# Do LLMs Model Human Linguistic Variation? A Case Study in Hindi-English Verb Code-Mixing

**Mukund Choudhary**[*1] **Madhur Jindal**[*1] **Gaurja Aeron**[2] **Monojit Choudhury**[1]
[1]MBZUAI [2]IIT Gandhinagar
{first.last}@mbzuai.ac.ae @iitgn.ac.in

## Abstract

Do large language models (LLMs) model linguistic variation? We investigate this question through Hindi-English (Hinglish) verb code-mixing, where speakers can use either a Hindi verb or an English verb with the light verb *karna* ('do'). Both forms are grammatical, but speakers show unexplained variation in language choice for the verb. We compare human preferences on controlled code-mixed minimal pairs to LLM perplexities spanning families, sizes, and training language compositions. We find that current LLMs do not reliably classify verb language preferences to match native speaker judgments. We also see that with specific supervision, some models do predict human preference to an extent. We release (here) native speaker acceptability judgments on 30 verb pairs, perplexity ratios for 4,279 verb pairs across 7 models, and experimental materials.

## 1 Introduction

> *The point isn't whether a text produced by any system contains words from more than one language - it is rather whether they're mixed in the way that a bilingual human might.* (Sterner and Teufel, 2025)

Social Science and policy researchers are increasingly using LLMs as human behavioral proxies: to simulate American survey responses on new surveys (Hewitt et al., 2024), qualitative interviewing of replicas fine-tuned on thousands of real demographic backstories (Argyle et al., 2023), and more. However, parallel literature also finds that LLMs are not grounded in human motives or embodiment, so in scenarios requiring strategic thinking, genuine uncertainty (Gao et al., 2025), or deeply subjective, story-driven research (Kapania et al., 2025), LLMs cannot be a reliable proxy for human psychology.

Meanwhile, linguistics research shows us that LLMs are already good at making human-like decisions on grammaticality judgments across languages (Jumelet et al., 2025) but struggle at matching human linguistic intuition in novel contexts like suffixing nonce adjectives (Weissweiler et al., 2025). These debates prompt us to think that while modeling human social behavior is a complex task because of its high subjectivity, to what extent do LLMs model human *linguistic variation*?

We test this question on the phenomenon of Hindi–English (Hinglish) verb code-mixing: As shown in Fig.1, for a given Hindi predicate, native speakers can use either a Hindi verb or an English verbal noun followed by the Hindi verbalizer *karna* ('to do'). We find this phenomenon a neat test of LLM-human linguistic variation alignment as: (a) Replacing the Hindi verb in a sentence with its Hinglish counterpart does not make it ungrammatical, (b) however native speakers find some Hinglish verbs awkward to mix, some comfortable, and some where they're preferred over the Hindi verb. (c) the phenomenon hasn't been explained theoretically.

| | | | | | |
|---|---|---|---|---|---|
| Hindi verb + frame | wo | naach | | rahe | hain |
| | 3PL | dance | | PROG | be.PRS.PL |
| English verb + Hindi frame | wo | dance | kar | rahe | hain |
| | 3PL | dance | do | PROG | be.PRS.PL |

Figure 1: Hindi vs. Hinglish code-mixed at verb, for: *They*(3PL) + *are*(be.PRS.PL) + *danc-ing*(dance-PROG).

We design an experiment to collect human preferences over 30 verbs of varying degrees of mixability in context of sentence pairs (Fig.1) and compare them to LLM perplexity ratios on the same. Our findings show that LLMs –across, sizes, families, and degrees of Hindi-English bilingual training data– do not model human linguistic variation shown by this phenomenon. Careful feature selection and supervision showed the highest (but insufficient) alignment only by using Sarvam-1 (AI, 2024)

[*]Equal contribution



*Findings of the Association for Computational Linguistics: EACL 2026*, pages 5491–5509

(F1=0.81) and doesn't generalize across models.

Apart from the broader findings on LLM usability for linguistic variation studies, this work also provides the first systematic evaluation of human preferences for Hinglish verb code-mixing. We release native bilingual preference ratings focused on this phenomenon for 30 verb pairs, perplexity ratios for 4,279 verb pairs across 7 base LLMs.

## 2 Background

### 2.1 Code-mixing and the *Do* verb in Hinglish

**Code-mixing** is used to describe the process of native speakers embedding words or phrases from one language (*embedded*) into the grammatical structure of another (*matrix*) (Myers-Scotton, 1997).

In Hinglish code-mixing, Hindi is typically the *matrix*, and mixing is mostly *insertional*, i.e. English items put into Hindi frames. So, English words combine in ways (e.g., taking Hindi postpositions) to maintain grammaticality. These account for the grammatical *shape*, but also leave room for *choice* in which words to mix and how often, varying across items and speakers (Bali et al., 2014).

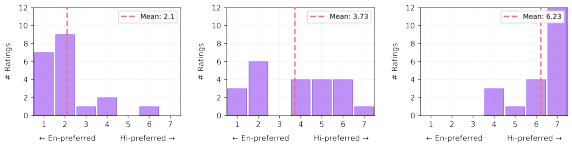
Predicates are majorly expressed in two ways: either the native Hindi verb, or an English noun/adjective followed by the Hindi light verb *karna* (do) (Dey and Fung, 2014). Both versions are grammatical, as the pattern of mixing aligns with how Hindi compound verbs work independently: the first element contributes core lexical meaning, while the light verb (e.g., *karna*) supplies functional morphology (e.g. tense) (Kumar, 1986). However, as shown by Fig.1 (common) and Fig.2 (unacceptable), *not all Hinglish mixes are equally natural to natives*.

| | | | |
|---|---|---|---|
| Hindi verb + frame | *khaana* | *aasaan* | *hai* |
| | eat.INF | easy | be.PRS.SG |
| English verb + Hindi frame | *eat karna* | *aasaan* | *hai* |
| | eat do.INF | easy | be.PRS.SG |

Figure 2: Unnatural but grammatical, for: (*It*(EXPL) + *is*(be.PRS.PL)) + *easy*(easy) + *to*(INF) + *eat*(eat).

### 2.2 Acceptability Variation and Why its Difficult to Model

The naturalness acceptability variation in code-mixing thus lies on a *continuum*. Bali et al. (2014)'s frequency-based corpus analysis shows: *extreme conditions* - highly frequent English items that are readily mixed vs. near-absent ones that speakers avoid, and a *large middle* where both Hindi and English variants are used with comparable rates.




(a) kiss - चूम (b) change - बदल (c) tear - फाड़

Figure 3: Human preference histograms. x-axis: En to Hi-preferred (1-7), y-axis: # ratings (0-12) on all fairly frequent verbs. *kiss* is preferred at Fig.3a, फाड़ is preferred at Fig.3c, while Fig.3b shows easy mixability

However, as Fig.3 shows, in the case of even fairly frequent verbs in Hinglish, we observe all 3 parts of the continuum. There is no theory that fully explains Hinglish verb-mixing variation yet. This could happen due to various reasons in addition to lexical frequency: *Domain dominance* (English forms are preferred in tech, work, pop culture), *No native substitute / lexical gap* (proper nouns, terms of art), *Degree of bilingualism* (Dey and Fung, 2014), *Attitudes toward mixing* (positive = higher acceptability) (Badiola et al., 2018).

Recent empirical work in Hinglish[1] by Kodali et al. (2025) shows that structure-only metrics of "mixing" in a sentence: *Code-Mixing Index* (Gambäck and Das, 2016), simple *number-of-switch* counts (Pratapa et al., 2018), correlate poorly with human acceptability. In contrast, **fine-tuned** multilingual language models (LMs) trained on human judgments substantially outperform such metrics at distinguishing natural vs. awkward code-mixed sentences. This shows that fine-tuned LMs can capture general Hinglish acceptability to some extent.

**Our focus:** We thus specifically study the verb code-mixing phenomenon in Hinglish, as it presents an isolated, interpretable test for base LLM alignment to human linguistic variation.

## 3 Data

Preference data was prepared by constructing Hindi **frames** (Sec.3.2), scraping Hindi-English **verb pairs** to insert in the frames (Sec.3.1), and present them in different orthographies/**forms** (Sec.3.3).

### 3.1 Verb Pairs

The verb pairs were obtained from Indowordnet (Bhattacharyya, 2010) parallel lexicon. It was preprocessed (App.B) to retain verbs, remove phrases, etc., yielding a total of 4,279 unique verb pairs.

[1]Prevalent online (Sengupta et al., 2024) and is a part of LLMs' training as seen empirically (Yang and Chai, 2025).

| Frames | Gloss | Meaning |
|---|---|---|
| wo X (kar) rahe hain | 3PL/SG X (do) PROG be.PRS.PL | 'They are X-ing' |
| X(kar)na aasan hai | X (do.)INF easy be.PRS.SG | 'Doing X is easy' |
| wo kal X (kar) chuke the | 3PL/SG yesterday X (do) finish.PERF be.PST.PL | 'They had finished doing X yesterday' |
| X( kar)na mana hona chahiye | X (do.)INF forbidden be.INF should | 'Doing X should be forbidden' |
| wo aaram se X( kar)te rahenge | 3PL/SG ease with X (do.)HAB continue.FUT.PL | 'They will keep doing X' |

Table 1: Hindi (and English) frame templates with glosses and translations. X marks the spot of the verb.

