in English
Bhojpuri	3 of 12	mixed Hindi, Bhojpuri and Maithili forms
Marathi	3 of 12	“inter-caste marriage” rendered as marriage within one’s caste
Urdu, Perso-Arabic	6 of 12	JEE as “JE six”; music genre as جنس, sex or gender
Urdu, Devanagari	5 of 6	inherits the Perso-Arabic errors

3.2 Read-outs and metrics

Qwen3-1.7B (28 layers, $d = 2{,}048$, tied embeddings) ran in bfloat16 over each conversation, teacher-forced through the chat template. I take the residual-stream state at the last token of the assistant reply at each layer, and drop the first token, whose largest activation is about 1,170 times a typical token’s from layer 3 onward [23]. Let $X_a \in \mathbb{R}^{n \times d}$ hold the states of version a, row i being item i.

Mean-centred retrieval from a to b is the fraction of items i for which $\arg\max_j \cos(x_{a,i} - \bar{x}_a,\, x_{b,j} - \bar{x}_b) = i$. Chance is $1/n = 0.1\%$. The raw version omits the centring.

Linear CKA [20] with column-centred X_a, X_b is

$$\mathrm{CKA}(X_a, X_b) = \frac{\|X_b^\top X_a\|_F^2}{\|X_a^\top X_a\|_F \, \|X_b^\top X_b\|_F}.$$

A logistic probe on the top 64 principal components gives version identity, and R^2 of log token count on the top 10 components gives the length confound. Maps use PCA frames fit on balanced samples; they illustrate and carry no claim.

4 Results

4.1 Retrieval rises in the middle and falls at the end

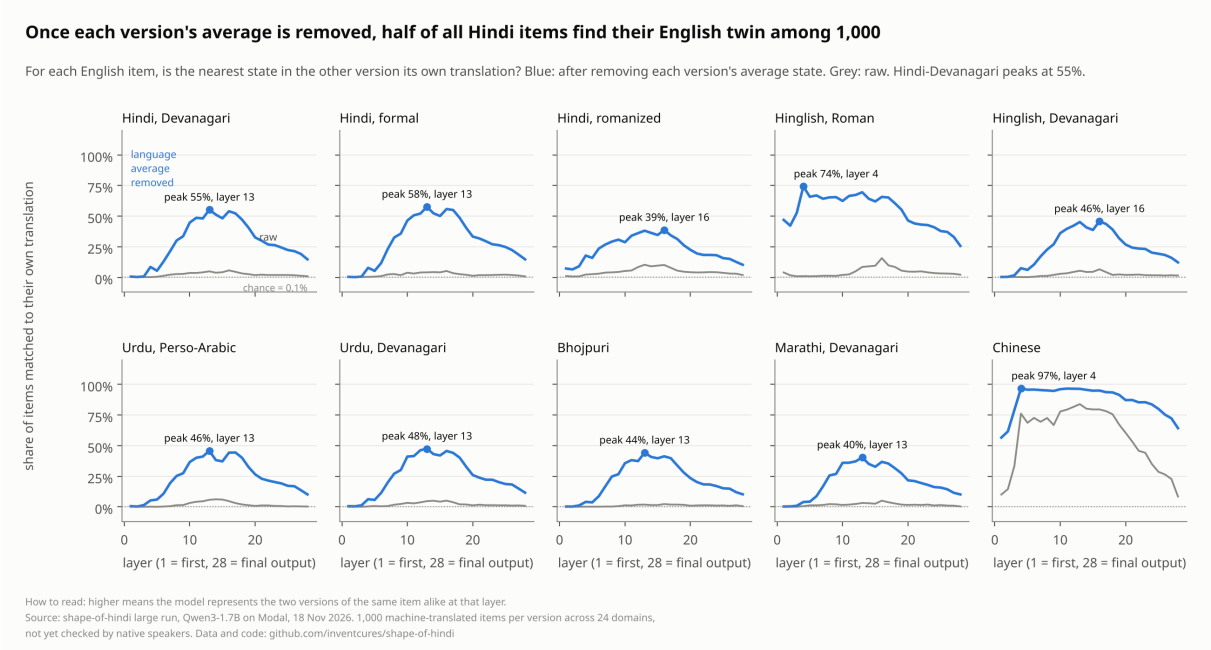


Figure 1: **Every version aligns with English in the middle layers and separates at the end.** Share of English items whose nearest state in the other version is their own translation, among 1,000 (chance 0.1%). Blue: after removing each version's mean. Grey: raw.

Raw retrieval from English stays below 16% at every layer for every version except Chinese (Figure 1). After centring, everyday Hindi peaks at 55% at layer 13 and falls to 15% at layer 28. Formal Hindi peaks at 58%, Urdu at 46% and 48% in its two scripts, Bhojpuri at 44%, Marathi at 40% and romanized Hindi at 39%, all at layers 13 to 16. Roman Hinglish peaks at 74% by layer 4, consistent with its English content words, and Chinese at 97%, which I attribute partly to close translations. Most of what separates the versions is therefore a per-version offset, as found for encoders [6, 7].

4.2 Script and language have effects of the same size

Two comparisons each change one factor (Table 3, Figure 2).

Devanagari Urdu exceeds Marathi at all 28 layers, by 0.05 at layer 14 and 0.09 at layer 28. Devanagari Urdu exceeds Perso-Arabic Urdu by 0.07 and 0.10. The two effects are comparable, which is H3. As a consequence, Perso-Arabic Urdu, which pays the script cost, and Marathi,

