Grammar Over Script: Cross-Lingual Transfer in Continual Pre-Training of Multilingual Large Language Models

Anonymous ACL submission

Abstract

Many global languages remain underrepresented in large language model (LLM) pre-training, yet curating high-quality corpora at scale for these languages is often prohibitively expensive. A common approach to adapting LLMs to low-resource languages is continual pre-training (CPT) on a related high-resource language, followed by supervised fine-tuning (SFT) on the target, making source-language choice critical. Yet the main driver of transfer, whether shared script or shared grammar, remains poorly understood. We disentangle these factors with a 2×2 factorial design that independently varies script overlap and grammatical overlap, using Mistral NeMo 12B, performing CPT on 2 billion words per language (standardised to control tokenizer fertility), and SFT on reasoning-focused data with no English natural-language text. Across Urdu and Persian targets, grammar-matched CPT consistently outperforms script-matched CPT. For Urdu, Hindi shares subject-object-verb grammar but zero of 131K subword tokens, yet beats Arabic, which shares script and 9,447 tokens, on every benchmark. Shared script without shared grammar also destabilises fine-tuning and can underperform an SFT-only baseline. We conclude that grammatical alignment is a stronger predictor of CPT transfer quality than script overlap and offers a practical criterion for choosing source languages in cross-lingual transfer training.

1 Introduction

Many of the world's diverse languages are underrepresented in LLM pre-training corpora, and curating new datasets for them at scale, often employing human annotators, is prohibitively expensive. *Cross-lingual transfer*, reusing linguistic knowledge from one language to improve another, offers one way forward. For low-resource languages (LRLs), where large, clean text and labelled data are scarce (Joshi et al., 2020), it allows practitioners to borrow competence from a related high-resource language (HRL) rather than building every resource from scratch. A common approach is *continual pre-training* (CPT): resuming pre-training on an HRL corpus, then supervised fine-tuning (SFT) on the target LRL (Gururangan et al., 2020; Ke et al., 2023). The premise is that structure learned during CPT, such as word-order regularities and morphological patterns, transfers to the LRL during SFT, letting the HRL act as a proxy source of supervision.

This makes the choice of HRL source a consequential design decision, yet it is poorly understood. As our comparison with an SFT-only baseline shows (Section 4.8), a poorly chosen source can score below a model fine-tuned without any CPT at all. Prior work shows that typological proximity between source and target predicts transfer quality (Pires et al., 2019; Lauscher et al., 2020), but the pairs studied typically share both script and grammar, so the two are confounded. The scarcity of LRL data compounds the problem: researchers often fall back on machine-translated benchmarks or synthetic data generation, which carry their own biases (Chua et al., 2026), and no existing guideline tells a practitioner whether to prioritise a script-matched or grammar-matched source when both are available. Two candidate mechanisms (surface-level transfer through shared subword tokens, and structural transfer through shared grammar) typically co-occur in the pairs studied, leaving open *which* drives transfer.

We adopt a 2×2 factorial design that isolates these two factors. For each target LRL, we select HRL sources that populate a factorial matrix crossing *script overlap* (present or absent) with *grammatical overlap* (present or absent), yielding four conditions: **Grammar only** (syntactic match, no shared tokens), **Script only** (shared tokens, no syntactic match), **Both** (expected ceiling), and **Neither** (negative control). This design yields

Grammar Overlap	Script Overlap: × (no script)	Script Overlap: ✓ (script)
✓	**Grammar only** Hindi → Urdu/Pers.	**Both** Persian → Urdu Urdu → Pers.
×	**Neither** Tamil → Urdu Russian → Pers.	**Script only** Arabic → Urdu/Pers.

Figure 1: The 2 × 2 factorial design. Each cell represents a CPT→SFT condition defined by the presence (✓) or absence (×) of script overlap and grammatical overlap between the CPT language and the target LRL. Example language pairs are shown for the Urdu and Persian targets.

two testable predictions. First (**H1**, *Grammar > Script*), the Grammar-only condition will outperform Script-only, even when the grammar-matched source shares zero subword tokens with the target. Second (**H2**, *More tokens ≠ better transfer*), because we hold word count fixed, CPT token count reflects only tokenizer fertility and should not predict downstream performance.

Two design choices defend against confounds. First, we standardise CPT volume at **2 billion words** per language rather than a fixed token count, because tokenizer fertility varies 2.7× across our scripts (Table 20). A fixed token budget would give high-fertility scripts substantially less linguistic content. Second, we use Mistral NeMo 12B, whose Tekken tokenizer was trained on over 100 languages and therefore offers balanced subword segmentation across scripts (Rust et al., 2021).

We note that Hindi and Urdu are closely related registers of Hindustani with shared colloquial grammar and cognate vocabulary, making the Urdu target a conservative test where transfer channels extend beyond abstract syntax. The Persian target, where Hindi is genealogically distant (Indo-Aryan vs. Iranian branch) and not mutually intelligible, provides stronger evidence that the effect extends beyond Hindustani proximity, and replicates the same ranking.

Key Contributions. We make three contributions. **First,** we propose a 2×2 factorial design that, to our knowledge, is the first to separate script and grammar in a CPT→SFT pipeline for autoregressive LLMs, replicated across two target languages with multi-seed evaluation. **Second**, we show that grammar-matched CPT with **zero** shared tokens outperforms script-matched CPT with thousands of shared tokens, and that the grammar main effect exceeds the script effect on both targets (Section 4.5). **Third**, we observe that script-matched CPT without grammar alignment produces unstable SFT recovery (Coefficient of Variation (CV) up to 30.1% on Belebele FA), while the grammar-only condition remains stable (CV ≤ 2.6% on Belebele FA), a pattern we term *script interference*. These findings indicate that grammatical alignment should be prioritised over script compatibility when selecting a CPT source.

2 Related Work

2.1 Multilingual Language Models

Recent decoder-only models use tokenizers trained on 100+ languages, partially mitigating the token fertility problem for non-Latin scripts, as in Mistral NeMo, Qwen2.5, and Llama 3 (Mistral AI, 2024; Qwen Team et al., 2024; Grattafiori et al., 2024).

2.2 Continual Pre-Training for Language Adaptation

CPT extends a model to new languages by resuming pre-training on targeted corpora (Gururangan et al., 2020; Ke et al., 2023; Ebrahimi and Kann, 2021; Muller et al., 2021). CPT risks catastrophic forgetting of the base model's existing capabilities (Kirkpatrick et al., 2017), a concern commonly mitigated by replaying a fraction of the original pre-training data during CPT (Scialom et al., 2022). Following Ibrahim et al. (2024) and Gupta et al. (2023), we include 25% English replay in all CPT runs.

2.3 Factors in Cross-Lingual Transfer

Prior work has investigated several factors that predict cross-lingual transfer quality.

Typological similarity. Pires et al. (2019) showed that mBERT's zero-shot transfer correlates with word-order similarity, while Lauscher et al. (2020) found that morphological similarity also predicts transfer quality. Lin et al. (2019) proposed selecting transfer languages based on typological features from WALS (Dryer and Haspelmath, 2013) and URIEL (Littell et al., 2017) databases. These studies show that typological proximity predicts transfer quality but do not isolate which property is operative, because the language pairs examined typically share both script and grammar.

Lexical and subword overlap. Conneau et al. (2020) showed that shared vocabulary is a strong predictor of XLM-R's cross-lingual performance, and K et al. (2020) identified anchor tokens (shared subwords) as critical for alignment. However, these studies examine zero-shot transfer from English, not CPT-based transfer between typologically varied language pairs. We extend this line of work by quantifying subword overlap in a controlled CPT transfer setting, showing that a source language with zero shared tokens outperforms one with thousands.

Multilingual data mixtures. Recent work has studied how *quantity* and *proportion* of multilingual data affect pre-training from scratch (Foroutan et al., 2025; Conneau et al., 2020), and Li et al. (2025) examined data-mixing strategies for multilingual continual pre-training across resource levels. These studies focus on data quantity and mixing. Our work instead targets source-language identity, investigating *which* HRL to select for CPT given a specific target LRL.

Script effects and interference. Languages sharing a script produce overlapping subword vocabularies, which can aid surface-level transfer. Khemchandani et al. (2021) showed that transliterating Indic languages to a common script improves cross-lingual transfer for encoder models, confirming the role of token overlap, and Fujinuma et al. (2022) found that script match is a strong predictor of cross-lingual transferability in multilingual pre-trained models. Tufa et al. (2024) recently studied the impact of unknown scripts on transfer, finding that script unfamiliarity degrades performance even when typological similarity is high. However, script sharing does not imply grammatical similarity: Arabic shares Perso-Arabic script with Urdu yet differs in canonical word order (VSO/SVO vs. SOV) and morphological type. Xhelili et al. (2024) showed that transliteration-based post-training alignment can partially bridge script barriers, suggesting that script effects operate at the surface level and can be circumvented. Wang et al. (2020) identified negative interference in multilingual models, where training on one language degrades performance on another. The script interference pattern we observe (Section 5.2) may reflect a similar dynamic at the CPT stage. Deshpande et al. (2022) used constructed counterfactual languages to isolate the contributions of word order, script, and other typological features to cross-lingual transfer in BERT. Their work is the closest methodological precedent for factorial isolation of linguistic variables. To our knowledge, no prior work has explicitly controlled for script overlap while varying grammatical similarity in a CPT→SFT transfer setting for autoregressive LLMs. Prior encoder-model studies reach mixed conclusions: Pires et al. (2019), Lauscher et al. (2020), and Deshpande et al. (2022) find typological similarity (especially word order) to be the dominant predictor, while Fujinuma et al. (2022) find script match to be a strong predictor. Our results align with the former group, extending the grammar-over-script finding to the CPT→SFT setting for autoregressive LLMs.