### 3.2 Frames

We make 5 Hindi frames (Tab.1) to control for:

1. **Length**: Frames are max. 6 words (∼12-15 tokens) as LLM perplexities get less reliable and interpretable with length (Hu et al., 2024).
2. **Grammaticality**: Frames are grammatical with Hindi verbs *and* the Hinglish parallels.
3. **Minimal confounds**: Verb is the only content word, with max. one subject (third person, plural (3PL) pronoun, in the nominative case).[2]
4. **Verbal confounds**: Frames contain variants of verb positions, adverbs (positive/negative), and grammatical features (e.g. tense, aspect).[3]

### 3.3 Forms

We consider three orthographies for digital Hindi:

(a) **Devanagari**: Native script (e.g., वो नाच रहे हैं)
(b) **Casual Latin**: Native speaker romanization in free variation (e.g., *wo naach rahe hain*)
(c) **Formal Latin**: Standardized rule-based transliteration (Gupta et al., 2010) (e.g., *vo naacha rahe hain*). While not commonly in use, they are still often readable with very little training (e.g., ITRANS, WX notation)

We transliterated Devanagari using GPT-4.1 with iteratively refined prompts by native speaker evaluation. Complete prompts, development processes, and statistical analyses are provided in App.A.[4]

## 4 Experimental Design

### 4.1 LLM Preferences

#### 4.1.1 Model Choices

We choose 7 open-weights LLMs: Gemma3-1B, -4B, -12B (Team et al., 2025), Llama3.1-8B, -70B (Grattafiori et al., 2024), Qwen3-32B (Yang et al., 2025), and Sarvam-1 (2B)[5]. These choices represent popular architectures, sizes, model families, and training language mixes. Different model sizes within the same family help us examine scaling effects. Note that we also used each model's base version, as the aim is to assess existing models' latent alignment with human preferences, not to align LLMs to them. Finally, aside from Sarvam-1, none of these models are trained from scratch on substantial mixtures of Indic languages.[6]

#### 4.1.2 Measuring LLM Preferences

We quantify an LLM's preference for a sentence using perplexity. For a sentence $x_{1:N}$ of length $N$, the perplexity under model $\theta$ is defined as

$$\text{ppl}_\theta(x_{1:N}) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log p_\theta(x_i \mid x_{<i})\right) \quad (1)$$

which corresponds to the model's average per-token surprise (Eqn.1). To compare a Hindi sentence with its English counterpart, we use the perplexity ratio as our metric. For a given Hindi sentence $x_{\text{hi}}$ (in any of the three forms) and its English version $x_{\text{en}}$, we define the following ratio (Eqn.2)

$$PR_\theta(x_{\text{hi}}, x_{\text{en}}) = \frac{\text{ppl}_\theta(x_{\text{hi}})}{\text{ppl}_\theta(x_{\text{en}})} \quad (2)$$

**A note on why surprisal/perplexity is a reasonable unsupervised proxy for model preference:** Our goal is to compare models' *relative* preference between two minimally different but grammatical sentences. Computational linguistics (like psycholinguistics (Hale, 2001; Levy, 2008)) evaluates models' preference signal between minimal-pairs by using higher probability/lower surprisal

[2] In Hindi, the 3PL form (*wo*) does not need differential verb inflection and works with all nouns.

[3] Modality variations make frame lengths more disparate as they are marked by auxiliary particles in Hindi verb phrases.

[4] The five frames for both the Hindi and English verb counterparts in all three transliterations are presented in Tab.11.

[5] Run using Hugging Face Transformers (Wolf et al., 2020) with BF16 quantization, on 2x NVIDIA RTX 6000 GPUs running for 10 GPU hours. Refer App.C for more details.

[6] We also considered closed-source models, but their APIs did not support extracting logprobs for sequences, full probability distributions or token-level cross-entropies, rendering them unsuitable for our perplexity-based analysis.

as a proxy, e.g., BLiMP-style paradigms (Warstadt et al., 2020; Marvin and Linzen, 2018). We use $PR_\theta$ similarly, but target *preference under grammatical equivalence* rather than judging grammaticality.

However, absolute perplexity is not comparable across languages, models, and is sensitive to tokenization/orthographic choices. In contrast, $PR_\theta(x_{\text{hi}}, x_{\text{en}})$ is computed and compared *within model, frame, and orthography* on matched sentences, thus reducing confounds. Under this design, a lower $PR_\theta$ corresponds to the model assigning lower average surprise to the Hindi variant thus preferring it over English verb in Hindi frame.

### 4.2 Human Preferences

#### 4.2.1 Participants

Participants were recruited via Prolific (2025), inclusion criteria required participants to be: 18-65 years old, native Hindi speakers, and fluent in English. We collected responses from 18 participants, removed an outlier on the basis of z-score analysis on responses, and 3 others due to incomplete preference data, yielding **14** for the final analysis.[7]

#### 4.2.2 Data Curation

We sampled 30 verb pairs and selected 1 form, as collecting human judgments for all 4,279 verb pairs across 15 conditions (5 frames x 3 forms) would be expensive (Sterner and Teufel, 2025):

**Verb Pairs:** We sample 30 verb pairs for the Human experiment using a hybrid strategy. First, we use k-means clustering using 18 features for each pair: Llama3.1-70B[8] $PR_\theta$ for all 15 combinations (5 frames x 3 forms) + min, max, and std. deviation. This results in 4 clusters from which we randomly sample 100 pairs (25 per cluster) ensuring diversity and coverage in terms of LLM preference.

From these 100, two native speaker judges filtered 30 verb pairs (Tab.13) to cover a range, e.g. some pairs where both verb choices are common, some "extreme" pairs where one verb is more infrequent compared to its counterpart, etc.[9]

**Form:** We conducted a preliminary survey to determine native speaker preferred form. 32 participants of varying demographics were shown sentences in all 3 forms, formed with a curated set of 5 visually different Hindi verbs [10].

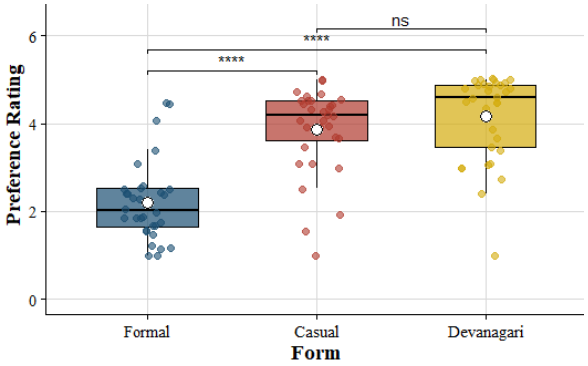



Figure 4: Preference distribution across 3 orthographies.

We decide to use Casual form as it is more comparable to how English verbs are presented and are more natural to native speakers. Note that Devanagari was almost equally preferred ($p > 0.05$, Fig.4), while Formal Latin was much lower in preference (ANOVA with post-hoc t-tests, $p < 0.05$).[11]

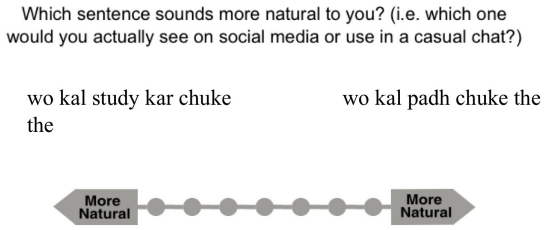



Figure 5: Experiment Screen

#### 4.2.3 Measuring Human Preferences

The experiment was implemented using Psytoolkit (Stoet, 2010, 2017)[12]. Each participant rated their preference between 2 sentences of the same frame and verb pair over 50 sentences (after 5 practice trials) on a 7-point scale, Fig.5. Verb pairs (and language sides) across frames were counterbalanced across participants over 150 sentence pairs. After outlier removal, 699 ratings across 100 sentences, provided by 14 participants, were analysed.

## 5 Analyses

We present our findings in a series of analyses, beginning with understanding how the data looks like, i.e. human judgment patterns on the verb pairs, and

[7] Demographics summary can be found at App.D

[8] We use this model for generating the features as it is expected to have seen the most data (as compared to other models in the set) thus minimizing effects due to OOD issues.

[9] We adopt a hybrid strategy because our study aims to observe emergent patterns from human preferences rather than inadvertently impose them through sampling conditions, changing it would alter the number of verbs per group, but not the underlying ground-truth human data or observed preferences.

[10] Based on factors like length, consonant cluster presence.

[11] This is because most native speakers are unaware of standardization rules meant for computational consistency.

[12] Instructions and Consent can be found at Figs. 6-7.

then evaluating model alignment through three connected experiments. Each experiment ends with observations about model-human alignment.