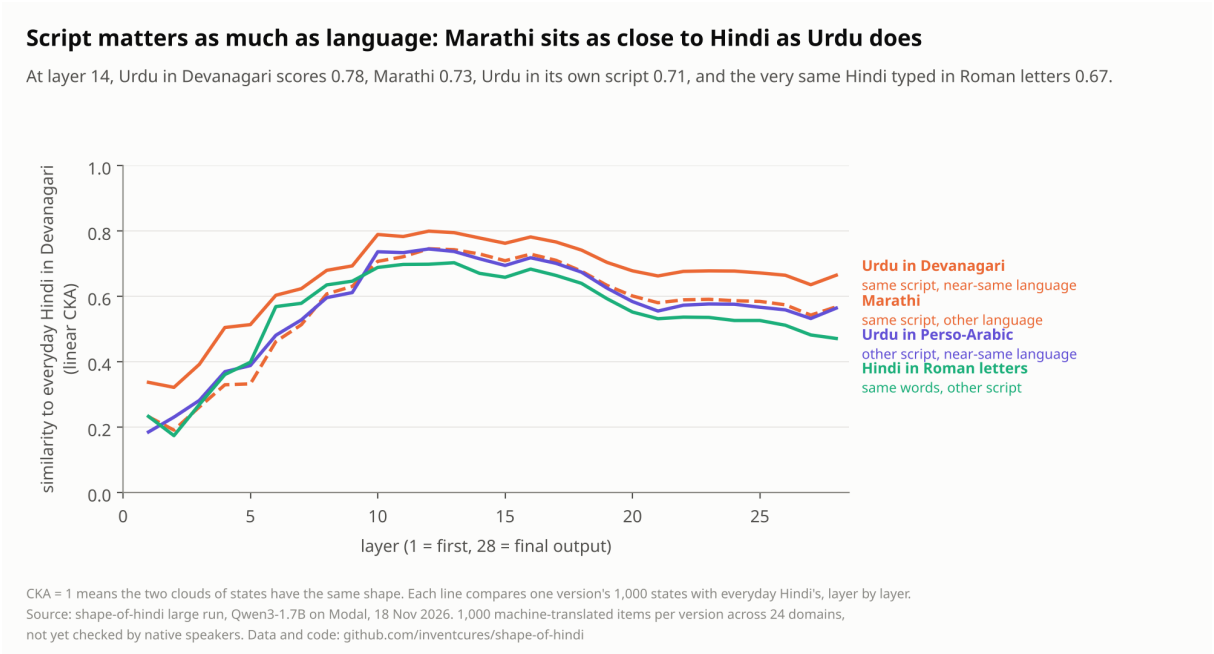


Figure 2: **Devanagari Urdu stays closest to Hindi; Marathi and Perso-Arabic Urdu tie; romanized Hindi drifts furthest after layer 10.** Linear CKA between each version's 1,000 states and everyday Hindi's.

Table 3: Controlled comparisons against everyday Hindi, linear CKA.

Comparison	Fixed	Varied	Layer 14	Layer 28
Devanagari Urdu − Marathi	script	language	0.78 − 0.73	0.66 − 0.57
Devanagari Urdu − Perso-Arabic Urdu	language	script	0.78 − 0.71	0.66 − 0.56

which pays the language cost, are close: Urdu leads at 8 of 28 layers, all before layer 12, and Marathi leads by at most 0.025 thereafter. The pilot's late lead for Urdu (0.86 against 0.80 with 20 items) did not replicate. Two caveats limit the comparison. I have not tested whether 0.09 and 0.10 differ, which would need bootstrap intervals over items or a rank test as in [9]; and I have not subtracted a shuffled-pair floor [10], which requires the full state files. The weaker Urdu translations (Table 2) likely bias both Urdu scores downward, which would understate the language effect in the second row and not the first.

4.3 Romanized Hindi is the outlier

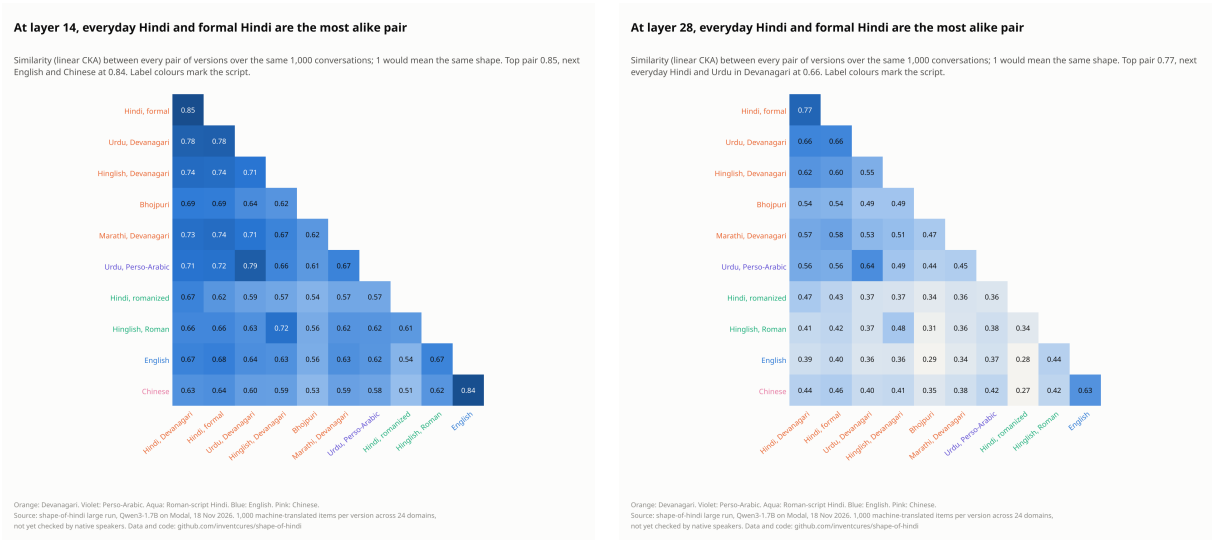


Figure 3: **All pairs are alike in the middle; by the last layer romanized Hindi stands apart.** Linear CKA between all pairs at layers 14 (left) and 28 (right).

At layer 14 every pair scores at least 0.51; everyday and formal Hindi reach 0.85 and English and Chinese 0.84 (Figure 3). At layer 28 the closest pair is still everyday and formal Hindi (0.77), so register moves states less than any other change in the set. Romanized Hindi scores 0.47 with the Devanagari Hindi it was transliterated from and less with every other version, down to 0.27 with Chinese. It is further from English than Devanagari Hindi at 24 of 28 layers, against my prediction that it would drift toward English. Work that improves romanized-Hindi handling does so by continued pretraining on romanized text (RomanSetu, [15]); Qwen3 is used here off the shelf. A plausible cause is inconsistent romanized spelling, which scatters each word over many token sequences, but I have not tested it.