3 Methodology

3.1 Experimental Design

We cross two binary factors: *script overlap* (shared writing system) and *grammatical overlap*, operationalised as shared canonical word order (SOV/SVO/VSO) and broad morphological type within the same macro-family. This binary coding enables clean factorial contrasts. Continuous alternatives are discussed in the Limitations section.

Table 6 shows the complete design for Urdu (4 conditions) and Persian (5 conditions). Urdu appears as both target and source, creating a symmetry test: if grammar transfers from Hindi→Urdu, it should also operate from Urdu→Persian.

We follow this pipeline: CPT on an HRL corpus (with 25% English replay) → SFT on the target LRL (zero English) → evaluation. We use Mistral-NeMo-Instruct-2407 (12B) (Mistral AI, 2024) with the Tekken tokenizer (100+ languages), whose broad coverage avoids confounding script overlap with tokenization quality.

3.2 Data Preparation

3.2.1 CPT Data

For each HRL, we sample monolingual text from FineWeb2 (Penedo et al., 2025) and SEA-PILE-v2 (Ng et al., 2025), standardising at **2 billion words** per language rather than a fixed token budget, because tokenizer fertility varies 2.7× across our scripts (Table 20). A fixed token budget would give high-fertility languages substantially less linguistic content (Rust et al., 2021) (see Table 21 for data source details).

To mitigate catastrophic forgetting (Kirkpatrick et al., 2017), we add replay data equal to 25% of the CPT word count to each CPT corpus.

3.2.2 SFT Data

SFT datasets are constructed for each target LRL with no English natural-language text (verified by checking character scripts, with formatting tokens retained), emphasising chain-of-thought reasoning rather than factual recall. For Urdu, we use a cleaned subset of the Urdu-Instruct dataset (Shafique et al., 2025). For Persian, we use a cleaned Maux-Persian-SFT-30k dataset (Team, 2025) (See Appendix F for more details). Within each target all conditions share the same SFT data, so cross-condition comparisons are unaffected, but cross-target effect magnitudes are not directly comparable (see Table 21).

3.3 Quantifying Subword Token Overlap

We quantify script overlap by classifying each of the Tekken tokenizer's 131,072 vocabulary entries by its Unicode script block. Since each writing system occupies a disjoint Unicode range, tokens belonging to different scripts never overlap.

For example, Urdu and Arabic both belong to the Perso-Arabic script family (U+0600–U+06FF and extended blocks), sharing all 9,447 Arabic-script tokens in the vocabulary. Hindi, written in Devanagari (U+0900–U+097F), shares **zero tokens** with either. Any Hindi→Urdu advantage over Arabic must therefore reflect deeper representational transfer, not surface token matching (Appendix B).

3.4 Identifying Grammar Overlap

We treat *grammatical overlap* as a binary variable based on *canonical word order*, defined as the dominant order of subject (S), object (O), and verb (V) in declarative main clauses. Throughout this paper, "grammar" refers specifically to this operationalisation (word order plus broad morphological type), not to the full linguistic sense of the term. Hindi, Urdu, and Persian are all SOV and therefore grammar-overlapping; Arabic (VSO/SVO) is grammar-mismatched with all three despite sharing Perso-Arabic script. This coding is deliberately coarse, capturing the broadest syntactic property while remaining unambiguous for factorial contrasts. Finer-grained features (case systems, morphological typology, pro-drop) also contribute to grammatical similarity, which informs our classification of Tamil as grammar-mismatched despite its shared SOV order.

3.5 Training Configuration

3.5.1 CPT

All CPT runs use LoRA (Hu et al., 2022) with rank 256 on 8× H100 GPUs, with batch size 4 per device, sequence length 4,096, and OneCycleLR scheduling (peak learning rate 6×10^{-5}, 5% warmup, weight decay 0.1).

The model, LoRA configuration, optimiser, scheduler, and 25% English replay ratio are identical across all six CPT runs; only the source-language corpus differs. Because word-count standardisation induces different token counts across scripts, the number of optimisation steps varies by source language. (See Appendix Tables 12–17 for per-language training configurations).

3.5.2 SFT

SFT was conducted on the CPT-merged model with identical learning-rate and scheduling settings across both target languages (see Appendix Tables 18–19 for training configurations).

3.6 Evaluation

We evaluate on Belebele (Bandarkar et al., 2024) (reading comprehension; 900 Q&A pairs per language) for both targets. For Urdu, we use UrduBench, a human-verified translated benchmark comprising MGSM (Shi et al., 2023), Math500, and OpenBookQA, covering mathematical, scientific, and commonsense reasoning under low-resource settings. For Persian, we evaluate on ParsiNLU (Khashabi et al., 2021) and Persian MuSR (ParsBench, 2024), which assess natural language understanding and multi-step reasoning.

All SFT experiments are run four times per condition using seeds 0, 42, 123, and 456, with the CPT checkpoint held fixed. We report mean ± standard deviation as well as coefficient of variation (CV%; see Appendix G for full dataset details).

4 Results

All CPT models reported below use a single CPT seed (seed 0); stochastic variation is isolated at the SFT stage via four independent seeds per condition. Inferences about CPT-level effects are therefore conditional on this single initialisation, a caveat we revisit in the Limitations section.

4.1 Urdu Target Matrix

H1 supported: grammar beats script. Hindi CPT (grammar match, no script overlap) ranks

Table 1: Urdu target results (mean ± std over 4 SFT seeds). Bold indicates best per column.

Cond.	Src.	Belebele	Math500	MGSM	OBQA
Gram. only	Hindi	**63.41**±1.26	**12.70**±0.53	**44.00**±1.63	**48.15**±1.95
Script only	Arabic	60.41±1.35	11.05±0.82	40.00±0.86	41.50±2.22
Both	Persian	58.70±1.37	10.95±1.02	40.80±1.42	44.00±2.28
Neither	Tamil	56.56±0.84	9.40±1.14	34.00±2.59	36.20±1.26

first on every benchmark (Table 1). The advantage over Arabic (script match, grammar mismatch) is statistically significant on Belebele (+3.0 percentage points (pp), $p = 0.027$, $d = 2.30$), MGSM (+4.0 pp, $p = 0.028$, $d = 3.07$), and OpenBookQA (+6.7 pp, $p = 0.028$, $d = 3.18$) by paired t-tests across seeds; Math500 shows the same direction (+1.7 pp, $d = 2.39$) but does not reach significance. All p-values are uncorrected; after Holm–Bonferroni correction across the ten pairwise tests reported in this paper, these three Urdu comparisons ($p \approx 0.027$–0.028) do not individually survive, though the consistent direction across all four benchmarks and large effect sizes ($d > 2.3$) support the same conclusion.

The *Both* condition underperforms *Grammar only*. Persian CPT (both script and grammar overlap) was expected to be the performance ceiling. Instead, it scores below Hindi: 58.7% vs. 63.4% on Belebele ($p = 0.023$, $d = 3.58$). Persian and Arabic perform similarly on MGSM (40.8% vs. 40.0%), suggesting that the additional grammar alignment of Persian is partially offset by surface-level script shortcuts (discussed in Section 5.3).

Negative control ranks last. The gap between Hindi and Tamil is significant on MGSM (+10.0 pp, $p = 0.003$, $d = 4.62$), OpenBookQA (+11.9 pp, $p = 0.004$, $d = 7.28$), and Belebele (+6.9 pp, $p = 0.005$, $d = 6.40$), consistent with genuine linguistic transfer rather than an artifact of additional pre-training.

Cross-seed stability. All four Urdu-target conditions show low coefficient of variation (max CV $\leq 12.2\%$; Table 10), with grammar-matched conditions being the most stable (Hindi CV $\leq 4.2\%$).

4.2 Persian Target Matrix

Table 2 presents results for the five Persian-target conditions, adding Urdu as "Both" and Russian as "Neither" alongside the shared Hindi/Arabic/Tamil sources.

Grammar dominates reasoning. A source language that shares the target's grammar outperforms one that shares its script on every reasoning benchmark, even with zero overlapping subword tokens. Hindi (grammar only, zero shared tokens) ranks first on both reasoning benchmarks: ParsiNLU (36.64%) and MuSR (51.35%). The advantage over Arabic (script only) is significant on MuSR (+14.9 pp, $p < 0.001$, $d = 12.06$) and over Tamil on both ParsiNLU (+21.3 pp, $p = 0.002$, $d = 7.27$) and MuSR (+6.2 pp, $p = 0.005$, $d = 5.16$).