### 5.1 Preference Classes & Analyses Overview

From the Human Preference experiments we observed moderate inter-annotator agreement (ordinal Krippendorff's $\alpha = 0.365$, on a 7-point scale), confirming variability in human judgments, within typical ranges in the field Kodali et al. (2025); Sterner and Teufel (2025). We used a Monte Carlo simulation test for uniformity to characterize this variation (100,000 iterations, App.E). Verb pairs with non-uniform distributions ($p < 0.05$) were then manually classified by distribution shapes (Fig.3)[13]:

1. UN (n=12): Uniform-like distributions (p > 0.05) = no clear preference, e.g. change-बदल
2. HIP (n=13): Strong right-skewed distributions = clear Hindi preference, e.g. tear-फाड़
3. ENP (n=3): Strong left-skewed distributions = clear English preference, e.g. kiss-चूम
4. MP (n=2): MultiPeak = divergent speaker groups/context dependence, e.g. imagine-सोच

We see that over half of the verb pairs elicited a strong one-language preference (HIP, ENP), particularly favoring Hindi (unsurprisingly, given Hindi native bilinguals and frames) while the rest showed fair mixability (UN + MP). This shows that Hinglish verb mixing acceptability lies on a spectrum and is speaker & context dependent in some cases.

This gives us well-defined distribution-shape based classes for a conservative evaluation of model-human alignment. Although one could, in principle, compare the full 7-bin human preference histograms to model outputs using distributional distances such as the Wasserstein metric, doing so would require an explicit calibration from the scalar $PR_\theta$ values (or their 15-dimensional profiles) to ordinal rating distributions, introducing additional modeling assumptions. Moreover, Wasserstein distances between empirical distributions can be sample-sensitive in small-N regimes (Fournier and Guillin, 2015; Panaretos and Zemel, 2019).

We therefore do a conservative intermediate test: do models recover these classes of human linguistic variation from $PR_\theta$ features? In the first experiment at Sec.5.2, we test whether models do so without supervision via a coarse classification (HIP, ENP, Mixable (UN+MP)), and then in Sec.5.3, 5.4 we study supervised recovery of the full 4-way labels.

[13]Tab.7 lists specific verb pairs in each category.

| Model Name | Accuracy | Macro-Precision | Macro-Recall | Macro-F1 |
|---|---|---|---|---|
| Gemma3-1b | 0.37 | 0.24 | 0.28 | 0.21 |
| Gemma3-4b | 0.37 | 0.22 | 0.28 | 0.22 |
| Gemma3-12b | 0.37 | 0.30 | 0.28 | 0.22 |
| Llama3.1-8B | **0.40** | **0.36** | **0.39** | **0.33** |
| Llama3.1-70B | 0.33 | 0.20 | 0.25 | 0.20 |
| Sarvam-1 | 0.20 | 0.28 | 0.32 | 0.20 |
| Qwen3-32B | 0.27 | 0.31 | 0.37 | 0.26 |
| Random Baseline | 0.27 | 0.29 | 0.37 | 0.27 |

Table 2: Exp. 1 stats, Bold = column best across models.

### 5.2 Exp. 1: Do LLMs model Human Variation in Hindi-English Verb Code-Mixing?

This represents the hypothetical use case: a linguist using an LLM as a proxy for human linguistic variation on novel items (Hinglish verb code-mixing).

**Setup:** For each verb pair, we conducted a majority vote across all 15 conditions based on their $PR_\theta$. If a majority of the conditions ($> 8$) showed the same language preference we label the verb as HIP/ENP, and Mixable otherwise. This yielded model predictions for all 30 verb pairs to compare against human classes using classification metrics.

**Results:** Tab.2 shows:

- **All models perform like the random baseline** (Macro-F1 = 0.27), none statistically significantly different from it[14]. Only Llama3.1-8B is slightly above baseline (F1 = 0.33).
- **No positive model size effect**: Scaling within Gemma does not change performance, and within Llama drops performance. Across families, Llama 70B performs identically to the much smaller Sarvam 2B (F1 = 0.20).
- **Pretraining language mixes do not help** as the only model (Sarvam-1) reportedly pretrained on a substantial Indic language mix, performs the worst, while Qwen3-32B which performs much better Chinese, also performs better (near baseline) than Sarvam.

**Model behavior patterns:** Examining model confusion matrices (Tabs.3-4) reveals *strong but random biases*. Llama and Gemma prefer Hindi, while *Sarvam prefers English*, misaligned with its training composition. Qwen is more balanced but is unrelated to humans. These patterns indicate that model preferences *neither reflect training distributions nor align with human acceptability*.

[14]McNemar's test on aggregated predictions after Bonferonni corrections, $p > 0.007$.

| Human \| Model | Eng | Hin | Mix |
|---|---|---|---|
| EnglishPreferred (n=3) | 0 | 2 | 1 |
| HindiPreferred (n=13) | 1 | 9 | 3 |
| MultiPeak (n=2) | 0 | 2 | 0 |
| Uniform (n=12) | 1 | 10 | 1 |
| **Total** | **2** | **23** | **5** |

(a) Llama3.1-70B

| Human \| Model | Eng | Hin | Mix |
|---|---|---|---|
| EnglishPreferred (n=3) | 1 | 2 | 0 |
| HindiPreferred (n=13) | 1 | 10 | 2 |
| MultiPeak (n=2) | 0 | 2 | 0 |
| Uniform (n=12) | 1 | 10 | 1 |
| **Total** | **3** | **24** | **3** |

(b) Llama3.1-8B

| Human \| Model | Eng | Hin | Mix |
|---|---|---|---|
| EnglishPreferred (n=3) | 2 | 0 | 1 |
| HindiPreferred (n=13) | 7 | 3 | 3 |
| MultiPeak (n=2) | 1 | 1 | 0 |
| Uniform (n=12) | 5 | 4 | 3 |
| **Total** | **15** | **8** | **7** |

(c) Qwen3-32B

| Human \| Model | Eng | Hin | Mix |
|---|---|---|---|
| EnglishPreferred (n=3) | 2 | 0 | 1 |
| HindiPreferred (n=13) | 8 | 3 | 2 |
| MultiPeak (n=2) | 1 | 1 | 0 |
| Uniform (n=12) | 9 | 2 | 1 |
| **Total** | **20** | **6** | **4** |

(d) Sarvam-1

Table 3: Confusion matrices for Llama, Qwen, and Sarvam models. Note Llama's strong Hindi bias, Sarvam's English bias, and Qwen's more balanced but still misaligned predictions.

| Human \| Model | Eng | Hin | Mix |
|---|---|---|---|
| EnglishPreferred (n=3) | 0 | 3 | 0 |
| HindiPreferred (n=13) | 1 | 10 | 2 |
| MultiPeak (n=2) | 0 | 2 | 0 |
| Uniform (n=12) | 0 | 11 | 1 |
| **Total** | **1** | **26** | **3** |

(a) Gemma3-1B

| Human \| Model | Eng | Hin | Mix |
|---|---|---|---|
| EnglishPreferred (n=3) | 0 | 2 | 1 |
| HindiPreferred (n=13) | 1 | 10 | 2 |
| MultiPeak (n=2) | 0 | 2 | 0 |
| Uniform (n=12) | 1 | 10 | 1 |
| **Total** | **2** | **24** | **4** |

(b) Gemma3-4B

| Human \| Model | Eng | Hin | Mix |
|---|---|---|---|
| EnglishPreferred (n=3) | 0 | 2 | 1 |
| HindiPreferred (n=13) | 3 | 10 | 0 |
| MultiPeak (n=2) | 0 | 2 | 0 |
| Uniform (n=12) | 0 | 11 | 1 |
| **Total** | **3** | **25** | **2** |

(c) Gemma3-12B

Table 4: Confusion matrices for Gemma model family. All variants show strong Hindi bias, no scaling effect.

**Implication:** Unsupervised LLMs do not model human linguistic variation. A linguist asking "which mix sounds more natural?" on a similar phenomenon might receive responses no better than chance. Llama 3.1-8B's performance is not useful in a novel linguistic variation study. Note that the scaling is also **not consistent with** Sterner and Teufel (2025)'s findings in the Llama family, where scaling sizes made them more aligned to human judgments in *code-switching*.

As models do not reliably recover even these liberal classes from $PR_\theta$ in the unsupervised setting, we do not do a full 7-bin matching. We instead ask *"do LLMs supervised with some human judgments reveal latent abilities to model the rest?"*

### 5.3 Exp. 2(a): Supervised 4-way classification

This experiment represents the use case of an NLP researcher leveraging LLMs as a proxy for human judgments of linguistic variation on novel items given access to some labeled gold data. If models contain systematic linguistic knowledge about Hinglish verb mixing acceptability, this oracle access to labels should reveal it.

Exp1 (Sec.5.2) showed LLMs do not model linguistic variation in the phenomenon. We now test if we can train classifiers built on $PR_\theta$ based features identified using some gold human labels to predict verb mixability preference types.