4.4 Maps

Figure 4 shows the same pattern as the metrics. Urdu is a separate cluster at layers 7 and 14 and merges with the Devanagari versions by layer 21. A token-level map at the last layer (in the repository, with rotating 3D versions) shows all six Devanagari versions covering the same region whatever their language, which is expected: with tied embeddings the final state is pulled toward the next token's embedding.

4.5 Predictions

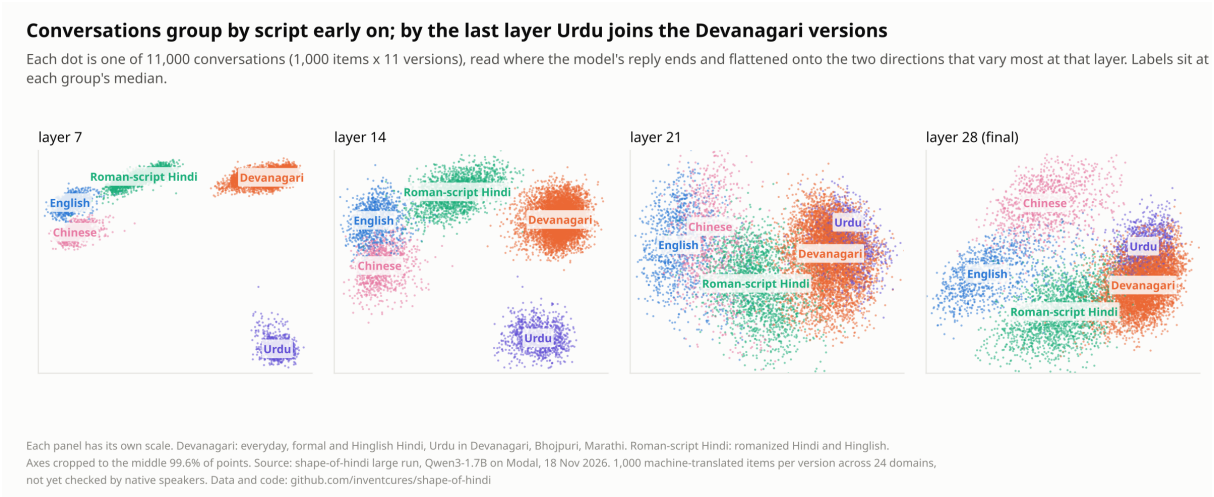


Figure 4: **Conversations group by script early; by the last layer Urdu joins the Devanagari versions.** End-of-reply states for 11,000 conversations, each panel in a PCA frame fit at that layer.

Table 4: Pre-registered predictions against the 1,000-item run.

	Outcome
P1	Holds: at layer 28 centred retrieval is 15% for Hindi against 64% for Chinese
P2	Holds: centred peak 55% at layer 13, 15% at layer 28; same shape in all versions
P3	Fails: romanized Hindi is further from English than Devanagari Hindi at 24 of 28 layers
P4	Fails as stated: Perso-Arabic Urdu and Marathi tie after layer 11; with script fixed, Urdu is closer at all layers
P5	Holds: Devanagari Hindi 4.95 times English, romanized 2.13
P6	Not testable: length and script are collinear in this model

5 Five models

Qwen3-1.7B is small, so I ran the same 11,000 conversations through four larger open-weight models: Qwen3.8-27B (64 layers), Gemma 4 31B in its QAT w4a16 release (60), gpt-oss-120b (36) and Sarvam-105B in bf16 (32), which Sarvam built for Indian languages. Extraction used the same end-of-reply read-out on H100 GPUs on Modal, for about $32 in total. Depth is shown on a common scale from 0 (first layer) to 1 (last). DeepSeek V4.1 Flash failed to load because the transformers library does not yet support its architecture; GLM-5.3 was out of budget. Table 5 summarises the results.

	Qwen3-1.7B	Qwen3.8-27B	Gemma 4 31B	gpt-oss-120b	Sarvam-105B
Layers	28	64	60	36	32
Hindi tokens per English token	4.95	2.14	1.32	1.60	1.32
Peak Hindi retrieval	0.55	0.91	0.99	0.85	0.66
at depth	0.46	0.23	0.38	0.19	0.47
Last-layer Hindi retrieval	0.15	0.44	0.62	0.14	0.01*
Language gain, last layer	+0.10	−0.00	−0.09	−0.02	−0.00*
Script gain, last layer	+0.10	−0.05	+0.02	−0.04	+0.01*
Romanized Hindi CKA, last layer	0.47	0.63	0.71	0.33	0.02*
Hindi/English participation ratio, last	2.44	0.63	0.81	0.97	1.16
First-token max ratio	1174	5.4	4.5	35	70
Length R^2, top 10 PCs, middle	0.98	0.86	0.54	0.78	0.64

Table 5: Summary across models. *Sarvam's final normalisation collapses the participation ratio from about 45 to 1.3; its last-layer values are not interpreted.

Alignment. All five models show the rise, mid-depth peak and late fall of centred retrieval (Figure 5), but the peak ranges from 0.55 to 0.99. Gemma 4 retrieves every version at 0.90 or better at its best layer; romanized Hindi is the hardest version for the other four.