Urdu leads on reading comprehension. For surface-level reading comprehension, having both grammar and script overlap provides a small edge. On Belebele FA (Persian), Urdu (both) scores 74.53% vs. Hindi's 73.90%, a gap of only 0.63 pp, well within noise. Urdu achieves the lowest variance across seeds (std = 0.29), fitting the expectation that the "Both" condition provides the most stable surface-level transfer for a reading comprehension task.

Script interference on the Persian target. Script-matched CPT without grammatical alignment produces erratic downstream performance across random seeds, making it an unreliable basis for source selection. Arabic (script only) is highly unstable, with Belebele FA accuracy ranging from 32.8% to 69.0% across four seeds (CV = 30.1%). Russian (neither) is even more unstable (CV = 51.4%), with one seed producing near-random performance (seed 456: 14.3% Belebele, 16.4% ParsiNLU, 19.7% MuSR) despite a normal SFT training loss (0.811, within 0.05 of other seeds), indicating that the failure is downstream task recovery rather than training divergence. In contrast, grammar-matched conditions are highly stable: Hindi CV $\leq 2.6\%$, Urdu CV $\leq 10.9\%$ (Table 10).

Tamil provides a second baseline. Tamil (second "Neither" control) performs inconsistently across benchmarks: strong on MuSR (45.20%, second only to Hindi) but weakest on ParsiNLU (15.34%). We do not have an account of this divergence, which shows the value of evaluating on multiple tasks instead of only a single metric.

Table 2: Persian target results (mean ± std over 4 SFT seeds). Bold indicates best per column.

Cond.	Src.	Belebele	ParsiNLU	MuSR
Gram. only	Hindi	73.90±1.90	**36.64**±0.65	**51.35**±0.56
Both	Urdu	**74.53**±0.29	35.94±1.22	41.70±4.53
Script only	Arabic	55.35±16.66	35.92±0.66	36.41±1.66
Neither	Russian	59.85±30.78	26.58±7.10	36.01±14.01
Neither	Tamil	69.08±5.01	15.34±4.09	45.20±1.59

4.3 Token Volume vs. Performance

Downstream performance is uncorrelated with CPT token count. The observed Urdu composite ranking (Hindi 42.1% > Persian 38.6% > Arabic 38.2% > Tamil 34.0%, a coarse composite dominated by higher-baseline tasks, see Table 7) bears no consistent relation to the token-count ordering (Tamil 8.1B > Hindi 4.6B > Arabic 4.4B > Persian 4.1B): Tamil has nearly double Hindi's tokens yet scores 8.0 pp lower. Additional tokens do not improve transfer, consistent with **H2** (More tokens $\neq$ better transfer) and with our word-count standardisation, which holds linguistic content fixed so that token count reflects only fertility.

4.4 Continuous Typological Distance

URIEL syntactic distances (Littell et al., 2017) place Hindi–Urdu at 0.049, 5–8× smaller than all other pairs, which accords with Hindi's dominance (Appendix D). However, Spearman rank correlations are non-significant ($\rho \approx -0.40$, $p > 0.49$), largely because URIEL syntax vectors underweight the morphological divergence that drives Tamil's poor performance despite its low syntactic distance.

4.5 Factorial Main Effects

Decomposing the factorial design confirms grammar as the larger effect on reasoning, while script overlap contributes little on its own and is slightly harmful when grammar is already present. Table 9 reports the factorial decomposition for one representative reasoning benchmark per target.

The grammar effect is larger than the script effect on reasoning in both targets (∼+5 pp), with script particularly weaker on Urdu (+1.40 pp vs. +4.32 pp on Persian). The interaction term is large and negative (-10.0 pp Persian, -9.2 pp Urdu): grammar's advantage concentrates in the *absence* of script overlap ($+10.1$ pp without script vs. $+0.0$ pp with script on Persian), while script overlap is slightly *harmful* when grammar is already present (-0.7 pp). This sub-additivity is compatible with the *forced-abstraction* hypothesis discussed in Section 5.3, though other explanations are equally consistent. Table 9 uses Russian to populate the Persian Neither cell. If Tamil is used instead, both effects increase (grammar $+10.7$, script $+9.9$ pp) because Tamil's ParsiNLU score is lower (15.3% vs. 26.6%). Averaging the two Neither conditions yields grammar $+7.9$, script $+7.1$ pp. The grammar-over-script ordering is preserved under all three choices. After Holm–Bonferroni correction across the ten pairwise tests reported in Section 4, the Persian-target and Hindi-vs-Tamil comparisons remain significant ($p \leq 0.009$), while the Urdu Hindi-vs-Arabic comparisons ($p \approx 0.027$–0.028) do not individually survive correction.

4.6 Multi-Seed Stability

Grammar-matched CPT sources produce more reliable SFT outcomes: variance across random seeds tracks the 2×2 conditions (Table 10), with grammar-mismatched sources showing the greatest instability. On the Persian target, Arabic CV = 30.1% and Russian CV = 51.4%, while grammar-matched conditions remain stable (Hindi CV $\leq$ 2.6%). The Urdu target shows a milder but consistent gradient, with grammar-mismatched conditions again less stable than grammar-matched ones.

4.7 Cross-Target Replication

All five key patterns replicate across both targets: Hindi ranks first on reasoning, script-only underperforms grammar ($p < 0.001$ on Persian MuSR, $p < 0.05$ on Urdu MGSM/OBQA), grammar effect exceeds script effect (Table 9), and grammar-matched conditions are most stable (CV $\leq$ 4.2%). We note that both targets share the same CPT checkpoints (Hindi, Arabic, Tamil), so this replication demonstrates consistent downstream transfer from the same CPT representations rather than an independent replication of the CPT-level effect itself.

4.8 Reference Point: SFT Without CPT

This comparison provides a reference point for target-language SFT without source-language CPT

(Tables 3–4). We fine-tune Mistral-NeMo-Base-2407 with identical SFT hyperparameters and the same four seeds. This baseline starts from the base model, whereas all CPT conditions start from the instruct checkpoint. The cross-condition comparisons that carry our thesis are unaffected, since all CPT conditions share the same instruct checkpoint. Script-only and neither conditions score below even this baseline, suggesting that grammar-mismatched CPT degrades performance enough to offset any benefit from the instruct initialisation.

CPT vs. SFT-only baseline. Hindi CPT improves over the baseline by up to $+13.9$ pp (Persian MuSR) and $+7.5$ pp (Urdu MGSM). Only Hindi CPT consistently exceeds the baseline across all benchmarks; script-only and neither conditions fall below it on most.

5 Discussion

5.1 Grammar as the Stronger Predictor

The grammar main effect exceeds the script effect on both targets (Table 9), supporting **H1** (*Grammar > Script*): during CPT the model appears to learn syntactic processing patterns (word-order expectations, case-role assignments, morphological agreement) that it can repurpose during SFT, even when surface tokens are entirely different. Grammar-matched sources in our design are also genealogically closer to the targets, so the grammar main effect may partly reflect broader typological proximity rather than grammar alone. Our design separates grammar-correlated properties from script overlap, but cannot further decompose grammar from genealogical relatedness (see the Limitations section). For the practical question of source-language selection, this distinction is immaterial: typologically close sources outperform script-matched sources regardless of which underlying property is operative.

5.2 Script Interference

We use the descriptive label *script interference* for the pattern in which shared script without grammatical alignment destabilises and degrades transfer. The label names an empirical regularity, not a confirmed causal mechanism. The instability is observed across all four SFT seeds, though because all CPT runs use a single seed the CPT-level contribution is not directly isolated. On the Urdu target, Arabic (9,447 shared tokens, VSO) is outperformed by Hindi (**zero** shared tokens, SOV) across all four benchmarks. On the Persian target, Arabic CPT appears to disrupt shared Perso-Arabic representations: Belebele FA swings from 32.8% to 69.0% across seeds (CV = 30.1%), compared to Hindi's stable 71.1–75.3% (CV = 2.6%). The instability is consistent with representational interference, where Arabic-grammar expectations (VSO) may overwrite the token encodings that Persian (SOV) relies on during SFT. *Both* and *Grammar* conditions are substantially more stable than *Script* and *Neither*, so **grammar-matched sources should be preferred for both accuracy and reliability**.

Script interference accounts for the failure of mismatched-grammar sources, but it does not explain why Hindi outperforms the expected ceiling within the grammar-matched conditions.

5.3 Possible Reasons for Hindi Outperforming the Expected Ceiling

Morphological depth. Hindi and Urdu share SOV order as well as detailed morphological paradigms: identical case systems (*ne, ko, se, mein*), gender×number agreement, compound verbs, and an ergative split in the perfective. Persian (Iranian branch) lacks grammatical gender, uses ezafe constructions, and has different verbal morphology; these differences are invisible to our binary classification but captured by continuous URIEL distances (Littell et al., 2017).

Cognate vocabulary. Despite zero shared tokens, Hindi and Urdu share extensive cognate vocabulary (Perso-Arabic loanwords), creating parallel distributional contexts during CPT that transfer even without surface token matching.