**Feature Setup:** We use $PR_\theta$ obtained from the 15 conditions as 'features', and compute Spearman correlations between each of these and human rat-

| Feature set | Accuracy | Macro-Precision | Macro-Recall | Macro-F1 |
|---|---|---|---|---|
| All Selected Features | 0.27 | 0.14 | 0.20 | 0.16 |
| above w/o Sarvam | 0.28 | 0.19 | 0.21 | 0.20 |
| Best for Llama3.1-8B | **0.43** | **0.43** | **0.53** | **0.43** |
| Best for Llama3.1-70B | 0.33 | 0.34 | 0.40 | 0.34 |
| Best for Sarvam-1 | **0.43** | 0.41 | 0.35 | 0.34 |
| Best for Qwen3-32B | 0.33 | 0.28 | 0.30 | 0.29 |
| Random Baseline | 0.25 | 0.24 | 0.29 | 0.25 |

Table 5: Exp2(a) stats, Bold = column best.

| Config | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| All Selected Features | 0.72 | 0.71 | 0.73 | 0.72 |
| above w/o Sarvam | 0.65 | 0.63 | 0.73 | 0.68 |
| Best for Llama3.1-8B | 0.45 | 0.41 | 0.23 | 0.30 |
| Best for Llama3.1-70B | 0.63 | 0.65 | 0.57 | 0.61 |
| Best for Sarvam-1 | **0.80**$^{*}$ | **0.76** | **0.87** | **0.81** |
| Best for Qwen3-32B | 0.63 | 0.75 | 0.40 | 0.52 |
| Random Baseline | 0.47 | 0.46 | 0.43 | 0.45 |

Table 6: Exp2(b) stats, * = significant diff. from random

ings. Note that we restricted this analysis to sentences from HIP and ENP verb pairs (n=52), where humans prefer one language clearly (Tab.10).

**Features Selection:** The correlation (based on a grid of 7 (models) x 15 (features)) analysis revealed heterogeneity across models. We prune non-Devanagari Gemma features from the feature set as they mostly showed positive correlations [15].

For the remaining models, we selected the four frames in Devanagari[16] that showed consistent negative correlations (all except *X(kar)na aasaan hai*).

**Classification:** We trained L2-regularized logistic regression classifiers with balanced verb class weights, grid search, and scaling to use the above features over 10 repeated randomly sampled 80:20 train-test splits, ensuring fair class representation.

**Results:** Tab. 5 shows that:

- **Performance remains weak** all combinations are near random baseline ($p > 0.008$).
- **No model size effect emerges again** after exploring individual models' features predictive strength, ("Best for X"). Smaller models (Llama 8B, Sarvam (2B)) match or outperform bigger models (Llama 70B, Qwen 32B).
- **Pretraining language mix seems to help** as Sarvam shows high accuracy with low Macro-F1, because of wrong MP prediction.

**Implication:** For the NLP researcher, even with human-labeled feature selection and exhaustive search, 4-way classification is barely above baseline. The optimal strategy to achieve better scores required model-specific feature selection, thereby decreasing utility, as this feature selection is model-specific. Even then, most models' best feature sets perform no better than random, and removing Sarvam features from the full model-feature set only marginally improves performance to F1 = 0.20, which is still unreliable and impractical.

However, this task faces class imbalance challenges (e.g. HIP (n=13) vs. MP (n=2)). We further simplify the task to binary classification: *do LLMs detect when the Hindi form is strongly preferred?*

### 5.4 Exp. 2(b): Supervised 2-way Classification

The supervision setup is the same as above, but the binary labels HIP (n=13,43%) and not-HIP (n=17, 57%) reduce the confounding effect of class imbalance. As this is a substantial simplification, failure in classification would mean that LLM perplexities are unreliable models of human judgment variation.

**Results:** Tab. 6 shows:

- **Performance improves across all models**. The set with all features is well above the binary random baseline, and Sarvam-1's best feature set achieves **F1 = 0.81** (significantly away from random $p < 0.008$).
- **Size effects are more consistent** as models perform better with size, except Sarvam, which is the smallest but best in this condition.

**Implication:** Sarvam emerges as the strongest performer with supervision (near-best in Exp2(a), best in Exp2(b)) after failing unsupervised prediction (Exp1). This suggests its Indic-focused pretraining may encode latent code-mixing knowledge that surfaces only with supervision, consistent with claims about Hindi and code-mixed pre-training.

Thus, LLMs do not model human linguistic variation, as shown by Hindi-English verb code-mixing; however, on an easier classification objective, supervised logits probing shows some predictive ability.

[15] Opposite the desired negative direction where lower perplexity should indicate higher preference of Hindi verb.

[16] This is interesting as it was the only form where English verbs were visibly in a different orthography (Latin) than the Hindi verbs and frames (Devanagari).

## 6 Discussions

### 6.1 Answering the Research Question

Our central question is: *Do LLMs model human linguistic variation in Hindi-English do verb code-mixing?* We approached this through a three-step evaluation: In **Exp. 1**, we asked the most direct and "LLM-native" question: do perplexity ratios over minimal pairs model whether humans tend to prefer the Hindi or the Hinglish verb form, or find the pair easily mixable? All models performed close to the random baseline, with no consistent benefit from pretraining language mix or scaling within a family. Thus, LLMs do not encode this variation in a way that is straightforwardly accessible through standard surprisal-based probing.

In **Exp. 2(a)**, we gave models more help: we treated the 15 perplexity ratios as features and supervised lightweight L2-regularized logistic regression classifiers to model all preference classes. Note that this is not fine-tuning the LLM or performing preference optimization but an *external probe over model-derived features*, analogous to factor analysis. Even with this oracle access to human labels and hand-selected features, performance only slightly exceeds chance and remains weak across families, pretraining language mixes, and sizes. In **Exp. 2(b)**, we simplify the task further to a conservative binary distinction between clearly Hindi-preferred verb pairs (`HIP`) from all others. Performance now improves, with the best configuration achieving F1 = 0.81. Thus, this is an optimistic upper bound because it depends on model-specific feature choices and an exhaustive feature sweep.

Taken together, these three experiments consistently show that for this well-controlled phenomenon, LLMs do not model human preferences in Hinglish verb code-mixing. Some models contain extractable signals, but recovering them requires careful feature engineering, task simplification and human labels, and the resulting predictors do not generalise across models or decision boundaries. At the same time, the three-step progression from Exp. 1 to 2(b) makes for a reusable LLM evaluation template for linguistic variation. It separates questions about what signal is present in a base model from questions about how much supervised probing can extract, and can be adapted to other variation phenomena without committing to preference-training the LLMs themselves.

### 6.2 Why Even "Success" Represents Fundamental Limitations

The Sarvam-1 result in Exp. 2(b) might appear to show that supervised approaches solve the problem: with the right feature subset, a probe reaches F1 $=$ $0.81$ on the binary task. However, there are several limitations that matter for practicality:

**Model-specific and non-generalisable.** The same probing workflow that yields high F1 for Sarvam-1 produces substantially lower scores for other models (e.g., around 0.61 for Llama3.1-70B and near-chance for Llama3.1-8B). There is no principled way, a priori, to know which model, which frames, or which orthographies will contain the most informative signal for a given phenomenon.

**Coarse-grained modeling of a rich phenomenon.** Our binary labels collapse the original 7-point ratings and distributional shapes into a yes/no decision about strong Hindi preference. This discards the gradient acceptability, contextual sensitivity, and speaker-level variability that a linguistic analysis of a novel phenomenon might seek to capture.

**Dependence on human labels rather than replacement.** This pipeline requires human preference data to select features, define classes, and evaluate performance. Ideally, a model of language that could be used to model human linguistic variation should not require more specific human data.

**Optimistic upper bounds.** Finally, we explicitly present the results as optimistic upper bounds over our exhaustive search over feature subsets in Exp. 2(b). Even under the best case, only one model achieves strong performance on the simple task.

### 6.3 What This Reveals About LLMs and Linguistic Variation

We do not observe a consistent scaling effect, i.e. within families, larger models do not systematically outperform smaller ones. We also do not observe the advertised differences in training language mix (e.g., Sarvam's Indic and Qwen's Chinese focus) correlate with some trends but do not straightforwardly explain unsupervised or cross-family behaviour. We treat this pattern of inability more as alignment than as firm causal claims.

More broadly, our findings are consistent with the view that standard next-token prediction on large web corpora is well-suited to learning average patterns and aggregate statistics, but not necessarily

to capturing the fine-grained, context- and speaker-dependent variation that characterises code-mixing preferences. Models exhibit family-level biases (e.g., some tending to prefer Hindi verbs, others English) that are not obviously justified by human data and are difficult to link back to specific corpus properties. This suggests a mismatch between the kinds of signals that matter for linguistic variation and the signals emphasised during pretraining.

### 6.4 Why Does This Question Matter?

Our narrow, tightly controlled setup is deliberate. From a Popperian perspective, even one well-specified counterexample matters: if LLMs fail to align with human preferences in a simple verb-choice alternation under their next-token objective, this challenges broad claims that such models generally capture human linguistic behaviour. By Occam's razor, the simplest explanation is that current training regimes do not yet endow base models with an ability to model natural linguistic variation, rather than assuming that more complex or abstract forms of variation are nonetheless encoded but remain inaccessible.

### 6.5 Conclusion

LLMs align well with human judgments on formal phenomena with relatively uniform speaker behavior (Jumelet et al., 2025), but for the specific case of Hindi-English *do* verb code-mixing, our findings clearly demonstrate that current LLMs cannot be used reliably for linguistic studies even for languages with **high to mid-range resources** (English is in class 5 and Hindi is in class 4 in Joshi et al. (2020) classification scheme - the top two classes)

Our findings also open several promising research directions. At the phenomenon level, future work should examine whether our results generalize to other code-mixing types such as adjective–noun alternations, discourse markers, or pragmatic particles; and to syntactic alternations like word order variation in mixed clauses. Testing across other bilingual communities (Spanish–English, Arabic–French, Cantonese–English) for subjective phenomena in the respective mixes would clarify whether the patterns we observe are specific to Hindi–English or reflect broader limitations in how LLMs handle linguistic variation.