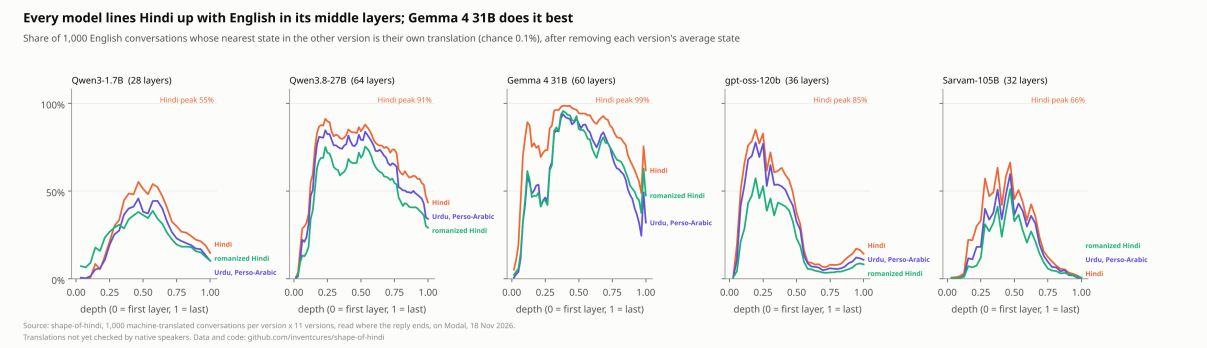


Figure 5: Centred retrieval from English to Hindi, Urdu and romanized Hindi across depth, one panel per model.

Token cost. Peak retrieval correlates with Hindi's token premium ($r = -0.68$, Figure 6), but the correlation rests mostly on Qwen3-1.7B, which is also by far the smallest model, and Sarvam and Gemma share a premium of 1.32 while differing by 33 points. Size, data and architecture are confounded with the tokenizer across five models. A lower premium does weaken the length confound: the share of token-count variance explained by the top 10 components at mid-depth falls from 0.98 in Qwen3-1.7B to 0.54 to 0.86 in the others.

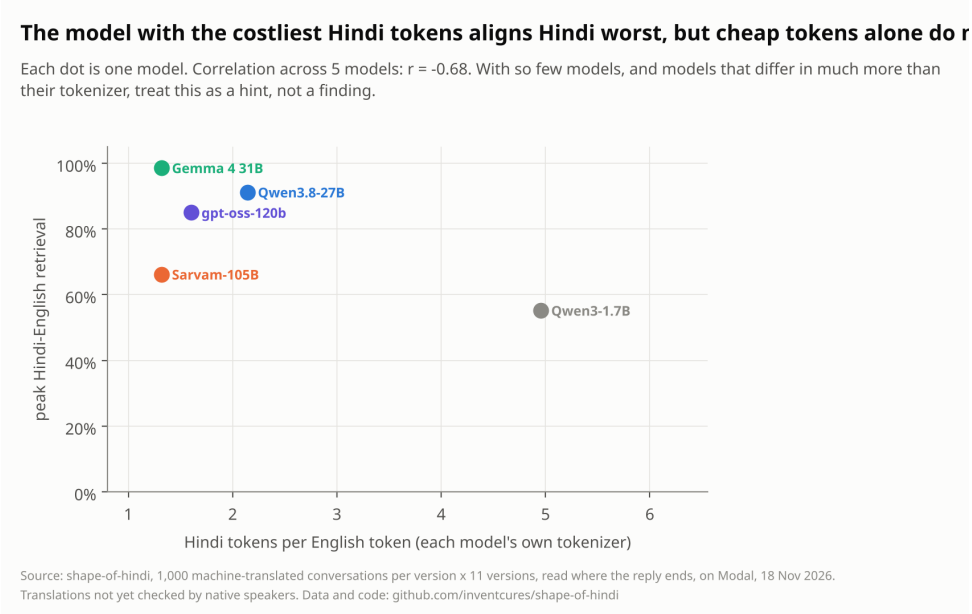


Figure 6: Hindi tokens per English token against peak Hindi–English retrieval.

Script against language. The equal script and language effects of Section 4 appear only in Qwen3-1.7B (Figure 7). In the four larger models all eight last-layer gains lie within 0.09 of zero and six are negative. I do not interpret their signs, given the Urdu translation errors and the absence of intervals, but the contrast with the small model is large. The PCA maps agree (Figure 8): Qwen3-1.7B separates Devanagari, Roman and Perso-Arabic Hindi at mid-depth, while Qwen3.8 and Gemma place them in one cloud, apart from English and Chinese. Gemma's last layer separates them again by script, consistent with late layers encoding the next token.

Ranking and romanized Hindi. Formal Hindi is the version closest to everyday Hindi at mid-depth in every model, tied with romanized Hindi in Qwen3.8 and Gemma (Figure 9). The rest of the order varies (lowest rank correlation with the mean order 0.54), and the level varies more: Gemma's mid-depth CKA spans 0.67–0.85, Sarvam's 0.28–0.45. Romanized Hindi is the furthest version from Devanagari Hindi at the last layer in Qwen3-1.7B and gpt-oss, but second-closest in Qwen3.8 (0.63) and Gemma (0.71).

Single-model effects. Three Qwen3-1.7B properties do not recur: the first-token massive activation (1,174 times a typical token's maximum, against under 6 in Qwen3.8 and Gemma) [23]; a Hindi participation ratio 2.4 times English's at the last layer (0.6–1.5 elsewhere); and the near-perfect length predictability. Sarvam's final normalisation maps every conversation onto nearly one direction, a rogue-dimension pattern known to distort similarity measures [26]; I have not checked whether it arises in the model or in its custom loading code.