***Forced abstraction* (hypothesized).** A third, more speculative possibility is that zero token overlap forces the model toward deeper syntactic representations. Because our LoRA adapts only attention and MLP projections while leaving input embeddings frozen, CPT cannot install new surface-token representations, so any transferred information must be encoded in the structural layers. SFT convergence curves are consistent with this account: Hindi (0.805 ± 0.001 at step 100) converges faster than Arabic (0.867 ± 0.002) or Persian (0.871 ± 0.003) despite sharing zero tokens (Table 11). However, faster convergence is equally consistent with Hindi simply being a closer language along every axis. On the Persian target, the pattern does not replicate: Hindi, Arabic, and Urdu converge at nearly identical rates, while Russian achieves the lowest SFT loss yet produces the

Table 3: Reference comparison: SFT-only baseline vs. CPT→SFT conditions for the Persian target. Values are mean ± std over four seeds. Δ is computed against the baseline. Bold indicates the best CPT→SFT condition per benchmark.

	Condition	Belebele FA	ParsiNLU	Persian MuSR
Baseline	Base SFT-Only	71.58 ± 3.54	36.59 ± 0.90	37.47 ± 11.84
CPT→SFT	Hindi → Persian (Grammar)	73.90 ± 1.90 (+2.3)	$\mathbf{36.64} \pm 0.65$ (+0.05)	$\mathbf{51.35} \pm 0.56$ (+13.9)
	Urdu → Persian (Both)	$\mathbf{74.53} \pm 0.29$ (+2.9)	35.94 ± 1.22 (−0.7)	41.70 ± 4.53 (+4.2)
	Arabic → Persian (Script)	55.35 ± 16.66 (−16.2)	35.92 ± 0.66 (−0.7)	36.41 ± 1.66 (−1.1)
	Russian → Persian (Neither)	59.85 ± 30.78 (−11.7)	26.58 ± 7.10 (−10.0)	36.01 ± 14.01 (−1.5)
	Tamil → Persian (Neither)	69.08 ± 5.01 (−2.5)	15.34 ± 4.09 (−21.3)	45.20 ± 1.59 (+7.7)

Table 4: Reference comparison: SFT-only baseline vs. CPT→SFT conditions for the Urdu target. Values are mean ± std over four seeds. Δ is computed against the baseline. Bold indicates the best CPT→SFT condition per benchmark. Belebele uses the standard multilingual split; Math500, MGSM, and OpenBookQA are Urdu translations from Shafique et al. (2026).

	Condition	Belebele UR	Math500	MGSM	OBQA
Baseline	Base SFT-only	62.81 ± 0.21	9.25 ± 0.53	36.50 ± 3.46	42.25 ± 4.01
CPT→SFT	Hindi → Urdu (Grammar)	$\mathbf{63.41} \pm 1.26$ (+0.6)	$\mathbf{12.70} \pm 0.53$ (+3.5)	$\mathbf{44.00} \pm 1.63$ (+7.5)	$\mathbf{48.15} \pm 1.95$ (+5.9)
	Arabic → Urdu (Script)	60.41 ± 1.35 (-2.4)	11.05 ± 0.82 (+1.8)	40.00 ± 0.86 (+3.5)	41.50 ± 2.22 (-0.8)
	Persian → Urdu (Both)	58.70 ± 1.37 (-4.1)	10.95 ± 1.02 (+1.7)	40.80 ± 1.42 (+4.3)	44.00 ± 2.28 (+1.8)
	Tamil → Urdu (Neither)	56.56 ± 0.84 (-6.3)	9.40 ± 1.14 (+0.2)	34.00 ± 2.59 (-2.5)	36.20 ± 1.26 (-6.1)

worst downstream performance, confirming that SFT loss alone does not predict task transfer. The hypothesis lacks a distinguishing testable prediction within the current design, and we flag it as a direction for future work.

The Hindustani register confound. We acknowledge that Hindi–Urdu proximity means the "Grammar only" condition on the Urdu target transfers more than abstract syntax: shared cognate vocabulary and near-identical colloquial grammar likely ease the remapping between scripts. However, the Persian target directly addresses this concern. Hindi and Persian are not mutually intelligible, share no colloquial register, and belong to different branches of Indo-Iranian (Indo-Aryan vs. Iranian), yet Hindi CPT still outperforms Arabic CPT on MuSR by +14.9 pp ($p < 0.001$) and ranks first on ParsiNLU. This replication on a typologically distant target indicates that the transfer mechanism extends beyond Hindustani register remapping.

5.4 Practical Implications for Source Selection

Grammatical structure (shared word order and morphological type) predicted transfer quality better than script overlap when selecting an HRL proxy for CPT. Coarse word-order matching alone is not enough: Hindi outperforms Persian on the Urdu target because the two share detailed morphological paradigms beyond SOV order (Table 7). Genetic (family) distance also tracks downstream performance well enough to serve as a practical proxy when typological databases are incomplete (Hindi 0.000 > Persian 0.691 > Arabic/Tamil 1.000). Appendix E provides a step-by-step selection guide. Script overlap alone did not predict transfer quality. When grammar is already aligned it can help, but a grammar-matched source in a different script outperforms a script-matched source with different grammar on both targets.

6 Conclusion

We presented a 2 × 2 factorial design that separates script and grammar as predictors of cross-lingual CPT transfer quality. Across six CPT configurations (2B words each) and thirty-six SFT runs on Urdu and Persian targets, Hindi CPT with **zero** shared tokens outperformed Arabic CPT with 9,447 shared tokens on every reasoning benchmark. Against an SFT-only baseline, grammar-matched CPT improves performance, while script-only and neither conditions score below it and show unstable SFT recovery (CV up to 30.1%). By showing that grammatical alignment is a stronger predictor of CPT transfer quality than script overlap, this work provides an empirically grounded practical guide (Appendix E) to optimise source-language selection and compute allocation for adapting multilingual LLMs to low-resource languages.

Limitations

Statistical power with four seeds. Four SFT seeds per condition enable paired t-tests but yield limited power for small effects. Several comparisons are marginally non-significant ($p \approx 0.05$–0.10) despite large observed differences, likely due to high-variance conditions inflating the standard error.

Coarse typological classification. Binary grammar overlap (word order + morphological type) enables clean 2×2 contrasts but misses finer-grained properties (case systems, agreement, pro-drop). The Hindi>Persian result on the Urdu target is better explained by a continuous gradient than a binary split. Tamil's classification is a borderline case: it shares SOV order with Urdu but differs in morphological type (agglutinative vs. fusional), case marking, and gender agreement. Under the alternative coding (partial grammar overlap), Tamil's last-place ranking would *strengthen* our claim. The core comparison, Hindi outperforming Arabic on every benchmark for both targets, involves only the Grammar-only and Script-only cells and holds regardless of how Tamil is classified.

Hindi–Urdu closeness and genealogical proximity. Hindi and Urdu are registers of Hindustani; the "Grammar only" label likely understates the transfer channels available on the Urdu target. While the tokenizer produces completely disjoint token sequences (0/131K overlap) and our CPT data is *written* web text where script and formal vocabulary divergence is maximal, shared colloquial grammar and cognate vocabulary ease the cross-script remapping. More broadly, grammar-matched sources in our design are also genealogically closer to the targets, raising the possibility that genetic relatedness, not grammar per se, drives transfer. The Persian target offers a partial test: Hindi and Urdu share identical URIEL genetic distance to Persian (0.691), yet Hindi leads on MuSR (+9.7 pp, $p = 0.019$) while the two are statistically indistinguishable on Belebele and ParsiNLU ($p > 0.35$). The near-parity on two of three benchmarks is what a genealogical account would predict, since both sources are equidistant from Persian. The MuSR divergence may reflect finer-grained typological differences or evaluation artifacts (higher parse-failure rates in the Urdu MuSR condition). These results do not rule out genealogical proximity as a partial driver of transfer, but they do confirm that script overlap is the weaker predictor regardless of which underlying mechanism is operative.

Benchmark validity. All benchmarks are multiple-choice; a dedicated syntactic evaluation (acceptability judgments, parsing) or generative tasks (translation, summarisation) would more directly measure the transfer we hypothesise.

Single CPT seed. All CPT checkpoints use a single seed (seed 0), so we cannot fully separate condition effects from CPT-run variance. Three properties of the experimental design narrow this exposure. First, the four SFT seeds per condition capture downstream variance from each CPT checkpoint, so the open question is whether a different CPT initialisation would change the *ranking* of conditions rather than just shifting absolute scores. Second, Hindi ranks first on six of seven evaluations, a consistency that a per-benchmark random-rank null makes unlikely even under conservative assumptions. Third, the same ranking replicates across two independent targets with different SFT corpora and different benchmarks, and the SFT-only comparison (Section 4.8) shows grammar-matched CPT above the reference point while mismatched CPT falls below it. The consistency of effect direction across targets, benchmarks, and seeds is itself evidence against a single-seed artifact, though we cannot rule one out without additional CPT seeds.