More broadly, our study also contributes (i) a three-step experimental template to probe LLM–human alignment in linguistic variation: unsupervised minimal-pair perplexity, supervised class modeling via simple perplexity ratio based classifier probes, and a conservative binary version of the same; (ii) the first systematic human acceptability dataset for Hinglish *do* verb code-mixing; and (iii) parallel perplexity-based measurements for seven open-weight models on the same material. Together, these resources highlight the need for models and training paradigms that develop genuine linguistic variational competence rather than only capturing aggregate statistics: modeling how different communities use language, including expressing disagreement or politeness, requires representing multiple valid perspectives rather than a single average (Saha et al., 2025).

## 7 Limitations

**Sample size and scope.** Our human study uses 30 verb pairs and 14 annotators, chosen to span the observed preference space under a carefully controlled design. Such small yet structured samples are common in (psycho)linguistics, but they limit the breadth of our claims.

**Demographic diversity and linguistics.** Our annotators are native Hindi speakers fluent in English, primarily from central and northern India. This demographic does not encompass the full diversity of Hinglish speakers across regions, ages, and social groups. We focus on linguistic acceptability rather than other possible linguistic variables. Although we collected some demographic metadata and observed weak trends (e.g., age-related differences in English preference), our sample is underpowered for strong claims; instead, we release the data to support future work that targets these questions.

**Model selection and training regimes.** We restrict attention to seven open-weight *base* models to avoid confounds from instruction-tuning and RLHF. We do not study closed-source models or models explicitly trained with preference-optimisation methods such as DPO. Consequently, our negative findings apply to these base models and to this phenomenon; they do not preclude the possibility that suitably trained or specialised models might better capture variation. Moreover, since detailed pretraining corpora are not publicly documented, our remarks about the role of Indic or code-mixed data in pretraining should be read as hypotheses rather than definitive statements about data composition.

**Methodological constraints.** Our main probing signal is per-token perplexity, used via within-model perplexity ratios over minimal pairs. This

aligns with standard practice in computational psycholinguistics, but it is not the only way to interrogate models. In Exp 2(a) and 2(b), we employ simple logistic regression probes over these ratios, rather than more complex representation-based probes, causal interventions, or sequence-level preference models. In addition, the exhaustive feature sweep in Exp 2(b) uses no held-out validation, so the best-reported numbers are optimistic upper bounds. A more conservative protocol with validation or cross-validation would likely lower absolute scores but, given the already weak unsupervised and multi-way results, is unlikely to reverse our qualitative conclusions. Future methodological work could explore richer probing signals and architectures, as well as alternative evaluation metrics such as distributional distance measures, once larger human datasets are available.

## 8 Ethics Statement

**Human subjects research.** All human data collection followed ethical research practices. Participants were recruited through Prolific, and informed consent was obtained within the survey. Participation was voluntary, and participants could withdraw at any time. Participants were compensated at an hourly rate of £8 (per Prolific's recommended pay guidelines) and took approximately 10-15 minutes to complete the survey. No personally identifiable information was collected beyond demographic data necessary for participant screening.

**Data release and privacy.** We will publicly release anonymised preference ratings, perplexity ratios, and experimental materials. All released data contains no personally identifiable information. Demographic summaries are presented in aggregate form only. In addition, we release the full rating distributions and basic non-identifiable demographic metadata so that future work can explore sociolinguistic patterns beyond the linguistic focus of this paper.

**Potential biases.** Our participant sample (primarily from central and northern India, ages 25–61) may not represent the full diversity of Hinglish speakers, potentially biasing our characterisation of code-mixing preferences toward specific age and regional groups. Our findings should therefore not be interpreted as universal patterns across all Hindi–English bilinguals.

**Broader impacts.** This work demonstrates limitations of using current LLMs as proxies for human linguistic behaviour in a specific code-mixing phenomenon. While our negative findings discourage certain kinds of unsupervised or lightly supervised applications, they also highlight the continued necessity of human participation in linguistic research. Our released resources enable replication and extension but should not be used to make claims about individual speakers' linguistic competence or to enforce prescriptive norms about "correct" code-mixing.

**Model use and Environmental considerations.** Model evaluation required approximately 10 GPU hours on 2× NVIDIA RTX 6000 GPUs. We used existing pretrained models to minimise computational costs and environmental impact. Finally, we also used LLMs for basic grammar checks and proofreading.

## 9 Acknowledgements

This research was supported, in parts, by the Microsoft Azure Foundation Model Research (AFMR) Grant. We thank all the participants involved in the human preferences study.

## References

Sarvam AI. 2024. Sarvam 1: The first indian language llm. https://www.sarvam.ai/blogs/sarvam-1.

Lisa P. Argyle, Ethan C. Busby, Nancy Fulda, Joshua R. Gubler, Christopher Rytting, and David Wingate. 2023. Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3):337–351.

Lucia Badiola, Rodrigo Delgado, Ariane Sande, and Sara Stefanich. 2018. Code-switching attitudes and their effects on acceptability judgment tasks. *Linguistic Approaches to Bilingualism*, 8(1):5–24.

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "I am borrowing ya mixing ?" an analysis of English-Hindi code mixing in Facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126, Doha, Qatar. Association for Computational Linguistics.

Pushpak Bhattacharyya. 2010. IndoWordNet. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Anik Dey and Pascale Fung. 2014. A Hindi-English code-switching corpus. In *Proceedings of the Ninth International Conference on Language Resources and*

*Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Nicolas Fournier and Arnaud Guillin. 2015. On the rate of convergence in wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3–4):707–738.

Björn Gambäck and Amitava Das. 2016. Comparing the level of code-switching in corpora. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1850–1855.

Yuan Gao, Dokyun Lee, Gordon Burtch, and Sina Fazelpour. 2025. Take caution in using llms as human surrogates: Scylla ex machina. *Preprint*, arXiv:2410.19599.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Rohit Gupta, Pulkit Goyal, and Sapan Diwakar. 2010. Transliteration among indian languages using wx notation. In *KONVENS*, pages 147–150.

John Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Luke Hewitt, Ashwini Ashokkumar, Isaias Ghezae, and Robb Willer. 2024. Predicting results of social science experiments using large language models. *preprint*.

Yutong Hu, Quzhe Huang, Mingxu Tao, Chen Zhang, and Yansong Feng. 2024. Can perplexity reflect large language model's ability in long text understanding? In *The Second Tiny Papers Track at ICLR 2024*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Jaap Jumelet, Leonie Weissweiler, and Arianna Bisazza. 2025. MultiBLiMP 1.0: A massively multilingual benchmark of linguistic minimal pairs. ArXiv:2504.02768 (Preprint).

Shivani Kapania, William Agnew, Motahhare Eslami, Hoda Heidari, and Sarah E Fox. 2025. Simulacrum of stories: Examining large language models as qualitative research participants. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA. Association for Computing Machinery.

Prashant Kodali, Anmol Goel, Likhith Asapu, Vamshi Krishna Bonagiri, Anirudh Govil, Monojit Choudhury, Ponnurangam Kumaraguru, and Manish Shrivastava. 2025. From human judgements to predictive models: Unravelling acceptability in code-mixed sentences. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 24(9):1–31.

Ashok Kumar. 1986. Certain aspects of the form and functions of hindi-english code-switching. *Anthropological Linguistics*, pages 195–205.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202.

Carol Myers-Scotton. 1997. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.

Victor M. Panaretos and Yoav Zemel. 2019. Statistical aspects of wasserstein distances. *Annual Review of Statistics and Its Application*, 6(1):405–431.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553.

Prolific. 2025. Prolific. `https://www.prolific.com`. Online participant recruitment platform.

Sougata Saha, Saurabh Kumar Pandey, and Monojit Choudhury. 2025. Meta-cultural competence: Climbing the right hill of cultural awareness. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8025–8042, Albuquerque, New Mexico. Association for Computational Linguistics.

Ayan Sengupta, Soham Das, Md. Shad Akhtar, and Tanmoy Chakraborty. 2024. Social, economic, and demographic factors drive the emergence of hinglish code-mixing on social media. *Humanities and Social Sciences Communications*, 11(1):606.

Igor Sterner and Simone Teufel. 2025. Minimal pair-based evaluation of code-switching. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18575–18598. Association for Computational Linguistics.

Gijsbert Stoet. 2010. PsyToolkit: A software package for programming psychological experiments using Linux. *Behavior Research Methods*, 42(4):1096–1104.

Gijsbert Stoet. 2017. PsyToolkit: A novel web-based method for running online questionnaires and reaction-time experiments. *Teaching of Psychology*, 44(1):24–31.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Leonie Weissweiler, Kyle Mahowald, and Adele E. Goldberg. 2025. Linguistic generalizations are not rules: Impacts on evaluation of LMs. In *Proceedings of the Second International Workshop on Construction Grammars and NLP*, pages 61–74, Düsseldorf, Germany. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP 2020: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Yilun Yang and Yekun Chai. 2025. CodeMixBench: Evaluating code-mixing capabilities of LLMs across 18 languages. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2139–2169, Suzhou, China. Association for Computational Linguistics.