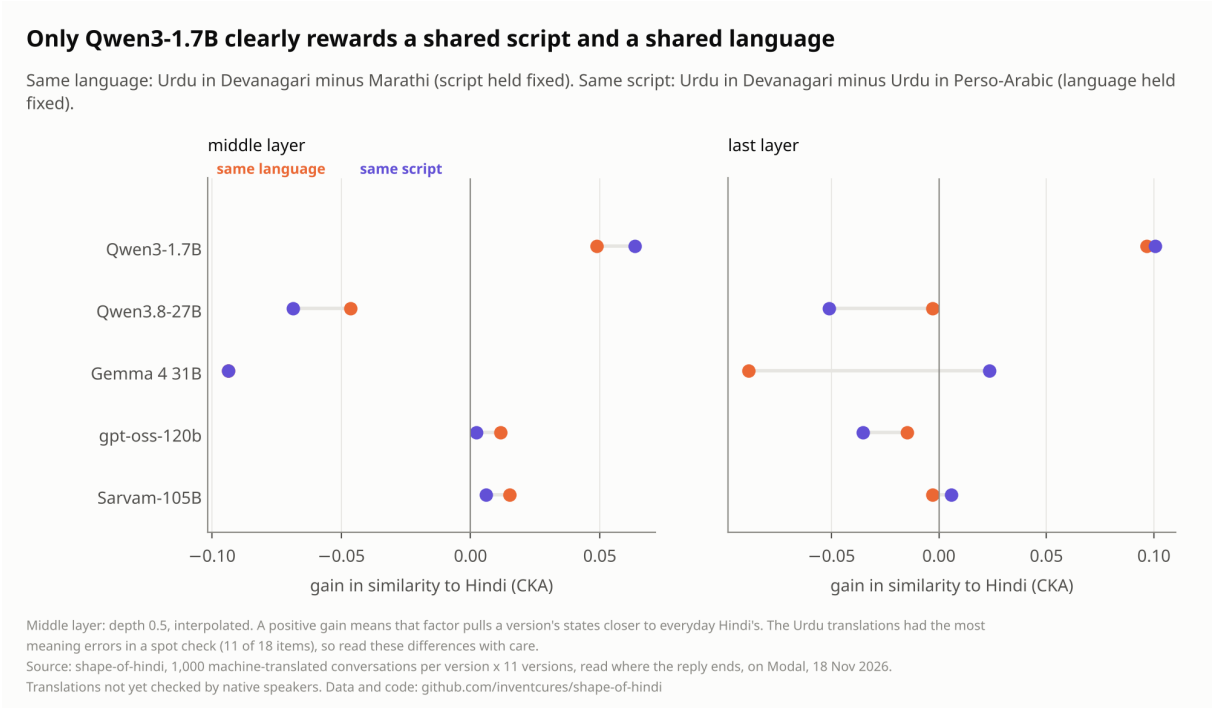


Figure 7: Gain in CKA with everyday Hindi from sharing its language and from sharing its script, middle and last layer.

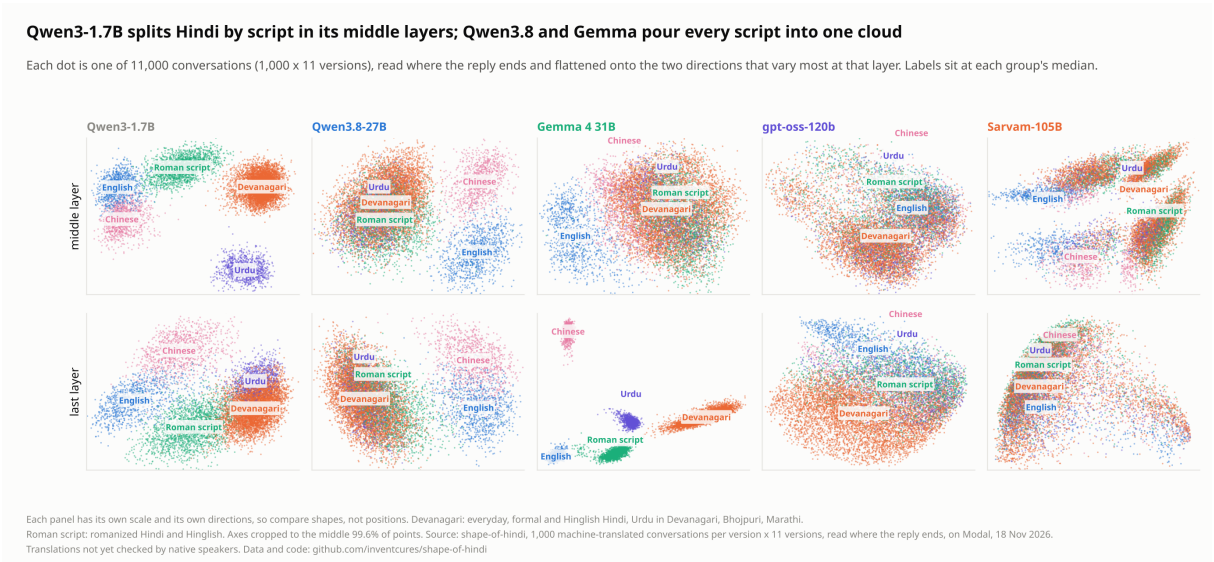


Figure 8: End-of-reply states projected on the top two principal components, middle and last layer, per model.