Language-level replication. Each cell of the factorial is populated by a single source language, so the design cannot fully separate language-specific properties from condition-level effects: the Grammar-only advantage could partly reflect Hindi-specific characteristics rather than grammar overlap in general. Populating each cell with multiple languages would require additional CPT runs at ∼130 GPU-hours each, and the set of high-resource languages that cleanly satisfy each cell's constraints (e.g., SOV word order, non-Perso-Arabic script, and a 2B-word corpus) is small. Cross-target replication and the convergence of multiple lines of evidence partially mitigate this concern, but a definitive test would require additional grammar-only sources such as Turkish for a Turkic target.

SFT composition and hyperparameters. Our SFT corpora contain no English natural-language text, a design choice that may favour grammar-matched transfer by removing a shared-language fallback. SFT datasets that include English could change the relative advantage of grammar-matched vs. script-matched sources. The Urdu and Persian

SFT runs also differ in effective batch size (4 vs. 16), so cross-target comparisons of effect magnitude should be interpreted with caution, though within-target condition rankings are unaffected.

CPT domain distribution. Hindi web text may be topically closer to Urdu (both South Asian) than Arabic or Tamil web text, and FineWeb2 applies no cross-lingual domain balancing. Tamil additionally draws from SEA-PILE-v2 rather than FineWeb2, introducing a possible corpus-quality confound for the negative control. Because we standardise by word count, high-fertility languages also receive more gradient updates during CPT (Tamil $\sim$62K steps vs. Hindi $\sim$35K). However, Tamil performs worst despite receiving the most training steps, so this confound works against the null hypothesis rather than explaining our results. The Persian target, whose web ecosystem is culturally distinct from South Asia, provides a partial control.

Single architecture and adaptation method. All experiments use Mistral NeMo 12B with LoRA, so results may not generalise to other models, tokenizers, or full-parameter training. The LoRA rank differs between CPT (rank 256) and SFT (rank 64), and whether this asymmetry interacts with the grammar-vs-script effect is untested. CPT starts from the instruct checkpoint rather than the base model; grammar-matched sources may better preserve the instruct model's existing instruction-following patterns, which could partly favour grammar-aligned conditions. However, script-matched sources also start from the same instruct checkpoint and additionally benefit from frozen embeddings that preserve pre-existing Perso-Arabic token representations, yet still underperform grammar-matched sources, suggesting the grammar advantage is not solely an artifact of instruct preservation.

Ethical Considerations

This work aims to improve language technology for low-resource languages, which are often spoken by communities underserved by current LLMs. All CPT and SFT data are drawn from publicly available corpora (FineWeb2, SEA-PILE-v2, and open-source instruction datasets). No human subjects were involved and no personally identifiable information was collected or used. Total compute was approximately 800 H100 GPU-hours. We will release reproducibility materials, including scripts and documentation, subject to licensing and privacy constraints. We used Claude (Anthropic) for language editing and proofreading of this manuscript. We note that our findings, while intended to guide practitioners toward more effective source-language selection, could also be used to deprioritise investment in script-matched resources; we encourage researchers to treat our results as one input among many when making resource allocation decisions for underserved language communities.

References

L. Bandarkar, D. Liang, B. Muller, M. Artetxe, S. N. Shukla, D. Husa, N. Goyal, A. Krishnan, L. Zettlemoyer, and M. Khabsa. 2024. The Belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775, Bangkok, Thailand. Association for Computational Linguistics.

Gabriel Chua, Leanne Tan, Ziyu Ge, and Roy Ka-Wei Lee. 2026. Lost in localization: Building RabakBench with human-in-the-loop validation to measure multilingual safety gaps. In *AAAI 2026 Workshop on Language Models for Underserved Communities (LM4UC)*.

A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.

Ameet Deshpande, Partha Talukdar, and Karthik Narasimhan. 2022. When is BERT multilingual? Isolating crucial ingredients for cross-lingual transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3610–3623. Association for Computational Linguistics.

M. S. Dryer and M. Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology.

A. Ebrahimi and K. Kann. 2021. How to adapt your pretrained multilingual model to 1600 languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4555–4567. Association for Computational Linguistics.

N. Foroutan, P. Teiletche, A. K. Tarun, and A. Bosselut. 2025. Revisiting multilingual data mixtures in language model pretraining. *arXiv preprint arXiv:2510.25947*.

Yoshinari Fujinuma, Jordan Boyd-Graber, and Katharina Kann. 2022. Match the script, adapt if multilingual: Analyzing the effect of multilingual pretraining on cross-lingual transferability. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1500–1512, Dublin, Ireland. Association for Computational Linguistics.

A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

K. Gupta, B. Thérien, A. Ibrahim, M. L. Richter, Q. Anthony, E. Belilovsky, I. Rish, and T. Lesort. 2023. Continual pre-training of large language models: How to (re)warm your model? In *Workshop on Efficient Systems for Foundation Models (ES-FoMo) at ICML 2023*.

S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360. Association for Computational Linguistics.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

A. Ibrahim, B. Thérien, K. Gupta, M. L. Richter, Q. Anthony, T. Lesort, E. Belilovsky, and I. Rish. 2024. Simple and scalable strategies to continually pre-train large language models. *Transactions on Machine Learning Research*.

P. Joshi, S. Santy, A. Budhiraja, K. Bali, and M. 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. Association for Computational Linguistics.

Karthikeyan K, Z. Wang, S. Mayhew, and D. Roth. 2020. Cross-lingual ability of multilingual BERT: An empirical study. In *International Conference on Learning Representations (ICLR)*.

Z. Ke, Y. Shao, H. Lin, T. Konishi, G. Kim, and B. Liu. 2023. Continual pre-training of language models. In *International Conference on Learning Representations (ICLR)*.

Daniel Khashabi, Arman Cohan, Siamak Shakeri, Pedram Hosseini, Pouya Pezeshkpour, Malihe Alikhani, Moin Aminnaseri, Marzieh Bitaab, Faeze Brahman, Sarik Ghazarian, Mozhdeh Gheini, Arman Kabiri, Rabeeh Karimi Mahabadi, Omid Memarrast, Ahmadreza Mosallanezhad, Erfan Noury, Shahab Raji, Mohammad Sadegh Rasooli, Sepideh Sadeghi, and 6 others. 2021. ParsiNLU: a suite of language understanding challenges for persian. *Transactions of the Association for Computational Linguistics*, 9:1147–1162.

Yash Khemchandani, Sarvesh Mehtani, Vaidehi Patil, Abhijeet Awasthi, Partha Talukdar, and Sunita Sarawagi. 2021. Exploiting language relatedness for low web-resource language model adaptation: An Indic languages study. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1312–1323. Association for Computational Linguistics.

J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

A. Lauscher, V. Ravishankar, I. Vulić, and G. Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499. Association for Computational Linguistics.

Zihao Li, Shaoxiong Ji, Hengyu Luo, and Jörg Tiedemann. 2025. Rethinking multilingual continual pretraining: Data mixing for adapting LLMs across languages and resources. In *Conference on Language Modeling (COLM)*.

Y.-H. Lin, C.-Y. Chen, J. Lee, Z. Li, Y. Zhang, M. Xia, S. Rijhwani, J. He, Z. Zhang, X. Ma, A. Anastasopoulos, P. Littell, and G. Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

P. Littell, D. R. Mortensen, K. Lin, K. Kairis, C. Turner, and L. Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Mistral AI. 2024. Mistral NeMo. Blog post.

B. Muller, Y. Elazar, B. Sagot, and D. Seddah. 2021. First align, then predict: Understanding the cross-lingual ability of multilingual BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2214–2231. Association for Computational Linguistics.

Raymond Ng, Thanh Ngan Nguyen, Yuli Huang, Ngee Chia Tai, Wai Yi Leong, Wei Qi Leong, Xianbin

Yong, Jian Gang Ngui, Yosephine Susanto, Nicholas Cheng, Hamsawardhini Rengarajan, Peerat Limkonchotiwat, Adithya Venkatadri Hulagadri, Kok Wai Teng, Yeo Yeow Tong, Bryan Siow, Wei Yi Teo, Wayne Lau, Choon Meng Tan, and 12 others. 2025. SEA-LION: Southeast asian languages in one network. *Preprint*, arXiv:2504.05747.

ParsBench. 2024. Persian musr: Multistep soft reasoning dataset. Accessed: 2026-05-20.

G. Penedo, H. Kydlíček, V. Sabolčec, B. Messmer, N. Foroutan, A. H. Kargaran, C. Raffel, M. Jaggi, L. Von Werra, and T. Wolf. 2025. FineWeb2: One pipeline to scale them all—adapting pre-training data processing to every language. *arXiv preprint arXiv:2506.20920*.

T. Pires, E. Schlinger, and D. Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Qwen Team, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

P. Rust, J. Pfeiffer, I. Vulić, S. Ruder, and I. Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135. Association for Computational Linguistics.

T. Scialom, T. Chakrabarty, and S. Muresan. 2022. Fine-tuned language models are continual learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6107–6122, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Muhammad Ali Shafique, Areej Mehboob, Layba Fiaz, Muhammad Usman Qadeer, and Hamza Farooq. 2026. Urdubench: An urdu reasoning benchmark using contextually ensembled translations with human-in-the-loop. *arXiv preprint arXiv:2601.21000*.