# A Transliteration

## A.1 Formal Transliteration

Formal transliteration follows established Devanagari romanization conventions with systematic mapping rules. We follow the transliteration system in Table 9 for the *Formal Latin* form. Our approach uses chain-of-thought reasoning with six processing stages: decomposition, analysis, mapping, rule application, combination, and verification.

| **Distribution Class** | **Verb Tuples** |
|---|---|
| SinglePeakRight | (पढ-read), (मिटा-rub), (बना-create), (फाड़-tear), (रो-weep), (पढ-study), (तरस-yearn), (चल्-move), (भगा-speed),(रोक-hinder), (देख्-witness), (गिन-enumerate), (पीस-pulverise) |
| SinglePeakLeft | (खौला-boil), (चूम-kiss), (महक-smell) |
| MultiPeak | (बजा-ring), (सोच-imagine) |
| Uniform/Random | (ले-collect), (छू-touch), (बदल-change), (खटखटा-knock), (छिड़क-spray), (भाँप-gauge), (भेद-pierce), (पिरो-string), (घबड़ा-panic), (लुभा-seduce), (फूट-crack), (ललकार-dare) |

Table 7: Verb distribution by class

| Statistic | Value |
|---|---|
| Count | 1044 |
| Mean | 1.10 |
| Std | 0.37 |
| Min | 1.00 |
| 25% | 1.00 |
| 50% | 1.00 |
| 75% | 1.00 |
| Max | 5.00 |
| Consistency Rate (%) | 91.19 |

Table 8: Distribution of Unique Spellings over 5 frames per Formal Transliterated verb

We employed verb spelling consistency rate as our primary quality metric—calculated by measuring identical transliterations when the same verb is substituted across five different sentence frames. This serves as an effective non-gold-standard metric for evaluating rule adherence without requiring manual annotation. This methodology achieved 91.19% consistency with a mean of 1.10 unique spellings per verb (SD=0.37, N=1,044 verbs) providing high confidence in the reliability of outputs. Analysis against established romanization rules identified systematic errors detailed in Appendix A.1.

### A.1.1 Prompt Design

Formal transliteration required extensive prompt architecture due to the complexity of Devanagari-to-Latin mapping rules. Despite the rule-based nature of this task, significant prompt engineering

was required to achieve reliable outputs, given the complexity of Devanagari orthography and the challenges of handling tokenization boundaries in neural language models. Our development process incorporated the following refinements (refer Figure Box 1 for final prompt):

1. Basic Mapping Rules: Initial prompt included character-to-character correspondences
2. Case Sensitivity: Added explicit rules for errors w.r.t. letter case handling
3. Nasalization Rules: Incorporated rules handling consonant clustering and nasalization
4. Chain-of-Thought Reasoning: Implemented six-step decomposition process (decomposition, analysis, mapping, rule application, combination, and verification)
5. Few-Shot Examples: Added exemplars showing complete reasoning chains (Prompt Box 3)
6. Self-Verification: Included re-verification step for error detection and correction

#### A.1.2 Error Analysis

Validation of formal transliteration outputs against established Devanagari romanization rules revealed three systematic error categories: (1) nasalization handling errors, particularly with 'm' nasal exceptions (e.g., संभल → "sanbhala" instead of "sambhala"), (2) character merging mistakes during combination steps (e.g., अटका → incorrect reasoning "'a' + 'Ta' + 'kaa' = 'aTaka'" instead of correct "'aTakaa'"), and (3) consonant clustering overtriggering with hallucinated halants creating false clusters (e.g., निपट → model incorrectly identifying प् + ट cluster where none exists). Despite these systematic errors, the distribution reveal exceptional consistency (refer table 8), with 91% consistency rate and a mean value close to 1 (1.10) with a low standard deviation (0.37) indicating sufficient constraint on formal transliteration by the provided rules leading to reliable outputs.

#### A.1.3 Frame Skeleton Analysis

Following is the frame skeleton analysis for all the 5 frames when transliterated where the verb gets substituted with X. We take the most popular frame spellings as the final frames using this analysis where the dictionary for each frame/format is structured as the actual spellings as the key with the total instance count as where this spellings were found as the value. This reveals high frame-level inconsistency, requiring limitation to the top 5 variants per frame due to substantial variability. This variance, despite formal transliteration's rule-based nature, results from computational complexity in the 6-step process involving character-level decomposition and token-level mapping operations. The most frequent variant was selected as the canonical frame for each format. The significant frame-level variance contrasts sharply with non-standardized transliteration's structural consistency, demonstrating the challenges of maintaining uniformity in complex multi-step linguistic processing despite rule-based constraints.

```
Format 1 Skeletons:
{'vo X rahe hain': 804, 'vao X rahe
↪  hain': 114, 'vo X rahe han': 57,
↪  'vao X rahe han': 32, 'vao X rahae
↪  hain': 16, ...}

Format 2 Skeletons:
{'Xnaa aasaan hai': 634, 'Xnaa aasaan
↪  haai': 95, 'Xna aasaan hai': 87,
↪  'Xna aasana hai': 82, 'Xnaa aasana
↪  hai': 38, ...}

Format 3 Skeletons:
{'vo kala X chuke the': 782, 'vao kala X
↪  chuke the': 90, 'vao kala X chukae
↪  thae': 64, 'vo kala X chukae thae':
↪  44, 'vao kala X chuke thae': 23,
↪  ...}

Format 4 Skeletons:
{'Xnaa manaa honaa chaahie': 436, 'Xnaa
↪  manaa honaa chaahiye': 222, 'Xnaa
↪  manaa honaa chaahiie': 70, 'Xna mana
↪  hona chaahiye': 57, 'Xnaa manaa
↪  honaa chaahiiye': 25, ...}

Format 5 Skeletons:
{'vo aaraama se Xte rahenge': 485, 'vao
↪  aaraama sae Xte rahenge': 137, 'vao
↪  aaraama se Xte rahenge': 94, 'vao
↪  araamaa sae Xte rahenge': 60, 'vao
↪  araama sae Xte rahenge': 53, ...}
```

### A.2 Non-standardized Transliteration

Non-standardized transliteration produces a popular version of the Devanagari counterpart, capturing how Hindi speakers naturally type Hindi words using standard English keyboards in informal digital contexts. This approach generates phonetic Latin representations reflecting authentic user behavior rather than prescribed schemes. We achieved a 47.03% consistency rate with a mean of 1.559 unique spellings per verb (SD=0.76, range=1-5). This moderate consistency reflects the inherent variability expected in non-standardized romanization

practices. Frame Skeleton analysis of the most frequent frame patterns selected canonical frames for subsequent experiments, ensuring stimuli reflect authentic usage patterns based on popularity.

### A.2.1 Verb Consistency Analysis

Since non-standardized transliteration captures popular usage patterns rather than rule-based accuracy, outputs were accepted based on frequency and consistency patterns. Analysis revealed systematic variations: (1) vowel length inconsistencies (e.g., "aasan" vs "asan"), (2) consonant doubling variations (e.g., "chuke" vs "chukke"), and (3) phonetic approximations of retroflex sounds, reflecting authentic user behavior in digital contexts. The verb consistency distribution (refer Table 12) exhibits moderate variability characteristic of non-standardized systems. While 25% of verbs achieve perfect consistency (single spelling), the median of 2.0 indicates that typical verbs generate two distinct spellings across frames. The higher standard deviation (0.76) compared to formal transliteration reflects authentic user behavior variability, where personal typing preferences and phonetic interpretations influence output. Notably, both approaches share the same maximum variance (5 unique spellings), but this represents different underlying causes: computational errors in formal transliteration versus legitimate alternative representations in non-standardized transliteration. The 47.03% consistency rate, while lower than formal systems, indicates substantial convergence toward popular spellings, validating the existence of informal but recognizable standards in digital Hindi typing practices.

### A.2.2 Frame Skeleton Analysis

Similar to Formal transliteration following is the frame skeleton analysis for Non-standardized transliteration demonstrating exceptional structural consistency. The dominant frames achieve near-perfect consistency (>97% across all formats), validating that structural elements are more standardized than lexical elements in non-standardized transliteration.

```
Format 1 Skeletons:
{'wo X rahe hain': 1044}

Format 2 Skeletons:
{'Xna aasan hai': 1036, 'X na aasan hai':
↪  5, 'Xnaa aasan hai': 3}

Format 3 Skeletons:
{'wo kal X chuke the': 1042, 'wo Xal k
↪  chuke the': 1, 'wo kaX l chuke the':
↪  1}

Format 4 Skeletons:
{'Xna mana hona chahiye': 1037, 'X na
↪  mana hona chahiye': 6, 'Xnā mana
↪  hona chahiye': 1}

Format 5 Skeletons:
{'wo aaram se Xte rahenge': 1022, 'wo
↪  aaram se Xthe rahenge': 5, 'wo aaram
↪  se X te rahenge': 3}
```

## B Verb list pre-processing

1. From all entries, we picked the ones POS tagged as the main verb: `V`, `V.VINT`, etc. This resulted in a list of 34,510 entries out of 136,154 total.

2. We cleaned the list further to obtain verb roots by removing:

   - **Phrasal verbs**: entries with spaces on either the Hindi or English sides,
   - **Compound verbs/Inflected verb forms**: entries with "–" or "," (such as गा, गे, गीई),
   - **Transliteration errors**: such as बनाअ.

## C Model details

At Tab.14

## D Participant Demographics

Participants' ages ranged from 25 to 50 (mean 35.9), with one participant aged 61. Most participants were educated in English-medium institutions, with a few having Hindi-, Gujarati-, or Marathi-medium education. All participants were originally from central or northern states of India, and a few had recently lived abroad (e.g., in the UK, Canada, or USA). Nine participants reported Hindi as their only native language, while the remaining five were native bilinguals (Hindi–Marathi or Hindi–English).

## E Testing Uniformity of Human Responses

We tested the null hypothesis $H_0$ that the Likert-scale responses were drawn from a uniform distribution across 7 categories. Let $N$ be the total number of responses and $O = (O_1, \ldots, O_7)$ the observed counts. Under $H_0$, each category has probability $p_i = 1/7$, and the probability of observing $O$ is

given by the multinomial PMF:

$$P(O \mid H_0) = \frac{N!}{O_1! \dots O_7!} \prod_{i=1}^{7} p_i^{O_i}.$$

Because $N$ was small, we used a Monte Carlo simulation to approximate the exact p-value: we generated $100,000$ datasets of size $N$ from the uniform multinomial distribution, computed the likelihood for each, and estimated the p-value as the proportion of simulated likelihoods less than or equal to the observed likelihood. We used a significance level of $\alpha = 0.05$ to determine statistical significance. This approach avoids large-sample approximations and provides an exact small-sample test of uniformity. Once we segregate uniform (UN) distributions we move to manually labelling the other samples as either single left peak (ENP, English preferred), single right peak (HIP, Hindi preferred) or Multiple (MP) peaks.

| **Hindi** | **Formal Latin** |
|---|---|
| क | ka |
| ख | kha |
| ग | ga |
| घ | gha |
| च | cha |
| छ | chha |
| ज | ja |
| झ | jha |
| ट | Ta |
| ठ | Tha |
| ड | Da |
| ढ | Dha |
| त | ta |
| थ | tha |
| द | da |
| ध | dha |
| न | na |
| प | pa |
| फ | fa |
| ब | ba |
| भ | bha |
| म | ma |
| य | ya |
| र | ra |
| ल | la |
| व | va |
| श | sha |
| स | sa |
| ह | ha |
| ड़ | Ra |
| ढ़ | Rha |
| अ | a |
| आ | aa |
| इ | i |
| ई | ii |
| उ | u |
| ऊ | uu |
| ए | e |
| ऐ | ai |
| ओ | o |
| औ | au |

Table 9: Devanagari - Formal Latin transliteration map

| Feature ↓ | Qwen3-32B | Gemma3-12b | Gemma3-1b | Gemma3-4b | Llama3.1-70B | Llama3.1-8B | Sarvam-1 |
|---|---|---|---|---|---|---|---|
| Frame0, Translit Casual | 0.079 | -0.115 | 0.103 | -0.273 | 0.430 | 0.358 | -0.188 |
| **Frame 0, Translit Devanagari** | **-0.564** | -0.345 | -0.394 | -0.261 | **-0.176** | **-0.636** | **-0.394** |
| Frame 0, Translit Formal | -0.018 | -0.103 | 0.333 | -0.164 | 0.285 | -0.006 | -0.273 |
| Frame 1, Translit Casual | 0.183 | 0.238 | 0.250 | 0.341 | 0.049 | -0.006 | 0.213 |
| Frame 1, Translit devanagari | -0.445 | 0.360 | 0.348 | 0.463 | -0.098 | -0.116 | -0.067 |
| Frame 1, Translit formal | 0.268 | 0.579 | 0.470 | 0.543 | 0.518 | 0.183 | 0.463 |
| Frame 2, Translit casual | 0.588 | -0.285 | -0.418 | -0.285 | 0.285 | 0.479 | 0.079 |
| **Frame 2, Translit devanagari** | **-0.624** | -0.079 | -0.345 | 0.079 | **-0.685** | -0.152 | 0.067 |
| Frame 2, Translit formal | 0.515 | 0.273 | 0.418 | 0.285 | 0.370 | 0.212 | -0.067 |
| Frame 3, Translit casual | 0.170 | 0.000 | 0.201 | 0.511 | -0.304 | -0.304 | 0.097 |
| **Frame 3, Translit devanagari** | **-0.340** | 0.401 | -0.261 | 0.535 | **-0.590** | **-0.310** | 0.085 |
| Frame 3, Translit formal | 0.085 | 0.085 | 0.128 | 0.188 | -0.049 | 0.079 | 0.170 |
| Frame 4, Translit casual | 0.193 | -0.249 | -0.354 | -0.168 | -0.291 | 0.119 | -0.095 |
| **Frame 4, Translit devanagari** | **-0.459** | -0.119 | 0.014 | 0.060 | **-0.515** | **-0.515** | **0.340** |
| Frame 4, Translit formal | 0.123 | 0.053 | 0.266 | 0.032 | -0.214 | 0.214 | 0.119 |

Table 10: Spearman correlation table between humans and models. Red indicates positive values, green indicates negative values, and bold marks selected cells.

| Feature ID | Feature Combination | English Frame | Hindi Frame |
|---|---|---|---|
| 0 | Frame 0, Translit Casual | wo X kar rahe hain | wo X rahe hain |
| 1 | Frame 0, Translit Devanagari | वो X कर रहे हैं | वो X रहे हैं |
| 2 | Frame 0, Translit Formal | vo X kara rahe hain | vo X rahe hain |
| 3 | Frame 1, Translit Casual | X karna aasan hai | Xna aasan hai |
| 4 | Frame 1, Translit Devanagari | X करना आसान है | Xना आसान है |
| 5 | Frame 1, Translit Formal | X karanaa aasaan hai | Xnaa aasaan hai |
| 6 | Frame 2, Translit Casual | wo kal X kar chuke the | wo kal X chuke the |
| 7 | Frame 2, Translit Devanagari | वो कल X कर चुके थे | वो कल X चुके थे |
| 8 | Frame 2, Translit Formal | wo kala X kara chuke the | wo kala X chuke the |
| 9 | Frame 3, Translit Casual | X karna manaa hona chahiye | Xna mana hona chahiye |
| 10 | Frame 3, Translit Devanagari | X करना मना होना चाहिए | Xना मना होना चाहिए |
| 11 | Frame 3, Translit Formal | X karanaa manaa honaa chaahie | Xnaa manaa honaa chaahie |
| 12 | Frame 4, Translit Casual | wo aaram se X karte rahenge | wo aaram se Xte rahenge |
| 13 | Frame 4, Translit Devanagari | वो आराम से X करते रहेंगे | वो आराम से Xते रहेंगे |
| 14 | Frame 4, Translit Formal | wo aarama se X karate rahenge | wo aarama se Xte rahenge |

Table 11: Feature frames and templates used in experiments.

## Box 1: Formal Transliteration Prompt | Part A

```
You are a meticulous and rule-based transliteration expert. Your sole task is to transliterate Hindi text from Devanagari script into the Latin script, following a precise set of rules. You must not deviate from these rules. Your work is methodical and verified.

# Procedure
Use the following mapping for transliteration.

## Mapping
Devanagari_Latin_mapping

## Rules
- **Rule 1 (Accuracy):** The transliteration must strictly follow the mapping provided. There are no exceptions. Be true to the source word without adding or omitting any characters or introducing halants or schwas that are not present in the original word.
- **Rule 2 (Case Sensitivity):** Letter case is critical. Preserve the case as specified in the mapping table (e.g., 'ट' is 'Ta', 'त' is 'ta').
- **Rule 3 (Consonant Clusters):** When a consonant is followed by a halant (्) and another consonant, this forms a consonant cluster. In clusters, only the final consonant gets the inherent 'a' sound.

Process:
Consonant + halant (्) → just the consonant sound (no 'a')
Final consonant in cluster → gets the 'a' sound (or whatever vowel follows)

Examples:
"चिल्ला" = चि + ल् + ल + आ = chi + l + la + a = chillaa
"नमस्ते" = न + म + स् + ते = na + ma + s + te = namaste
"अच्छा" = अ + च् + छ + आ = a + ch + chha + a = achchhaa
- **Rule 4 (Nasalization):** All nasals, including the bindu (ं), chandrabindu (ँ), and other nasal forms (like ङ, ञ, ण, ऩ) must be transliterated as 'n'. The only exception to this rule is when the nasal is an 'm', there you may use 'm'. For example, "हंस" is "hansa", and "ढूँढ" is "DhuunDha" but for "कांप" it is "kaampa" and not "kaanpa".
- **Rule 5 (Examples):** Observe the examples below to understand the nuances.

### Examples
- Input: अटक
  Output: अ (a) + ट (Ta) + क (ka) = aTa + ka = aTaka
- Input: उकसा
  Output: उ (u) + क (ka) + सा (sa + a) = uka + saa = ukasaa
- Input: उछल
  Output: उ (u) + छ (chha) + ल (la) = uchhala = uchhala
- Input: ओढ
  Output: ओ (o) + ढ (Rha) = o + Rha = oRha
- Input: ढूँढ
  Output: ढ (Dha) + ऊ (uu) + ँ (n) + ढ (Dha) = Dhuun + Dha = DhuunDha
- Input: फांद
  Output: फ (fa) + आ (aa) + ं (n) + द (da) = faan + da = faanda
- Input: लौटा
  Output: ल (la) + औ (au) + ट (Ta) + आ (aa) = lau + Taa = lauTaa
- Input: बांट
  Output: ब (ba) + आ (aa) + ं (n) + ट (Ta) = baan + Ta = baanTa
- Input: खटखटा
  Output: ख (kha) + ट (Ta) + ख (kha) + ट (Ta) + आ (aa) = khaTa + kha + Taa = khaTakha + Taa = khaTakhaTaa

# Steps
1. **Analyze and Decompose:** Read the input Devanagari word along with the transliterated sentence provided where the word will be substituted for X. Break the word down into its fundamental units as individual characters or syllables or consonant-vowel unit. Identify any consonant clusters. For "नमस्ते", the units are "न", "म", "स्", "ते".
2. **Initial Mapping match:** Transliterate each unit using the provided `Mapping` table. For "नमस्ते", "न" -> "na", "म" -> "ma", "स्" -> "s" and not "sa", "ते" -> "te".
3. **Apply Rules and Combine:** Combine the transliterated units two at a time iteratively and make sure that the border between the units is also joined as per the rules and examples provided. This step is highly error-prone; be extremely careful. For this example, na + ma + s + te = nama + ste = namaste.
4. **Verification:** The output after Step 3 is your candidate output. Now, verify it against **all** rules (1-6) one by one by breaking it down into constituent units. Create a mental checklist. Is the mapping correct? Is the case correct? Are clusters handled? Are nasals 'n' or 'm'? Are word endings correct? If you find any error, correct it and re-verify. For this example, splitting "namaste" into its components, we have "na", "ma", "s", and "te". Verifying the new components with the original mapping from step 2, we can see all components match the mapping. This transliteration is correct.
5. **Final Output:** After thorough verification, construct your final transliteration output.
6. **Confidence Level:** Determine your confidence in the output. If you are certain that the output is correct and follows all rules, mark it as "Certain". If you have any doubts or if there are edge cases that might not be covered by the rules, mark it as "Uncertain".

--continued--
```

**Box 2**: Formal Transliteration Prompt | Part B

```
## Output Format
The output should be a valid JSON object with the following structure

```json
{
    "step 1": <Follow the step 1 instructions>,
    "step 2": <Follow the step 2 instructions>,
    "step 3": <Follow the step 3 instructions>,
    "output": <The transliterated text in Latin script from step 3 copied identically>,
    "step 4": <Follow the step 4 instructions and verify the output against the rules very thoroughly>,
    "final_output": <The final output after verification>,
    "confidence": <"Certain" or "Uncertain">
}
```
```

**Box 3**: Few-Shot Example in the Formal Transliteration Prompt

```
**Few-Shot Example for Formal Transliteration**

*User Input:*
Input Hindi (Devanagri) Word: 'छुप'
Transliterated Sentence where word will be substituted: 'Xnaa manaa honaa chaahie'

*Assistant Output:*
```json

    "step 1": "Analyze and Decompose: The input word is 'छुप' which will be substituted into the sentence
    'Xnaa manaa honaa chaahie' post transliteration. Breaking it down: 'छ' + 'ु' + 'प'. 'छ' is the
    consonant 'chha' with the vowel sign 'ु' (short 'u'). 'प' is the consonant 'pa'. There are no
    consonant clusters or nasalization.",
    "step 2": "Initial Mapping match: 'छ' + 'ु' = 'chhu' (from 'छ' = 'chha', but with 'ु' it becomes
    'chhu'). 'प' = 'pa'.",
    "step 3": "Apply Rules and Combine: Combine the transliterated units: 'chu' + 'pa' = 'chupa'. There
    are no consonant clusters, so no schwa dropping is needed. No nasalization is present.",
    "output": "chhupa",
    "step 4": "Verification: Breaking down the final output `chhupa` into constituents: chhu + pa.
    - Rule 1 (Accuracy): 'chhu' is 'छ' with 'ु' , 'pa' is 'प'. Both match the mapping and combine into
    'छुप'.
    - Rule 2 (Case Sensitivity): All lowercase as per mapping.
    - Rule 3 (Consonant Clusters): No clusters present.
    - Rule 4 (Nasalization): No nasal present.
    - Rule 5 (Examples): Structure matches similar words in the examples.
    All rules are satisfied.",
    "final_output": "chhupa",
    "confidence": "Certain"

```
```

**Box 4**: Casual Transliteration Prompt

```
Your task is to transliterate Hindi text written in the Devanagari script into a casual, phonetic Latin
     letters.

The output should be easy to read for an English speaker and should reflect how a native Hindi speaker
     might type Hindi words using a standard English keyboard.

# Rules
1. **No Diacritics:** Do not use any special characters, accents, or diacritics (e.g., use 'a', not ā'').
2. **Phonetic Spelling:** Use common English letters that best represent the Hindi sound.

# Task
Now, transliterate the following Hindi text. Only provide the transliterated text as the output.
```

| Statistic | Value |
|---|---|
| Count | 1044 |
| Mean | 1.559 |
| Std | 0.76 |
| Min | 1.00 |
| 25% | 1.00 |
| 50% | 2.00 |
| 75% | 2.00 |
| Max | 5.00 |
| Consistency Rate (%) | 47.03 |

Table 12: Distribution of Unique Spellings over 5 frames per Casual Transliterated verb

| **Hindi verb** | **English verb** |
|---|---|
| देख् | witness |
| भगा | speed |
| भेद | pierce |
| महक | smell |
| खौला | boil |
| लुभा | seduce |
| पढ | read |
| पढ | study |
| चल् | move |
| खटखटा | knock |
| भाँप | gauge |
| घबड़ा | panic |
| सोच | imagine |
| पिरो | string |
| गिन | enumerate |
| छू | touch |
| बदल | change |
| ललकार | dare |
| बजा | ring |
| फाड़ | tear |
| तरस | yearn |
| छिड़क | spray |
| चूम | kiss |
| रोक | hinder |
| मिटा | rub |
| रो | weep |
| बना | create |
| ले | collect |
| फूट | crack |
| पीस | pulverise |

Table 13: Hindi–English verb pairs used in our experiments.

| **Model** | **HuggingFace ID** | **Parameters** |
|---|---|---|
| Gemma3-1b | `google/gemma-3-1b-pt` | 1B |
| Gemma3-4b | `google/gemma-3-4b-pt` | 4B |
| Gemma3-12b | `google/gemma-3-12b-pt` | 12B |
| Llama3.1-8B | `meta-llama/Llama-3.1-8B` | 8B |
| Llama3.1-70B | `meta-llama/Llama-3.1-70B` | 70B |
| Qwen3-32B | `Qwen/Qwen3-32B` | 32B |
| Sarvam-1 | `sarvamai/sarvam-1` | 2B |

Table 14: Overview of evaluated language models and their characteristics.

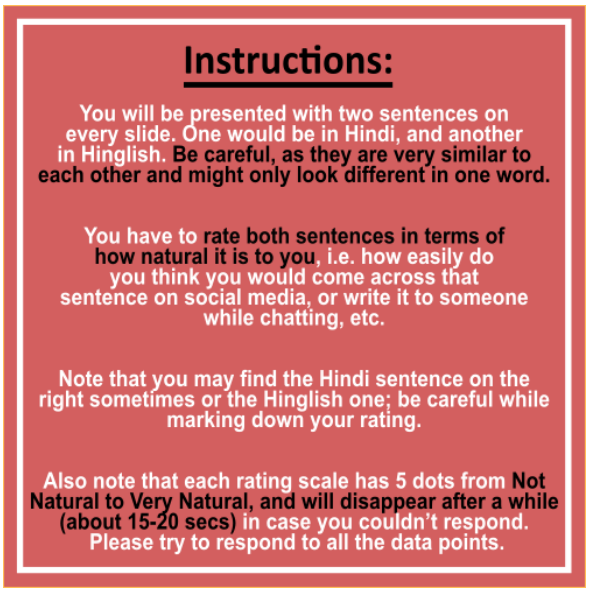



Figure 6: Instructions for human experiment.

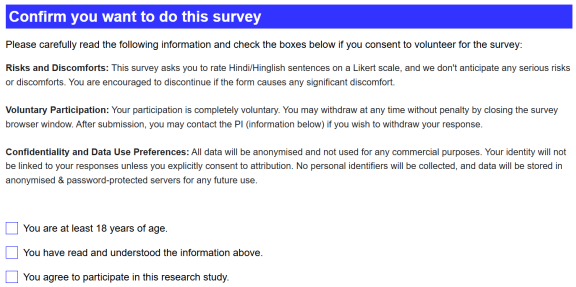



Figure 7: Consent for experiment.