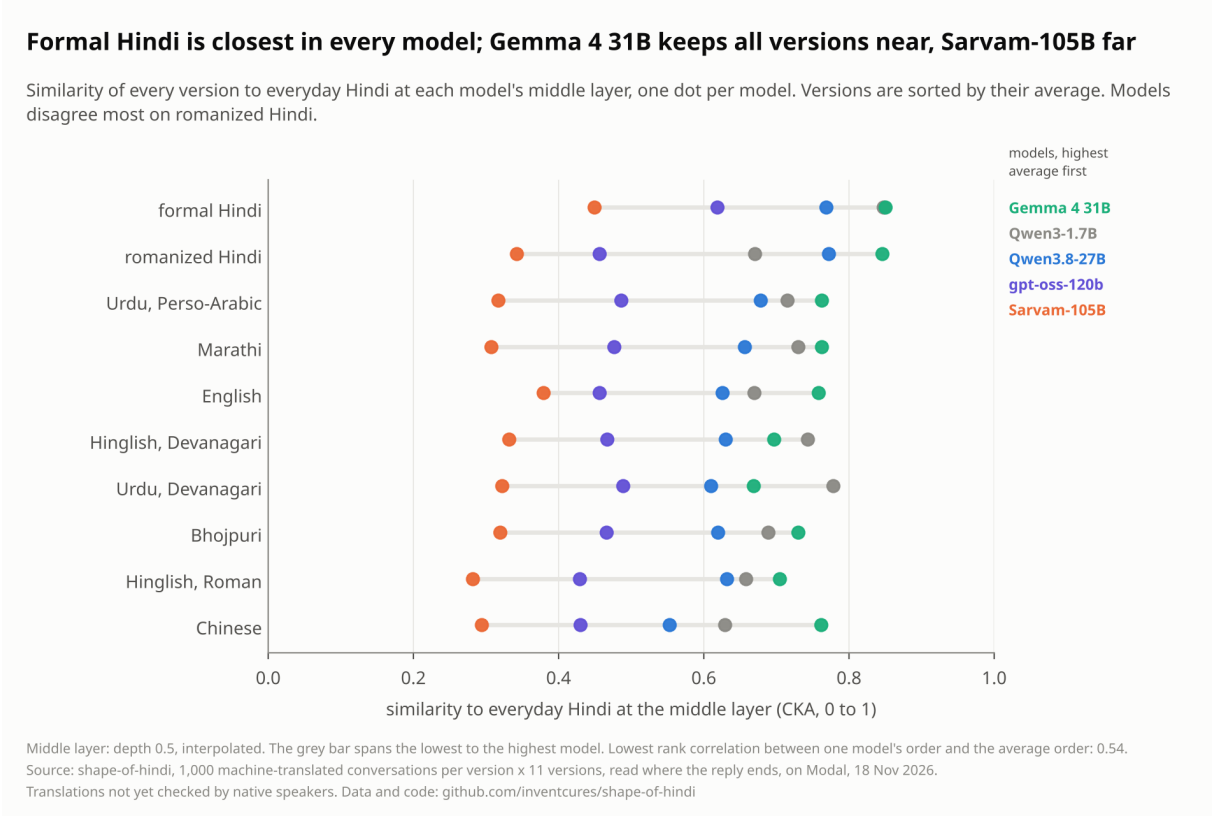


Figure 9: Mid-depth CKA of every version with everyday Hindi, one dot per model.

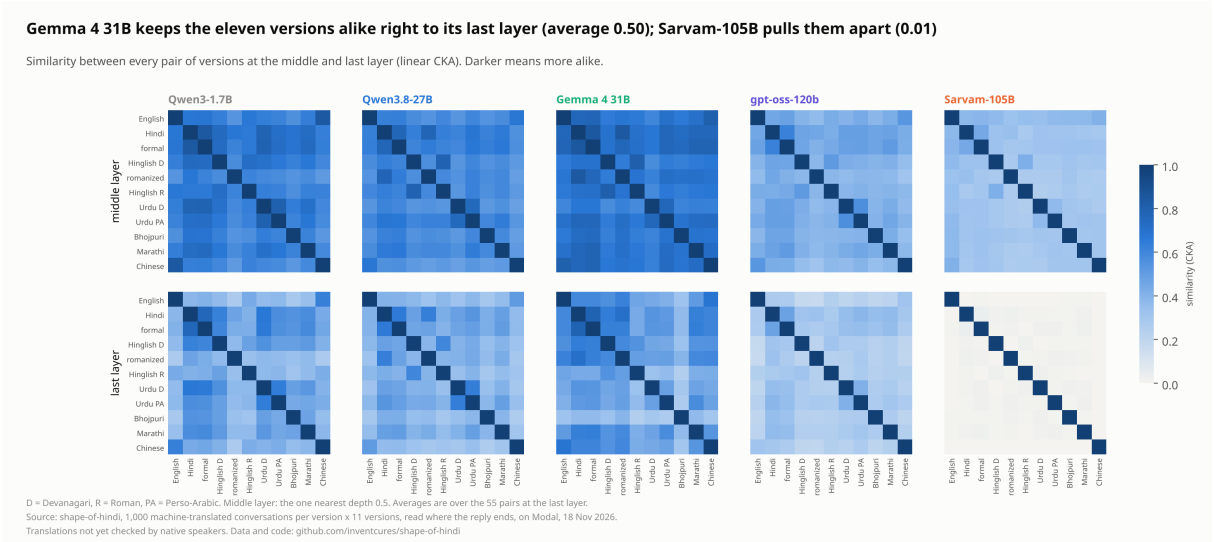


Figure 10: CKA between every pair of versions at the middle and last layer, per model.

6 Token cost

For parallel texts s_A and s_B the token premium is $\pi_t = |t(s_A)|/|t(s_B)|$ [16], computed here as a ratio of totals over the 1,000 conversations. It factors exactly into a byte premium and a tokens-per-byte premium:

$$\pi_t = \underbrace{\frac{|b(s_A)|}{|b(s_B)|}}_{\pi_b,\ \text{script}} \times \underbrace{\frac{|t(s_A)|/|b(s_A)|}{|t(s_B)|/|b(s_B)|}}_{\pi_{t/b},\ \text{vocabulary}},$$

where $|b(\cdot)|$ counts UTF-8 bytes. Here "cost" means token count under one tokenizer.

Table 6: Qwen3 token premium over English and its two factors, 1,000 conversations.

Version	Bytes per conv.	π_b	$\pi_{t/b}$	π_t
English	282	1.00	1.00	1.00
Chinese	277	0.98	1.00	0.99
Hinglish, Roman	349	1.24	1.35	1.67
Hindi, romanized	357	1.27	1.68	2.13
Urdu, Perso-Arabic	561	1.99	1.83	3.64
Bhojpuri	765	2.71	1.72	4.66
Hindi, everyday	808	2.87	1.73	4.95
Marathi	817	2.90	1.72	4.99
Urdu, Devanagari	828	2.94	1.73	5.09
Hindi, formal	844	2.99	1.71	5.14

Devanagari takes three bytes per letter in UTF-8, giving $\pi_b \approx 2.9$ for every Devanagari version (Table 6); my 2.87 for Hindi compares with 2.37 on Arnett et al.'s corpus [18]. Of Qwen3's 151,669 vocabulary entries only 71 contain a Devanagari letter, against 3,635 with Arabic script and 25,557 with a Chinese character, which gives the remaining $\pi_{t/b} \approx 1.7$. Hindi averages about one token per character, against 4.8 characters per token for English. Urdu, nearly the same language, has a lower byte premium (1.99; two-byte letters and unwritten short vowels) and ends at 3.64.

Table 7: Token premium over English by tokenizer, same 1,000 conversations. Tokenizers identical to one listed (p50k, gpt-oss, Sarvam-105B) are omitted. Qwen3's 4.94 here comes from a separate counting script and differs slightly from the 4.95 in Table 6.