Muhammad Ali Shafique, Kanwal Mehreen, Muhammad Arham, Maaz Amjad, Sabur Butt, and Hamza Farooq. 2025. Alif: Advancing urdu large language models via multilingual synthetic data distillation. In *Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL)*.

F. Shi, M. Suzgun, M. Freitag, X. Wang, S. Srivats, S. Vosoughi, H. W. Chung, Y. Tay, S. Ruder, D. Zhou, D. Das, and J. Wei. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2024. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Maux Team. 2025. Maux-persian-sft-30k: A large-scale persian conversational dataset.

Wondimagegnhue Tsegaye Tufa, Ilia Markov, and Piek Vossen. 2024. Unknown script: Impact of script on cross-lingual transfer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, pages 124–129, Mexico City, Mexico. Association for Computational Linguistics.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450. Association for Computational Linguistics.

Orgest Xhelili, Yihong Liu, and Hinrich Schütze. 2024. Breaking the script barrier in multilingual pre-trained language models with transliteration-based post-training alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11283–11296, Miami, Florida, USA. Association for Computational Linguistics.

A Language Family Tree

Figure 2 shows the phylogenetic relationships among all languages in our experimental design. The horizontal axis represents genetic distance from Urdu: languages sharing a branch share a common ancestor and tend to share grammatical features, while languages on distant branches diverge in word order, morphology, and script. The rightmost column reports the URIEL genetic distance to Urdu from Table 7, making explicit why Hindi–Urdu transfer (same sub-branch, distance 0.000) is stronger than Persian–Urdu transfer (same family, different branch, 0.691) or Arabic–Urdu transfer (different family entirely, 1.000).

B Language Pair Token Overlap

Table 5 summarises the script-level token inventory. Urdu and Arabic both belong to the Perso-Arabic script family, sharing all 9,447 Arabic-script tokens. Hindi, written in Devanagari, shares **zero tokens** with either. Any Hindi→Urdu advantage over Arabic must therefore reflect deeper representational transfer, not surface token matching.

C Coefficient of Variation Formula

$$\mathrm{CV} = \frac{\sigma}{\mu} \times 100\%$$

where σ is the standard deviation and μ is the mean of the evaluation results.

D URIEL Distance Analysis

Table 7 reports continuous URIEL syntactic and genetic distances alongside downstream performance for both targets.

Tamil–Urdu (0.255) is syntactically *closer* than Persian–Urdu (0.366) or Arabic–Urdu (0.382), yet Tamil performs worst. URIEL syntax vectors capture shared SOV word order but underweight morphological divergence (agglutinative vs. fusional). For the Persian target, all sources are syntactically equidistant (0.37–0.40), rendering syntactic distance uninformative despite performance ranging from 15.3% to 51.4%.

Genetic (family) distance proves more reliable: on the Urdu target, the ranking Hindi (0.000) > Persian (0.691) > Arabic/Tamil (1.000) aligns with downstream performance, as shared ancestry better captures the morphological and structural overlap that drives transfer. However, genetic distance is not a perfect predictor either; on the Persian target, Hindi and Urdu share identical family distance (0.691) yet Hindi transfers more effectively, suggesting that the specific typological properties within a family branch (e.g. word order, morphological type) matter beyond raw phylogenetic proximity. We also note that on the Persian composite, Tamil (43.21) edges Arabic (42.56). This is driven by Tamil's high Belebele FA and MuSR scores outweighing its ParsiNLU collapse (15.34%); on the per-benchmark reasoning tasks the grammar-over-script ranking holds, with Hindi first on both. The composite's equal weighting across benchmarks of differing difficulty makes this a summary artifact rather than evidence that script-only transfer beats a neither-control.

E Practitioner's Guide: Selecting an HRL Source

Given a low-resource target language, the following procedure uses URIEL typological distances to rank candidate high-resource source languages for CPT. The steps operationalise the findings from Sections 4–5; the code requires only the open-source `lang2vec` library (Littell et al., 2017).

1. **Identify candidates.** List high-resource languages for which large monolingual corpora (≥2B words) are available.
2. **Compute genetic distance.** Use `lang2vec fam` vectors. A genetic distance near 0 indicates shared ancestry and correlates with deeper morphological overlap that URIEL syntactic features may underweight (Section 5.3).
3. **Check script overlap.** Count shared subword tokens between each candidate and the target in the model's tokenizer vocabulary. Shared tokens are helpful but not sufficient: our results show that a grammar-matched source with zero token overlap (Hindi→Urdu) outperforms a script-matched source with 9,447 shared tokens (Arabic→Urdu).
4. **Rank and select.** Prioritise candidates with the lowest syntactic *and* genetic distance. When two candidates have similar syntactic distance, prefer the one with lower genetic distance (shared morphological paradigms). Script overlap is a secondary tiebreaker, not a primary criterion.

Example. Suppose the target is a low-resource Turkic language (e.g., Uyghur). Candidate HRLs might include Turkish (same family, SOV, agglu-

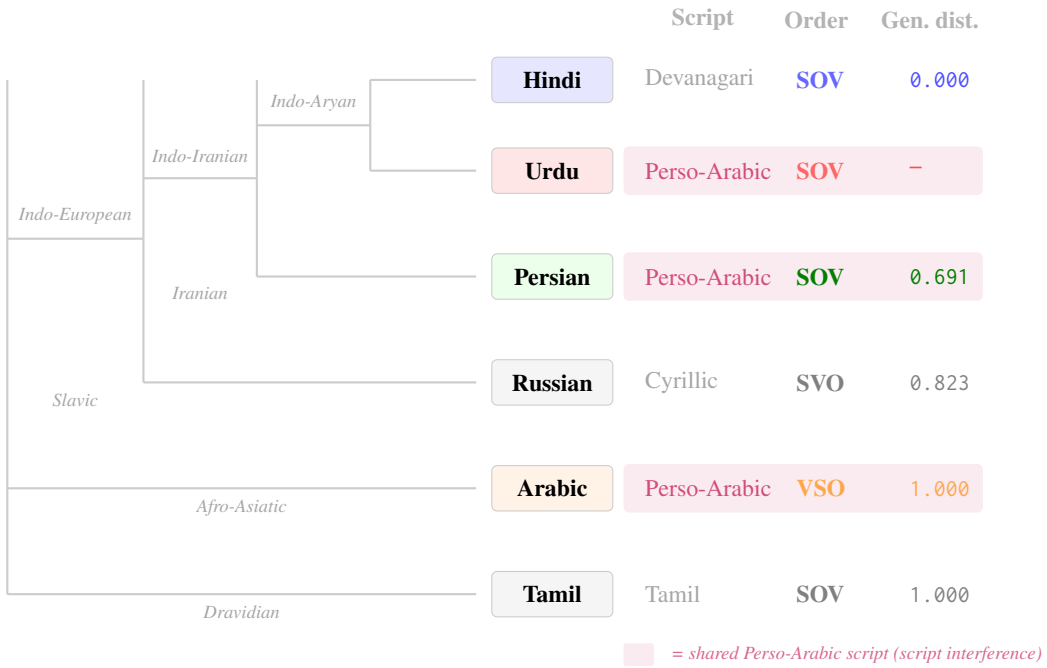

Figure 2: Phylogenetic relationships among languages in the experimental design, shown as a horizontal cladogram derived from Glottolog. Node colours match the 2×2 conditions for the Urdu target: grammar only, script only, both, neither; Urdu is the primary target. The **Script** and **Order** columns show writing system and canonical word order. The **Gen. dist.** column reports URIEL genetic distance to Urdu (Table 7): 0.000 = same sub-branch, 1.000 = different language family. The highlighted rows mark the three languages sharing Perso-Arabic script despite spanning two language families, illustrating script interference (Section 5.2).

Table 5: Subword token overlap between CPT source languages and both target languages in the Tekken vocabulary (131,072 entries). Both targets use Perso-Arabic script; only Perso-Arabic sources share tokens with them.

CPT Source	Script	Tokens	∩ Urdu	∩ Persian
Arabic	Perso-Arabic	9,447	9,447	9,447
Persian	Perso-Arabic	9,447	9,447	—
Urdu	Perso-Arabic	9,447	—	9,447
Hindi	Devanagari	1,569	**0**	**0**
Russian	Cyrillic	3,187	**0**	**0**
Tamil	Tamil	1,042	**0**	**0**

tinative, with low syntactic *and* genetic distance), Arabic (different family, VSO, shared script, with high syntactic distance despite script overlap), and Russian (different family, SVO, high on both dimensions). Based on our Indo-Iranian results, we predict Turkish CPT would transfer best, even if Uyghur uses Arabic script, because grammatical alignment outweighed script overlap on both targets we tested. This prediction awaits verification on non-Indo-Iranian language families.

Minimal code.

```
import lang2vec.lang2vec as l2v
import numpy as np

target = "uig"  # Uyghur
candidates = ["tur", "ara", "rus"]

for feat in ["syntax_knn", "fam"]:
    vecs = l2v.get_features(
        [target] + candidates, feat
    )
    t = np.array([float(x) for x in vecs[target]
                  if x != '--'])
    for c in candidates:
        v = np.array([float(x) for x in vecs[c]
                      if x != '--'])
        d = 1 - np.dot(t, v) / (
           np.linalg.norm(t) * np.linalg.norm(v))
        print(f"{feat}: {c} -> {d:.3f}")
```

Table 6: Language pair selections for the 2 × 2 design.

Target	Source	Script	Grammar	Family	Order
Urdu	Hindi	×	✓	Indo-Aryan	SOV
	Tamil	×	×	Dravidian	SOV†
	Arabic	✓	×	Semitic	VSO/SVO
	Persian	✓	✓	Iranian	SOV
Persian	Hindi	×	✓	Indo-Aryan	SOV
	Russian	×	×	Slavic	SVO
	Arabic	✓	×	Semitic	VSO/SVO
	Urdu	✓	✓	Indo-Aryan	SOV
	Tamil	×	×	Dravidian	SOV†

† Tamil shares SOV word order with Urdu and Persian but differs morphologically (agglutinative vs. fusional), and is therefore classified as having no grammatical overlap.

Target	Source	Condition	Syntactic Distance	Genetic Distance	Composite	Stable?
Urdu	Hindi	Gram. only	**0.049**	**0.000**	**42.07**	yes
	Persian	Both	0.366	0.691	38.61	yes
	Arabic	Script only	0.382	1.000	38.24	yes
	Tamil	Neither	0.255	1.000	34.04	no
Persian	Hindi	Gram. only	0.393	0.691	**53.97**	yes
	Urdu	Both	0.366	0.691	50.72	yes
	Arabic	Script only	0.402	1.000	42.56	no
	Russian	Neither	0.374	0.796	40.81	no
	Tamil	Neither	0.402	1.000	43.21	no

Table 7: URIEL typological distances vs. downstream performance for all source–target pairs. Syntactic Distance. = cosine distance between URIEL `syntax_knn` vectors (103 features covering word order, dependency structure, and POS distributions; lower = more grammatically similar). Genetic Distance. = Glottolog phylogenetic distance, measuring how many steps up the family tree are needed to find a common ancestor (0 = same sub-branch, 1 = no shared ancestry). Composite = unweighted mean of per-benchmark accuracies (multi-seed means). Because benchmarks differ in difficulty (e.g. Math500 ∼12% vs. Belebele ∼63%), the composite is dominated by higher-baseline tasks and should be read as a coarse summary, not a normalised score; per-benchmark results (Tables 1, 2) are authoritative. Stable = coefficient of variation ≤ 10% across four random seeds.

Table 8: Target language properties.

Target	Script	Family	Order
Urdu	Perso-Arabic	Indo-Aryan	SOV
Persian	Perso-Arabic	Iranian	SOV

F SFT Data Filtering

Urdu. We start from the Urdu-Instruct dataset (51,686 examples across seven categories). We first drop three categories not relevant to reasoning: Translation (10,001), Classification (4,662), and Sentiment Analysis (4,347), yielding 32,264 examples. We then remove any example whose output field contains more than five consecutive Latin-alphabet characters (excluding single-character math variables and structural tags), and reformat the remaining examples into the Mistral chat template. After an 80/20 train-validation split, the final training set contains 25,811 examples.

Persian. We start from Maux-Persian-SFT-30k (30,000 conversations from eleven sources). We first remove all conversations containing any Latin-alphabet character in any message, reducing the set to 11,456 examples. We then apply three additional filters: (1) remove conversations where any assistant response is shorter than 50 characters, (2) remove conversations whose first assistant message matches common Persian greeting patterns, and (3) remove translation tasks identified by keyword matching in the user message. After reformatting and an 80/20 split, the final training set contains 9,165 examples.

G Evaluation Datasets

We provide detailed descriptions of all evaluation benchmarks used in this work.

Belebele. Belebele (Bandarkar et al., 2024) is a multilingual reading comprehension benchmark covering over 100 languages. Each instance con-

sists of a short passage followed by multiple-choice questions designed to test information retrieval and reasoning across languages. We use 900 items per language for both Urdu and Persian, following the standard evaluation split.

UrduBench. UrduBench (Shafique et al., 2026) is a curated Urdu reasoning benchmark comprising human-verified and translation-based evaluation sets for language understanding and reasoning. It includes three subsets: MGSM (Shi et al., 2023), which evaluates multi-step mathematical reasoning on 250 grade-school problems; Math500, a 500-question translated mathematics benchmark covering arithmetic and algebraic reasoning; and OpenBookQA, a 500-item multiple-choice science reasoning dataset translated into Urdu. Together, these tasks cover mathematical, scientific, and commonsense reasoning in Urdu under low-resource conditions.

OpenBookQA. OpenBookQA is a multiple-choice question answering dataset requiring elementary science knowledge and simple reasoning. We use the Urdu translation consisting of 500 questions. The task assesses the ability of models to combine factual recall with reasoning under language transfer conditions.

ParsiNLU. ParsiNLU (Khashabi et al., 2021) is a Persian natural language understanding benchmark designed to evaluate reasoning and comprehension across multiple-choice tasks. It includes questions spanning commonsense reasoning, inference, and language understanding in Persian. We use the full benchmark as provided in prior work.

Persian MuSR. Persian MuSR is a Persian translation (ParsBench, 2024) of the MuSR benchmark (Sprague et al., 2024), designed to evaluate multi-step reasoning ability in complex natural language scenarios. Problems require chaining intermediate reasoning steps to arrive at a final answer, making it particularly suitable for testing compositional generalization in low-resource settings.

H Training Hyperparameters

All CPT and SFT runs use the same hyperparameters across language pairs, ensuring that performance differences arise solely from the source language, not training configuration. Training is performed with Mistral's official fine-tuning framework using LoRA (Hu et al., 2022).

SFT steps and seeds. SFT runs use four random seeds (0, 42, 123, 456) per condition, with the CPT checkpoint held fixed. The seed controls only the shuffle order of the SFT training data.

H.1 Per-Language CPT Configurations

Tables 12–17 report the full configuration for each CPT run. Each CPT checkpoint is trained once and reused across all applicable SFT conditions.

H.2 Per-Target SFT Configurations

Tables 18–19 report the full SFT configuration for each target language. All CPT→SFT conditions within a target share the same SFT hyperparameters. Each condition is trained with four seeds (0, 42, 123, 456), for a total of 20 Persian and 16 Urdu SFT runs. The SFT datasets were trained for multiple epochs without observable overfitting based on validation loss. The number of training passes was kept constant across target languages.

I Data Sources

SFT data constraints. All SFT datasets are filtered to contain no English *natural-language* text in model outputs: English prose and translation tasks are removed so the model reasons in the target language's grammar. Code and formatting tokens are retained as structural content.

Details of all datasets used for CPT, SFT, and evaluation are provided in Table 21.

J Evaluation Prompt Templates

All evaluations use the Mistral-NeMo-Instruct-2407 chat template (single user turn, no system prompt, zero-shot). Answers are extracted via regex from the generated text. For Belebele, evaluation is performed with the target-language prompts; the main tables report the target-language prompt (Urdu prompt for the Urdu target, Persian prompt for the Persian target). Table 22 summarises the prompt structure for each benchmark.

Table 9: Factorial decomposition of grammar and script effects. We select one reasoning benchmark per target (Persian: ParsiNLU; Urdu: MGSM) where all four conditions separate well. The grammar-over-script ordering also holds on Belebele and MuSR (Tables 1–2).

Target		Gram. YES	Gram. NO	Gram. eff.	Script eff.
Persian	Script YES	35.94	35.92	**+5.04**	+4.32
	Script NO	36.64	26.58		
Urdu	Script YES	40.80	40.00	**+5.40**	+1.40
	Script NO	44.00	34.00		

Table 10: Seed stability (max CV% across target-language benchmarks). Lower = more stable.

Condition	Source	Persian	Urdu
Both	Urdu / Persian	10.9%	9.4%
Grammar only	Hindi	**2.6%**	**4.2%**
Script only	Arabic	30.1%	7.4%
Neither	Russian / Tamil	51.4%	12.2%

Table 11: SFT validation loss on the Urdu target (mean over 4 seeds; std ≤ 0.003 in all cells). Hindi converges fastest at every checkpoint.

Step	Hindi	Arabic	Persian	Tamil
50	0.883	0.956	0.969	1.068
100	0.805	0.867	0.871	0.942
150	0.783	0.837	0.839	0.904
200	**0.761**	0.813	0.813	0.869
250	0.767	0.818	0.819	0.874
300	0.766	0.816	0.817	0.870

Table 12: CPT configuration: Arabic.

Aspect	Value
Dataset	2B words Arabic + 500M words English replay
Sequence length	4,096
Per-device batch size	4
Effective global batch size	32 (4×8 GPUs, DDP)
Total training steps	33,600
Optimiser	AdamW
Learning rate	6×10^{-5}
Learning rate schedule	OneCycleLR (linear warmup, first 5% of steps)
Weight decay	0.1
LoRA rank	256
LoRA scaling factor (α)	512 (scaling = 2.0)
LoRA dropout	0.0
Random seed	0
Shared across targets	Persian + Urdu

Table 13: CPT configuration: Hindi.