Tokenizer	Hindi	Romanized Hindi	Urdu
GPT-3 (r50k)	7.89	2.26	6.84
GPT-3.5/4 (cl100k)	5.30	2.14	4.98
Mistral 7B	5.04	2.13	4.78
Qwen3	4.94	2.13	3.65
GLM-5.3	4.77	2.11	3.11
DeepSeek-V4	3.09	2.12	2.66
Qwen3.8	2.13	2.03	2.19
Mistral Large 4	1.66	1.97	1.64
GPT-4o and later (o200k)	1.61	1.97	1.74
BLOOM	1.34	1.98	1.42
Gemma 4	1.32	1.74	1.53
Sarvam-1	1.22	2.36	8.33

The vocabulary factor is a design choice (Figure 11, Table 7). Under the newest tokenizers romanized Hindi costs more than Devanagari, reversing the older pattern, and Sarvam-1, built for Indian languages, charges Urdu 8.33.

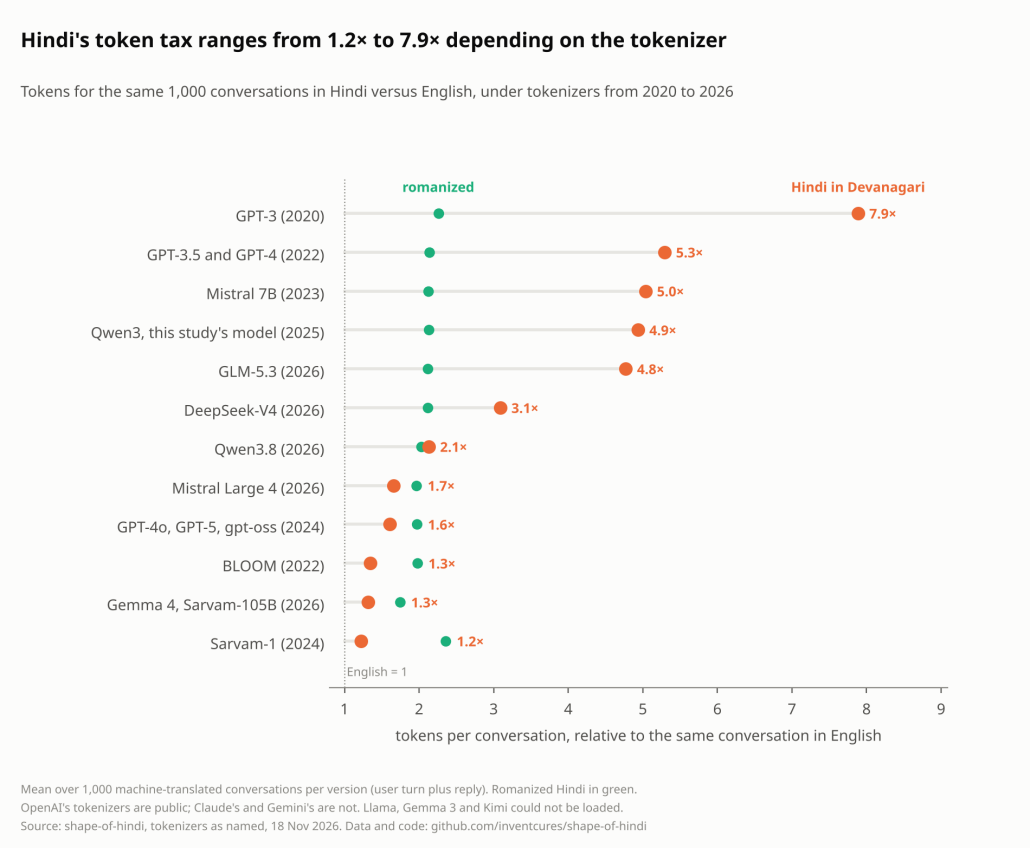


Figure 11: **Hindi's premium ranges from 7.9 to 1.2 depending on the tokenizer.** Same 1,000 conversations under 12 tokenizers. Orange: Devanagari Hindi. Green: romanized Hindi.

Consequences. A premium $\pi_t = 5$ scales per-token price, generation time and context use linearly. At $0.50 per million output tokens, a million 70-word replies a day cost about $40 in English and $200 in Hindi; at 50 tokens per second a reply takes 1.6 against 8 seconds; a 32k context holds about 540 English conversations from this set against 110 Hindi ones. A forward pass costs about $2N$ FLOPs per token, so for $N = 1.7 \times 10^9$ an English conversation of 59 tokens costs about 0.2 TFLOP and a Hindi one of 292 tokens about 1 TFLOP; attention over long inputs grows as $\pi_t^2 = 25$. Deng et al. measured Qwen3-8B, which shares the tokenizer, and found Hindi used 0.20 J per output token against 0.10 for English and 8.5 times the total energy [19].

7 Discussion

The results fit the three-phase picture with one refinement. In the middle layers, all eleven versions share much of their structure (every pair at least 0.51, centred retrieval peaking at layers 13 to 16). At the last layer the state is pulled toward the next token's embedding, so the alphabet about to be written dominates. Neither factor wins the script-against-language contest: in this model, sharing Hindi's script and sharing its language are worth about the same. Section 5 qualifies this: in four larger models neither effect is distinguishable from zero, so H3 describes Qwen3-1.7B rather than decoder LLMs in general. What does hold across all five is the mid-depth alignment peak and formal Hindi's position nearest everyday Hindi. The romanized result suggests a third factor, how consistently a version is spelled, that neither hypothesis names.

8 Limitations

Data provenance. Every non-English text was written by one open model and checked by no native speaker. A 12-item reading found meaning errors in half the Urdu items, and the Bhojpuri mixes Hindi, Bhojpuri and Maithili forms.

Five models, one dataset. The detailed results come from Qwen3-1.7B. The four larger models differ from it in size, tokenizer, data and architecture at once, so Section 5 can say what changes but not why. Sarvam's last layer is excluded from interpretation.

Length and script. The top 10 components predict log token count with R^2 between 0.94 and 0.98 at every layer, but the Devanagari versions are both about five times longer and grouped together, so length and script cannot be separated here. In models with lower token premiums the R^2 falls to 0.54–0.86 (Section 5).

No uncertainty or floor. The CKA differences lack confidence intervals and a shuffled-pair baseline.

Teacher forcing. The model reads fixed texts; this measures representation, not generation.

Method. Dimensions were not standardised before PCA, although the original plan specified it.

9 Next experiments

A native-speaker check of a fixed sample; DeepSeek V4.1 Flash and GLM-5.3 once they can be loaded, and a check of Sarvam's final layer; bootstrap intervals and shuffled-pair floors for every CKA and retrieval curve; token-by-token trajectories through Hinglish switch points; and a warmth direction, following Anthropic's finding that Claude leans furthest toward warmth in Hindi [24], tested on the pronouns *aap*, *tum* and *tu* with universal and India-specific topics kept apart [25].

Reproducibility. Data, code, results and figures: `https://github.com/inventcures/shape-of-hindi`. The study step runs in about 82 minutes on one H100. This edition was drafted by Claude from the same data as the first, restructured after human-written exemplars; the first edition is unchanged.

A Pilot

A 20-item pilot in 8 versions (17 November 2026) gave a centred Hindi–English retrieval peak of 90% at layer 16 with 5% chance, Devanagari Hindi at 4.1 times English tokens, Marathi closer to Hindi than Urdu in layers 1 to 4, and Urdu closer from layer 20. Only the first of the Urdu–Marathi findings survives at 1,000 items.

B Exercises

1. Using Table 3, which pair of numbers isolates the cost of script, and why does Perso-Arabic Urdu tie with Marathi?
2. Why can centred retrieval fall from 90% to 55% between 20 and 1,000 candidates without any change in the model?
3. Show that $\pi_t = \pi_b \cdot \pi_{t/b}$ exactly, and compute $\pi_{t/b}$ for Urdu from Table 6.
4. A tokenizer brings Hindi to $\pi_t = 1.6$ and the energy per token to 1.2 times English's. Estimate the energy ratio per reply.

Answers

1. Devanagari Urdu against Perso-Arabic Urdu (0.66 and 0.56 at layer 28) varies only the script. Perso-Arabic Urdu pays for its script and Marathi for its language, by similar amounts.
2. More candidates bring more near neighbours that are similar but wrong, so the top-1 match fails more often.
3. The byte counts cancel. For Urdu, $3.64/1.99 \approx 1.83$.
4. About $1.6 \times 1.2 \approx 1.9$.

References

[1] T. Tang et al. Language-specific neurons: the key to multilingual capabilities in large language models. ACL 2024. arXiv:2402.16438.

[2] D. Gurgurov et al. Language arithmetics: towards systematic language neuron identification and manipulation. IJCNLP-AACL 2025. arXiv:2507.22608.

[3] C. Wendler, V. Veselovsky, G. Monea, R. West. Do Llamas work in English? On the latent language of multilingual transformers. ACL 2024. arXiv:2402.10588.

[4] J. Lindsey, E. Ameisen, et al. On the biology of a large language model. Transformer Circuits, 2025.

[5] C. Dumas et al. Separating tongue from thought: activation patching reveals language-agnostic concept representations in transformers. ACL 2025. arXiv:2411.08745.

[6] T. Chang, Z. Tu, B. Bergen. The geometry of multilingual language model representations. EMNLP 2022. arXiv:2205.10964.

[7] J. Libovický, R. Rosa, A. Fraser. On the language neutrality of pre-trained multilingual representations. Findings of EMNLP 2020. arXiv:2004.05160.

[8] Bhat et al. A house united: bridging the script and lexical barrier between Hindi and Urdu. COLING 2016.

[9] Moosa, Akhter, Habib. Does transliteration help multilingual language modeling? Findings of EACL 2023. arXiv:2201.12501.

[10] Liu et al. How transliterations improve crosslingual alignment. arXiv:2409.17326, 2024.

[11] Verma et al. Multilingual language models encode script over linguistic structure. ACL 2026.

[12] Karne. One language, two scripts. arXiv:2603.08869, 2026.

[13] A. Saji et al. RomanLens: the role of latent romanization in multilinguality in LLMs. arXiv:2502.07424, 2025.

[14] D. Gurgurov, A. Saji, et al. The Latin substrate: how language models represent and mediate script choice. arXiv:2605.31363, 2026.

[15] RomanSetu: efficiently unlocking multilingual capabilities of large language models via romanization. arXiv:2401.14280, 2024.

[16] A. Petrov, E. La Malfa, P. Torr, A. Bibi. Language model tokenizers introduce unfairness between languages. NeurIPS 2023. arXiv:2305.15425.

[17] O. Ahia et al. Do all languages cost the same? Tokenization in the era of commercial language models. EMNLP 2023. arXiv:2305.13707.

[18] C. Arnett, T. A. Chang, B. K. Bergen. A bit of a problem: measurement disparities in dataset sizes across languages. arXiv:2403.00686, 2024.

[19] Deng, Shen, Feng, Nwatu, Chung, Chowdhury, Chen, Mihalcea. The language–energy divide: measuring energy costs of multilingual LLM inference. EMNLP 2026.

[20] S. Kornblith, M. Norouzi, H. Lee, G. Hinton. Similarity of neural network representations revisited. arXiv:1905.00414, 2019.

[21] M. Davari et al. Reliability of CKA as a similarity measure in deep learning. ICLR 2023. arXiv:2210.16156.

[22] M. Del, M. Fishel. Cross-lingual similarity of multilingual representations revisited. AACL 2022.

[23] M. Sun, X. Chen, J. Z. Kolter, Z. Liu. Massive activations in large language models. arXiv:2402.17762, 2024.

[24] M. Kearney, M. Zhang, et al. Claude's values across models and languages. Anthropic, 13 July 2026.

[25] Sattigeri. Extending Beacon to Hindi. arXiv:2602.00046, 2026.

[26] W. Timkey, M. van Schijndel. All bark and no bite: rogue dimensions in transformer language models obscure representational quality. EMNLP 2021. arXiv:2109.04404.