Aspect	Value
Dataset	2B words Hindi + 500M words English replay
Sequence length	4,096
Per-device batch size	4
Effective global batch size	32 (4×8 GPUs, DDP)
Total training steps	34820
Optimiser	AdamW
Learning rate	6×10^{-5}
Learning rate schedule	OneCycleLR (linear warmup, first 5% of steps)
Weight decay	0.1
LoRA rank	256
LoRA scaling factor (α)	512 (scaling = 2.0)
LoRA dropout	0.0
Random seed	0
Shared across targets	Persian + Urdu

Table 14: CPT configuration: Persian.

Aspect	Value
Dataset	2B words Persian + 500M words English replay
Sequence length	4,096
Per-device batch size	4
Effective global batch size	32 (4×8 GPUs, DDP)
Total training steps	32,000
Optimiser	AdamW
Learning rate	6×10^{-5}
Learning rate schedule	OneCycleLR (linear warmup, first 5% of steps)
Weight decay	0.1
LoRA rank	256
LoRA scaling factor (α)	512 (scaling = 2.0)
LoRA dropout	0.0
Random seed	0
Shared across targets	Urdu only

Table 15: CPT configuration: Russian.

Aspect	Value
Dataset	2B words Russian + 500M words English replay
Sequence length	4,096
Per-device batch size	4
Effective global batch size	32 (4×8 GPUs, DDP)
Total training steps	40,800
Optimiser	AdamW
Learning rate	6×10^{-5}
Learning rate schedule	OneCycleLR (linear warmup, first 5% of steps)
Weight decay	0.1
LoRA rank	256
LoRA scaling factor (α)	512 (scaling = 2.0)
LoRA dropout	0.0
Random seed	0
Shared across targets	Persian only

Table 16: CPT configuration: Tamil.

Aspect	Value
Dataset	2B words Tamil + 500M words English replay
Sequence length	4,096
Per-device batch size	4
Effective global batch size	32 (4×8 GPUs, DDP)
Total training steps	62,170
Optimiser	AdamW
Learning rate	6×10^{-5}
Learning rate schedule	OneCycleLR (linear warmup, first 5% of steps)
Weight decay	0.1
LoRA rank	256
LoRA scaling factor (α)	512 (scaling = 2.0)
LoRA dropout	0.0
Random seed	0
Shared across targets	Persian + Urdu

Table 17: CPT configuration: Urdu.

Aspect	Value
Dataset	2B words Urdu + 500M words English replay
Sequence length	4,096
Per-device batch size	4
Effective global batch size	32 (4×8 GPUs, DDP)
Total training steps	32,950
Optimiser	AdamW
Learning rate	6×10^{-5}
Learning rate schedule	OneCycleLR (linear warmup, first 5% of steps)
Weight decay	0.1
LoRA rank	256
LoRA scaling factor (α)	512 (scaling = 2.0)
LoRA dropout	0.0
Random seed	0
Shared across targets	Persian only

Table 18: SFT configuration: Persian target (5 conditions $\times$ 4 seeds = 20 runs).

Aspect	Value
Target language	Persian
Training dataset	Maux-Persian-SFT-30k (cleaned, $\sim$9K; Appendix F)
Validation dataset	Maux-Persian-SFT-30k (cleaned, $\sim$2K; Appendix F)
Sequence length	4,096
Per-device batch size	4
Effective global batch size	16 (4×4 GPUs, DDP)
Total training steps	378
Optimiser	AdamW
Learning rate	6×10^{-5}
Learning rate schedule	OneCycleLR (linear warmup, first 5% of steps)
Weight decay	0.1
LoRA rank	64
LoRA scaling factor (α)	128 (scaling = 2.0)
LoRA dropout	0.0
Random seeds	0, 42, 123, 456
Upstream CPT checkpoints (5 conditions)	
Arabic $\rightarrow$ Persian (Script only)	Arabic CPT, step 33,600
Hindi $\rightarrow$ Persian (Grammar only)	Hindi CPT, step 29,150
Russian $\rightarrow$ Persian (Neither)	Russian CPT, step 40,800
Urdu $\rightarrow$ Persian (Both)	Urdu CPT, step 32,950
Tamil $\rightarrow$ Persian (Neither)	Tamil CPT, step 62,170

Table 19: SFT configuration: Urdu target (4 conditions $\times$ 4 seeds = 16 runs).

Aspect	Value
Target language	Urdu
Training dataset	Urdu-Instruct (cleaned, $\sim$26K; Appendix F)
Validation dataset	Urdu-Instruct (cleaned, $\sim$6K; Appendix F)
Sequence length	4,096
Per-device batch size	1
Effective global batch size	4 (1×4 GPUs, DDP)
Total training steps	300
Optimiser	AdamW
Learning rate	6×10^{-5}
Learning rate schedule	OneCycleLR (linear warmup, first 5% of steps)
Weight decay	0.1
LoRA rank	64
LoRA scaling factor (α)	128 (scaling = 2.0)
LoRA dropout	0.0
Random seeds	0, 42, 123, 456
Upstream CPT checkpoints (4 conditions)	
Arabic $\rightarrow$ Urdu (Script only)	Arabic CPT, step 33,600
Hindi $\rightarrow$ Urdu (Grammar only)	Hindi CPT, step 29,150
Persian $\rightarrow$ Urdu (Both)	Persian CPT, step 32,000
Tamil $\rightarrow$ Urdu (Neither)	Tamil CPT, step 62,170

Table 20: Token counts per CPT corpus. All corpora start from 2B words in the source language plus 500M words of English replay. The fertility column shows tokens per word for the source-language portion only, revealing the 2.7$\times$ compression difference that motivates word-count standardisation.

CPT Language	Total Tokens	Source Tokens†	Fertility
Tamil	8,148,128,868	7,479,561,261	3.74
Russian	5,341,590,538	4,673,022,931	2.34
Hindi	4,563,520,252	3,894,952,645	1.95
Arabic	4,395,420,336	3,726,852,729	1.86
Urdu	4,313,309,955	3,644,742,348	1.82
Persian	4,137,310,928	3,468,743,321	1.73

†Source tokens excludes the 500M-word English replay ($\approx$669M tokens), which is identical across all non-English corpora. Fertility = source tokens / 2B words. The 2.7$\times$ fertility range (Tamil 3.74 vs. English 1.34) motivates our word-count standardisation: a fixed token budget would give Tamil one-third the linguistic content of English.

Table 21: Open-source datasets used for CPT, SFT, and evaluation.

Stage	Language(s)	Dataset
Continual Pre-Training (CPT)		
CPT	Hindi, Urdu, Persian, Arabic	HuggingFaceFW/fineweb-2
CPT	Russian, English	HuggingFaceFW/fineweb-2
CPT	Tamil	SEA-PILE-v2
Supervised Fine-Tuning (SFT)		
SFT	Urdu	Urdu-Instruct (Shafique et al., 2025) (cleaned, 25,811 train)
SFT	Persian	Maux-Persian-SFT-30k (cleaned, 9,165 train)
Evaluation Benchmarks		
Eval	Urdu	UrduBench (MGSM, Math500, OpenBookQA)
Eval	Persian	ParsiNLU
Eval	Persian	MuSR
Eval	All	Belebele (Bandarkar et al., 2024)

Table 22: Evaluation prompt templates. All prompts are wrapped in the Mistral instruct chat template. `{field}` denotes dataset-provided content. Non-Latin labels are rendered as transliterations; the actual prompts use **native script**.

Benchmark	Answer Format	Prompt Structure
Belebele (EN prompt)	Letter (A–D)	`Given the following passage, query, and answer choices, output the letter corresponding to the correct answer.` → `### Passage: {passage} ### Query: {query} ### Choices: (A)...(D)... ### Answer:`
Belebele (UR prompt)	Letter (A–D)	Same ###-delimited structure as EN with Urdu instruction and labels ("Passage"→*iqtibas*, "Query"→*sawal*, "Choices"→*intikhabat*, "Answer"→*jawab*).
Belebele (FA prompt)	Letter (A–D)	Same ###-delimited structure as EN with Persian instruction and labels ("Passage"→*matn*, "Query"→*so'al*, "Choices"→*gozine-ha*, "Answer"→*pasokh*).
UrduBench: MGSM, Math500	Free-form number	Question text passed as-is. No system prompt or additional instructions. Final number extracted via regex.
UrduBench: OpenBookQA	Letter (A–D)	`{question}` + Urdu choices block (*ikhtiyarat*: A–D) + Urdu instruction to select the correct letter (*barah-e karam durust intikhab batayen*) + *jawab*:
ParsiNLU	Free-form	`{instruction} ### {question} ###` + Persian answer cue (*jawab*:). Instruction field is dataset-provided.
Persian MuSR	Letter (A–C)	Persian instruction ("Read the text and write only the letter of the correct choice") → ### + *matn*: `{narrative}` + ### + *so'al*: `{question}` + ### + *gozine-ha*: (A)–(C) + ### + *pasokh*: