One Form to Transfer Them All: Pretraining Multilingual Language Models Beyond Native Orthography

Muge Zhang[1] **Aaron Jencks**[1] **Krishna Badikela**[1]
Yulia Tsvetkov[2] **Sachin Kumar**[1]
[1]Department of Computer Science and Engineering, Ohio State University
[2]Paul G. Allen School of Computer Science & Engineering, University of Washington
contact: {zhang.16414,kumar.1145}@osu.edu

Abstract

Multilingual language models transfer knowledge across languages through shared subword vocabulary, a mechanism that breaks down when related languages use different writing systems. Prior work addresses this via script equalization (romanization or IPA transcription), but direct comparisons are rare; the focus has been on encoder-only models, with most work adapting existing pretrained models. We systematically compare different input representations in autoregressive multilingual pretraining, comparing orthographic text, IPA, and romanization in a controlled setup across three scales (467M, 709M, and 1.03B) on eight languages in four typologically motivated pairs. Across a wide range of downstream tasks on seen and unseen languages, romanized pretraining yields the strongest cross-lingual transfer, and the advantage over text widens with scale. IPA improves over text in most settings but trails romanization. Surprisingly, finetuning a text-pretrained model on romanized data hurts performance on languages already covered by the base model, only marginally helping when the model lacks script coverage. Our results indicate that for multilingual models spanning typologically diverse scripts, to obtain maximum benefits, romanization should be treated as a core design choice applied at pretraining rather than a post hoc fix. Code and datasets will be released upon acceptance.

1 Introduction

In multilingual language models (LMs), vocabulary overlap is the primary source of cross-lingual transfer among languages (Conneau et al., 2020; BigScience Workshop et al., 2022; Üstün et al., 2024). When two languages do not share writing systems, this overlap is largely impossible, even for linguistically close languages as the surface form of the text obscures the similarities (Wu and Dredze, 2020; Muller et al., 2021; Stap et al., 2023).

To increase cross-lingual overlap, prior work has explored different methods of *script equalization*—converting all languages to a shared form (see Figure 1). This is done primarily through two mechanisms. First is romanization (Hermjakob et al., 2018), where non-Latin scripts are mapped to Latin characters using transliteration tools (Purkayastha et al., 2023; Husain et al., 2024; Ebing et al., 2026). Second is phonemization, where text is mapped to phonemic representations such as IPA (International Phonetic Association, 1999), where words that sound alike across different languages would share forms (Nguyen et al., 2023a; Goriely et al., 2024; Jung et al., 2024; Goriely and Buttery, 2025). However, these works have largely focused on improving transfer during *finetuning* with off-the-shelf encoder-only models pretrained with orthographic text, almost exclusively on English. Furthermore, both mechanisms have been explored separately but not compared in modern language models, which are dominantly autoregressive.

In this work, we focus on *pretraining autoregressive* multilingual LMs, comparing both script equalization mechanisms in a controlled setting with orthographic text. We pretrain models from scratch on an eight-language corpus organized into four typologically motivated pairs—English–Spanish, Russian–Polish, Hindi–Urdu, and Tamil–Malayalam—chosen to vary the relationship between orthography and phonology. We train at three scales (467M, 709M, and 1.03B parameters) with each representation, holding architecture, data, vocabulary size, and training procedure constant. We evaluate the trained models under prompting on pretraining-seen languages and under fine-tuning on both seen and unseen languages, covering classification and generative tasks.

We find that romanized pretraining is the strongest configuration in every evaluation regime and at every model size, with the gap from text widening as parameters scale. IPA improves over

Pair	Original Text	IPA	Romanized
EN	The universi ty is very big.	ðə j un ɪvɜːsɪti ɪz vɛɹi bɪg	The universi ty is very big.
ES	La universi dad es muy grande.	la un ibersidad es mui ɣrande	La universi dad es muy grande.
RU	Наша каша вкусная.	naʂa kaʂa fkusnaja	Nas ha kas ha vkusnaya.
PL	Nasza kasza jest smaczna.	naʂa kaʂa jɛst smatʂna	Nas za kas za jest smaczna.
TA	பல்கலைக்கழகம் பெரியது.	pal kala jkkaɻakam peɾijaṯu	pal kala ikkazhakam periyathu.
ML	സർവ്വകലാശാല വലുതാണ്.	sarʋʋa kala ːʃaːla ʋaluṯaːɳ	sarvva kala ashaala valuthaan.
HI	किताब बहुत अच्छी है।	kitaːb bəhut at͡ʃt͡ʃʰiː hɛː	kitaab bahut achchhii hai .
UR	کتاب بہت اچھی ہے۔	kitaːb bəhut at͡ʃt͡ʃʰiː hɛː	kitaab bahut achchhii hai .

Figure 1: Example sentences across four language pairs in original script, IPA, and romanized form. Shared subword tokens are highlighted: blue for original text, gold for IPA, and green for romanized.

orthographic text in most settings and matches romanization in the narrow case of Hindi and Urdu, the pair in our corpus that is most phonologically aligned and most orthographically disjoint, but lags romanization on transfer to unseen languages. Surprisingly, the recipe of romanized finetuning applied to a text-pretrained model, reported as beneficial in prior work, instead regresses performance on the languages a multilingual model already covers; we reproduce the prior gains on a controlled English-only setup and show that the recipe helps only when the base model lacks script coverage in the first place. Together, these results indicate that romanization helps by aligning input with the lexical structure the model has internalized during pretraining; the gains it delivers are therefore most effectively captured by adapting romanization at pretraining rather than downstream. Our findings suggest that future work on multilingual modeling should treat the input representation as a deliberate design choice, with romanization as a strong default when transfer to unseen scripts is a priority.

2 Related Work

Script barrier in multilingual LMs. Multilingual pretrained models (Devlin et al., 2019; Conneau et al., 2020; BigScience Workshop et al., 2022; Aryabumi et al., 2024; Xue et al., 2021; Lin et al., 2022; Grattafiori et al., 2024; Qwen et al., 2024; Yang et al., 2025) transfer well across languages that share scripts but show little improvement or even degrade if not (Wu and Dredze, 2020; Lauscher et al., 2020). Prior studies identify script mismatch, and hence lack of vocabulary overlap, as the dominant failure mode of transfer to unseen languages (Pires et al., 2019; Muller et al., 2021; Stap et al., 2023). Our work targets this barrier by changing the input representations. Prior work has also shown that amount of vocabulary overlap is not correlated with better performance (Meyer and Buys, 2024; Limisiewicz et al., 2023; K et al., 2020). Our results echo these findings: improvements do not track overlap, and, as an example, English–Spanish transfer holds up under IPA even as their shared vocabulary becomes smaller.

Romanization for cross-lingual transfer. A growing body of work has used romanization to expose lexical overlap that the original orthography hides. Prior work on encoder-only models has shown that transliteration, or romanization, can substantially benefit multilingual learning, particularly for low-resource languages, both in Indic settings and across broader multilingual corpora (Purkayastha et al., 2023; Moosa et al., 2023; Ebing et al., 2026; Jung et al., 2026). More recent work has extended these ideas to decoder-only LMs through continued pretraining and instruction tuning on romanized text (Husain et al., 2024), contrastive alignment (Liu et al., 2024), and tokenizer adaptation for transliterated input (Liu et al., 2025a). We instead pretrain decoder-only autoregressive LMs from scratch on romanized text, performing controlled comparisons with orthographic text to isolate when romanization helps and

when it hurts across both finetuning and prompting regimes.

Phonemic representations. A parallel line of work has explored phonemic input as another way to bridge scripts. Nguyen et al. (2023b) introduced XPhoneBERT, a multilingual phoneme-level encoder trained on phoneme sequences from nearly 100 languages for speech-related applications; Jung et al. (2024) showed that phoneme-based models reduce the cross-lingual performance gap; Goriely et al. (2024) pretrained a GPT-2-scale model on phonemized English and found minimal cost on meaning-level tasks. Concurrent work (Miletić et al., 2026) compares Text vs. IPA subword tokenizers across 24 languages and pretrains a 240M model, finding IPA improves text compression without changing task performance. Our work extends this study to include romanization, echoing their findings on compression but also find improved downstream performance with both IPA and romanization over text, contrary to their findings.

3 Experimental Setup

3.1 Language Pairs

We study eight languages organized into four typologically motivated pairs that span a range of orthographic and phonological relationships (see examples in Figure 1).

1. **English–Spanish**: Both are Latin-scripted with shared cognates and borrowed words.
2. **Russian–Polish**: Different scripts (Cyrillic vs. Latin), but related Slavic languages with higher phonological overlap.
3. **Hindi–Urdu**: Different scripts (Devanagari vs. Arabic), but are mutually intelligible in their spoken form (often considered variants of a single language).
4. **Tamil–Malayalam**: Different scripts but from the same language family (Dravidian), with high phonological similarity.

We choose the last three pairs because they provide varying degrees of phonological and phonetic overlap, and stand to benefit via cross-lingual transfer from shared representations. Pairs such as Hindi–Urdu have minimal orthographic overlap but extensive phonemic overlap, so any benefit from a phonemic representation should be especially visible in such cases. English–Spanish serves as a control where text representation already provides shared subword tokens, where phonemic representations might reduce overlap. We train a single multilingual model jointly on all eight languages; the four pairs serve to organize our analysis and to anchor finer-grained comparisons in subsection C.2.

3.2 Pretraining Dataset Construction

For each language pair, we subsample monolingual documents from FineWeb-2 (Penedo et al., 2025), maintaining the word count ratio.[1] Each pair-level corpus is sized to match OpenWebText (Gokaslan and Cohen, 2019). This procedure preserves natural within-pair resource imbalance while equalizing budget across pairs. Within each pair, we randomly sample a subset for validation ($\sim$3.2M words per pair with the same ratio). The four bilingual corpora together form our 8-language corpus with $\sim$21.7B words in total. After tokenization, the word-matched corpus yields 50B Text tokens versus 33B IPA/Romanized tokens, putting IPA/Romanized models at a compute disadvantage and making their gains potentially conservative under FLOP-matched training.

3.3 Text Representations

We compare the following setups. In the first three, we pretrain/finetune with the same representation.

Orthographic text. The original text in each language's native script. For language pairs that share a script (English–Spanish, both Latin), there will be natural token overlap. On the other hand, for language pairs with different scripts (e.g., Russian–Polish, Cyrillic vs. Latin), we expect little overlap, limiting transfer.

IPA transcription. We convert the corpus into IPA using the Phonemizer library (Bernard and Titeux, 2021).[2] To further improve cross-lingual overlap, we remove certain diacritics from the transcriptions, including stress marks (ˈ, ˌ) and length marks (ː), tone accents, and nasalization such that only phonemic segments remain. This logic makes

[1] We use white-space separated tokens as a proxy for words. All of our pretraining languages use white space as delimiters.

[2] We use Phonemizer for its speed and broad language coverage. Alternative phonemizers such as Epitran (Mortensen et al., 2018) and neural grapheme-to-phoneme models (Peters et al., 2017) offer higher per-language accuracy but are much slower and become a bottleneck at our corpus scales. Future work may revisit this quality-throughput tradeoff as faster tools become available.

the representation more coarse-grained, increasing overlap.[3] We additionally modify the Phonemizer library to preserve words containing digits or characters outside the source language's script, which the default pipeline transcribes ambiguously.

Romanization. We romanize all texts using the Uroman library (Hermjakob et al., 2018).[4] For languages already in Latin (e.g., English, Spanish, Polish), the romanized form is close to the original but not always identical.[5]

Text-to-romanized (Text→Rom). In this setup, we finetune our *text*-pretrained checkpoints on romanized downstream task data, mirroring the standard recipe in prior work (Husain et al., 2024; Purkayastha et al., 2023). This tests whether romanization's benefits can be recovered downstream, without the cost of romanized pretraining.[6]

3.4 Tokenization

For each representation, we train a single Byte-Level BPE tokenizer (Sennrich et al., 2016; Radford et al., 2019) jointly on the eight-language corpus using the HuggingFace tokenizers library (Hugging Face, 2019).[7] We use a vocabulary size of 100K to accommodate the combined character inventory across the eight languages. Keeping vocabulary size constant across representations ensures a controlled comparison in which the only variable is the surface form of the text.

Tokenizer overlap analysis. Before pretraining, we quantify the potential for cross-lingual transfer at the lexical level by measuring how much of each tokenizer's vocabulary is shared between languages. Two languages tokenized into largely disjoint token sets cannot easily share embedding-level representations with limited transfer. Therefore, we expect orthographic tokenizers to show high overlap only for script-sharing pairs (English–Spanish), while IPA and romanization are expected to produce higher overlap across similar languages by collapsing scripts into a common symbol set. We report a corpus-size-normalized, frequency-weighted Jaccard overlap between the unique tokens observed in each language's monolingual corpus; full details, including the corrections we apply for rare tokens, punctuation artifacts, code-switching, and corpus-size imbalance, are in Appendix A.

Sequence length analysis. Another consequence of representation choice is how compactly each language's text is encoded. Sequence-length differences across languages are a source of unfairness in multilingual models. Languages that tokenize into more tokens per document require higher training compute, inference latency, per-token API cost, and consume more of the model's effective context window (Petrov et al., 2023; Ahia et al., 2023). We therefore measure token sequence length for the three tokenizers as the average number of tokenizer output tokens per document on the FLORES parallel corpus (NLLB Team et al., 2022), which expresses the same content across all eight languages and therefore makes per-document token counts directly comparable across languages.

3.5 Pre-training

For each representation, we pretrain a causal language model from scratch. We use a modified version of NanoGPT, `modded-nanogpt` (Jordan et al., 2024), which incorporates a number of training efficiency improvements over the original (Karpathy, 2022), obtaining better performance.

Architecture. We train models at three scales to study the interaction between model size and representation, summarized in Table 1. Following `modded-nanogpt`, each model uses value embeddings, which are three additional vocabulary-sized tables whose outputs are mixed into the value projection of the first and last three attention blocks via learned scalars, inspired by the value-residual connections of Zhou et al. (2025). These contribute negligible FLOPs but a large share of parameters; we treat non-embedding parameters as the basis for cross-scale comparison.

Training procedure. All models use a shared optimization setup and checkpoint selection proce-

[3] Retaining these diacritics in early experiments yielded roughly 50% the cross-lingual overlap (weighted Jaccard) of the stripped version, motivating this choice.

[4] We use Uroman because it covers almost all scripts with a rule-based system; Purkayastha et al. (2023) find it performs comparably to or better than language-specific transliterators.

[5] English text passes through unchanged (e.g., *The quick brown fox* → *The quick brown fox*), whereas Spanish and Polish lose diacritics and special characters (e.g., Spanish *el niño está* → *el nino esta*; Polish *Łódź* → *Lodz*).

[6] We omit Text→IPA because IPA has essentially no vocabulary overlap with the orthographic pretraining data; therefore, the pretrained token embeddings provide little useful initialization for IPA tokens.

[7] After the paper submission, we discovered an integer-overflow bug in the BPE training library. Our analysis in Appendix E shows that its effect on our results are negligible and does not alter any conclusions.

Size	L / H / d_{model}	Params	Non-emb.
Small	12 / 6 / 768	467M	83M
Medium	16 / 8 / 1024	709M	197M
Large	20 / 10 / 1280	1.03B	387M

Table 1: Model configurations. Each is trained for all three representations.

dure to ensure controlled comparisons across representations. We use a dual-optimizer setup: AdamW (β_1=0.8, β_2=0.95) for embedding, head, and scalar parameters, and Muon (momentum=0.95) for hidden-layer matrix parameters, with a linear cooldown learning-rate schedule (peak Muon learning rate as 0.025). Each training step processes 524,288 tokens across 8 NVIDIA H100 GPUs. Full training details are provided in Appendix B.

Bilingual reference models. As controlled references that isolate representation effects on individual language pairs, we additionally train bilingual models on each of the four pairs. These models and their analyses are not part of the main results; we describe them and report their behavior in subsection C.2, where they primarily serve as motivation for the multilingual setting that is our main focus.

3.6 Evaluation

We evaluate the models under two regimes: directly prompting the pretrained checkpoints, and supervised fine-tuning on classification and generation tasks. In both settings, we evaluate on languages that were seen during pretraining. To test whether representation effects generalize beyond the training distributions, we finetune on select unseen languages, which include Arabic and French (which share scripts with the pretraining languages) as well as Bengali and Greek (which do not).

Prompting. We score each pretrained model under zero-shot and few-shot setups on two benchmarks: XStoryCloze (Lin et al., 2022) and XCOPA (Ponti et al., 2020). Each benchmark covers a subset of our pretraining languages along with a set of unseen languages, which lets us probe for transfer. Within each ⟨*task, language, representation*⟩ cell, we score every candidate label by likelihood and predict by argmax: per-token-average log-likelihood for label-word tasks (normalized by the number of candidate tokens) (Zhao et al., 2021) and sum log-likelihood for completion-style multiple choice (Brown et al., 2020; Lin et al., 2022). Label words and structural scaffolding are localized to each cell so that all three representations evaluate identical underlying inputs, and the model never sees out-of-representation tokens mid-prompt.

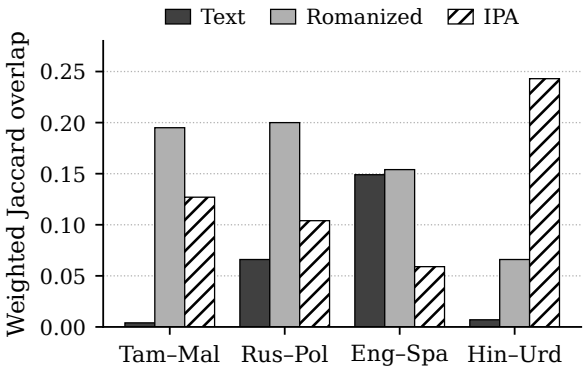

Figure 2: Frequency-weighted subword overlap between languages within each pair under a shared 100K multilingual tokenizer, normalized for corpus size.

Fine-tuning. We finetune the pretrained checkpoints on three downstream tasks: natural language inference (XNLI; IndicNLI, Conneau et al., 2018; Aggarwal et al., 2022), multilingual intent classification (MASSIVE, FitzGerald et al., 2023), and multilingual abstractive summarization (XL-Sum, Hasan et al., 2021). The classification tasks jointly cover all eight pretraining and the unseen languages; For XL-Sum, we finetune on English, Spanish, Russian, Hindi, Urdu, and Tamil (other languages were not available). For each ⟨*task, representation, scale*⟩ cell, both inputs and references (including target summaries) are converted into the corresponding representation using the same Phonemizer and Uroman pipelines as in pretraining. We report macro-F1 for classification tasks and ROUGE-L on XL-Sum. Hyperparameter search and other training details are in Appendix C.

4 Results and Discussion

We organize our results into two parts. We first report cross-lingual subword overlap and sequence length results, which provide context for the downstream results (§4.1). We then report downstream performance across pretraining and fine-tuning regimes, on both seen and unseen languages (§4.2). In all result tables, boldface indicates a statistically significant improvement over the baseline under an approximate randomization test ($p < 0.05$).

4.1 Properties of the Three Representations

Subword overlap between languages. Figure 2 reports frequency-weighted subword overlap be-

Size	Rep.	XNLI									MASSIVE								
		EN	ES	HI	UR	RU	PL	TA	ML	Macro	EN	ES	HI	UR	RU	PL	TA	ML	Macro
Small	Text	75.88	73.78	64.48	60.78	70.89	79.12	69.85	69.84	69.36	75.56	72.62	71.13	63.32	73.10	71.02	65.58	66.16	69.81
	IPA	76.22	73.58	68.07	63.73	69.58	78.40	**71.11**	70.84	70.45	76.87	74.72	74.08	71.63	71.75	72.68	**70.17**	69.21	72.64
	Romanized	**78.99**	**76.25**	**69.65**	**64.98**	**73.86**	**81.81**	**71.80**	**72.71**	**72.61**	**79.90**	**77.80**	**75.70**	**73.70**	**77.80**	**77.40**	**69.80**	**71.90**	**75.50**
	Text→Rom.	74.67	71.72	56.47	59.11	62.77	56.71	62.23	64.07	63.47	66.17	62.00	33.46	29.32	41.19	60.29	32.15	37.42	45.25
Medium	Text	81.98	79.08	70.97	68.75	70.99	79.18	**76.87**	75.76	74.91	81.98	80.31	**80.60**	75.31	80.95	78.49	**77.64**	**80.05**	**79.42**
	IPA	81.09	78.96	73.08	68.55	76.38	**84.30**	**75.95**	76.28	75.76	**82.77**	79.19	**80.97**	**78.68**	80.10	**80.65**	76.57	**79.15**	**79.76**
	Romanized	**83.18**	**80.49**	**74.20**	**70.37**	**77.82**	**84.20**	**76.86**	**77.55**	**77.21**	**83.04**	**81.40**	**80.06**	77.36	**82.54**	**81.45**	**76.84**	78.21	**80.11**
	Text→Rom.	**82.63**	**80.06**	62.95	58.76	67.15	59.10	65.09	65.15	67.61	74.51	71.18	51.48	47.34	56.99	71.12	44.86	50.61	58.51
Large	Text	84.05	81.40	76.55	71.44	79.88	**88.30**	78.40	77.37	79.67	83.99	82.21	84.67	79.89	83.59	82.95	**82.28**	**83.96**	82.94
	IPA	**86.61**	**83.65**	**78.84**	72.08	80.26	**89.10**	**79.72**	**78.98**	**81.16**	**86.75**	83.96	**85.81**	**82.48**	84.80	**84.57**	**82.55**	**84.67**	**84.45**
	Romanized	**86.69**	**84.13**	**79.00**	**73.23**	**81.68**	**88.70**	**79.80**	**79.56**	**81.60**	**86.42**	**85.88**	**86.28**	**82.72**	**86.28**	**85.27**	**82.28**	**84.30**	**84.93**
	Text→Rom.	84.33	81.36	65.58	62.75	70.13	62.48	70.88	65.50	70.38	79.05	75.35	57.13	53.19	61.63	76.09	52.49	57.16	64.01

Table 2: Fine-tuning F1 (%) on the eight pretraining languages. Bold marks the best representation and statistical ties per benchmark, scale, and language. *Text→Rom.* denotes text-pretrained models fine-tuned on romanized data.

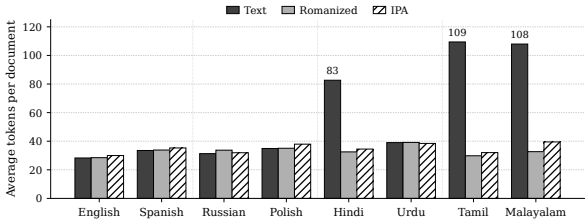

Figure 3: Average sequence length (tokens per document) by language and representation on the FLORES parallel corpus.

tween the two languages in each typological pair under the shared tokenizer. Under orthographic text, meaningful overlap appears only for the Latin-script English–Spanish pair; the three different-script pairs sit at or near zero. Romanization lifts two of those three pairs (Russian–Polish and Tamil–Malayalam) into a range comparable to English–Spanish, but does not close the gap for Hindi–Urdu. IPA inverts this pattern: it produces its highest overlap on Hindi–Urdu, where phonology is nearly identical despite disjoint orthographies, but lags romanization on every other pair. Romanization and IPA are therefore not interchangeable. Prior work has reached mixed conclusions on the role of overlap in cross-lingual transfer (K et al., 2020; Limisiewicz et al., 2023; Meyer and Buys, 2024). We also treat overlap as one channel among several.

Sequence length. Figure 3 reports average tokens per document on the FLORES parallel corpus, where the same content is expressed across all eight languages (NLLB Team et al., 2022). Compared to the other five, Hindi, Tamil, and Malayalam tokens inflate under text by factors of roughly three times their romanized counterparts. IPA and romanization compress them back into the same range

Setting	Size	Rep.	AR	BN	EL	FR	Macro
ZS	Small	Text	6.84	0.94	6.03	**21.97**	8.95
		IPA	5.66	10.25	**10.74**	6.96	8.40
		Romanized	**8.57**	**12.87**	**10.43**	27.68	**14.89**
	Medium	Text	**11.06**	2.38	7.10	31.22	12.94
		IPA	7.87	19.82	**13.50**	13.98	13.79
		Romanized	9.34	**22.47**	**13.37**	**39.29**	**21.12**
	Large	Text	**15.67**	3.56	11.60	36.11	16.73
		IPA	13.01	25.66	**19.20**	21.49	19.84
		Romanized	13.27	**31.24**	**19.33**	**44.35**	**27.05**
FT	Small	Text	34.63	30.66	26.59	47.35	34.81
		IPA	35.93	**48.72**	49.15	45.73	44.88
		Romanized	**41.14**	**48.84**	**52.24**	**54.41**	**49.16**
		Text→Rom.	31.78	36.63	40.10	39.17	36.92
	Medium	Text	51.87	51.77	54.05	62.15	54.96
		IPA	53.68	59.76	**64.94**	61.21	59.90
		Romanized	**55.28**	**61.13**	**65.78**	**65.87**	**62.02**
		Text→Rom.	45.24	49.68	53.44	53.76	50.53
	Large	Text	57.53	59.15	61.47	67.52	61.42
		IPA	61.87	68.36	**70.98**	68.93	67.53
		Romanized	**64.69**	**70.11**	**71.25**	**75.62**	**70.42**
		Text→Rom.	49.58	58.58	61.39	60.30	57.46

Table 3: Cross-lingual transfer to unseen languages on MASSIVE, F1 (%). *Zero-Shot* evaluates pretrained models directly; *Unseen FT* fine-tunes on the unseen language. Bold marks the best representation and its statistical ties per setting, scale, and language.

as the others, consistent with Miletić et al. (2026) who report similar compression gains under IPA tokenization. With text, the inflated languages require roughly three times the inference latency and API costs compared with Latin- and Cyrillic-script peers, and romanization and IPA close this gap.

4.2 Downstream Performance

Multilingual romanized pretraining is the strongest. Pretraining on romanized text produces the strongest cross-lingual transfer across every regime we evaluate: zero- and few-shot in-context learning on XStoryCloze and XCOPA

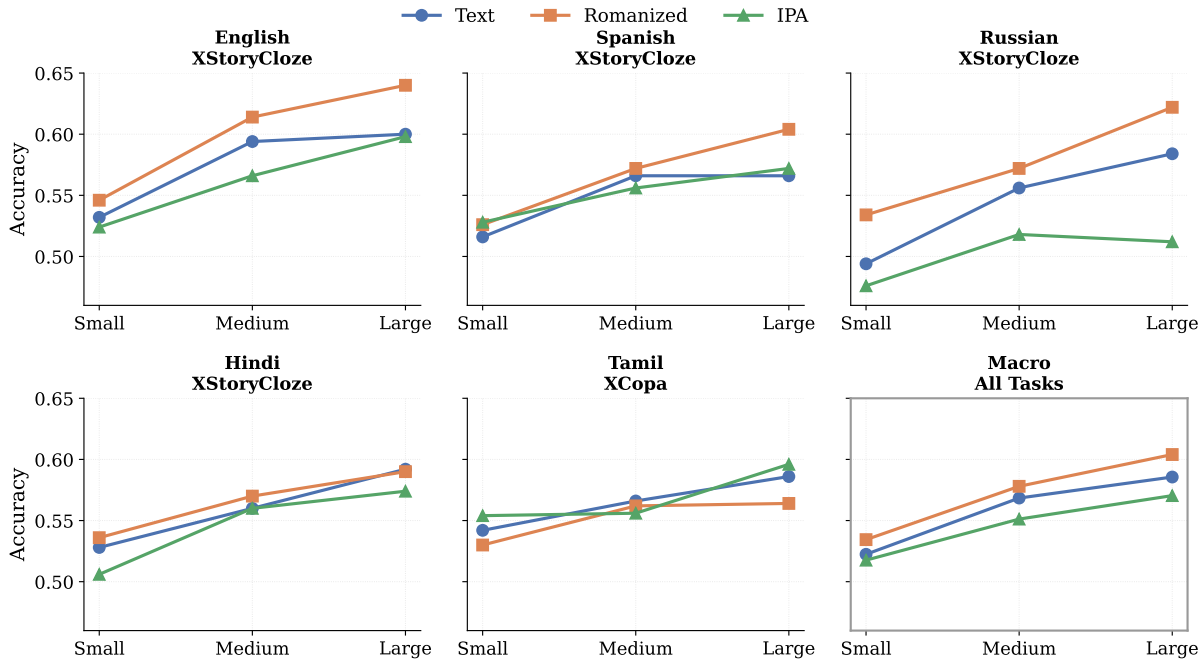

Figure 4: Zero-shot accuracy on XStoryCloze and XCOPA across scales for the three representations. Each panel reports one evaluation language.

Size	Rep.	EN	ES	HI	RU	TA	UR	Macro
Small	Text	17.47	15.70	12.85	10.91	3.48	21.47	13.65
	IPA	11.33	13.41	16.66	9.15	14.22	18.76	12.26
	Romanized	**20.49**	**17.71**	**23.97**	**15.93**	**15.89**	**23.23**	**19.54**
	Text→Rom.	20.12	17.13	12.31	8.91	9.09	20.09	14.63
Medium	Text	22.10	20.49	22.15	16.54	12.23	26.82	20.06
	IPA	21.00	18.63	**29.95**	19.74	**20.99**	29.75	23.34
	Romanized	**24.04**	**20.79**	28.17	**19.75**	20.91	**30.83**	**24.08**
	Text→Rom.	26.51	22.21	21.91	13.02	10.02	18.55	18.73
Large	Text	23.51	20.79	22.55	18.17	17.36	29.36	21.96
	IPA	23.96	21.15	**29.30**	20.31	22.32	**32.69**	24.95
	Romanized	**25.52**	**21.23**	28.45	**22.17**	**22.68**	31.95	**25.33**
	Text→Rom.	27.18	22.83	20.45	15.56	13.38	22.42	20.30

Table 4: XL-Sum summarization, ROUGE-L F1 (%). Bold marks the best representation per scale and language.

Size	Rep.	AR	BN	EL	FR	Macro
Small	Text	61.88	65.37	**66.02**	66.01	64.82
	IPA	61.70	65.31	**65.74**	64.64	64.35
	Romanized	**63.61**	**67.32**	**66.66**	**68.25**	**66.46**
	Text→Rom.	**62.86**	65.25	**65.92**	66.98	65.25
Medium	Text	63.31	64.16	66.38	69.20	65.76
	IPA	63.85	68.33	68.24	67.00	66.86
	Romanized	**68.89**	**72.56**	**71.23**	**75.07**	**71.94**
	Text→Rom.	64.72	68.11	68.47	70.61	67.98
Large	Text	65.64	66.16	68.43	72.33	68.14
	IPA	65.63	71.54	70.52	69.14	69.21
	Romanized	**71.27**	**75.63**	**72.74**	**78.27**	**74.48**
	Text→Rom.	66.59	69.06	69.68	71.78	69.28

Table 5: Cross-lingual transfer to unseen languages on XNLI, F1 (%). Models are fine-tuned on the unseen language's training set. Bold marks the best representation and its statistical ties per scale and language.

(Figure 4; full k-shot results in Appendix D, Table 12; the same ordering holds at intermediate pretraining checkpoints,), supervised fine-tuning on NLI and MASSIVE (Table 2) and XL-Sum summarization (Table 4), and transfer to unseen languages on MASSIVE (Table 3) and XNLI (Table 5). The effect holds at all three model scales. Under prompted setups, the gap to text is smaller than under fine-tuning, but the ranking is preserved.

IPA improves over text in most settings we evaluate. This is consistent with Jung et al. (2026), who find that IPA improves over orthographic text in encoder-only models, especially in unseen languages, and contrasts with prior works (Goriely et al., 2024; Bunzeck et al., 2024), who report phoneme-trained language models slightly underperform their text-trained counterparts. Outside Hindi and Urdu, which share little orthography but are nearly identical phonologically, IPA lags romanization, especially on unseen languages.

To our knowledge, no prior work has directly compared phonemic and romanized pretraining for autoregressive LMs; our results provide the first such evidence and show that the two are not interchangeable. We attribute this to two factors: First, Phonemizer's quality varies across different languages. The phonemization is noisier, especially for low-resource languages (Bernard and Titeux, 2021; Goriely et al., 2024). Second, the phonem-

Size	Rep.	AR	BN	FR	Macro
Small	Text	7.21	2.64	16.40	8.75
	IPA	4.39	6.64	13.29	8.11
	Romanized	**10.72**	**9.08**	**17.79**	**12.53**
	Text→Rom.	10.06	6.48	15.83	10.79
Medium	Text	10.72	5.32	18.37	11.14
	IPA	9.25	9.58	16.90	11.91
	Romanized	**12.45**	**11.80**	**19.57**	**14.61**
	Text→Rom.	11.84	9.54	17.62	13.00
Large	Text	12.58	7.31	17.38	12.42
	IPA	11.29	11.76	20.03	14.36
	Romanized	**15.36**	**12.43**	**21.43**	**16.41**
	Text→Rom.	12.76	8.99	18.20	13.25

Table 6: XL-Sum cross-lingual transfer to unseen languages, ROUGE-L F1 (%). Bold marks the best representation per scale and language.

ized output of an unseen language may also include symbols not produced by the eight pretraining languages, which would compound the problem.

Romanized finetuning a text-pretrained model degrades performance on seen languages. Given the strong results of romanization, a natural intervention is to finetune a text-pretrained model on romanized data, hoping to recover cross-lingual transfer benefits of romanization without paying the cost of romanized pretraining. Husain et al. (2024) report that such an intervention helps when adapting Llama 2 to non-Latin-script languages, and it is the standard recipe in prior romanization-for-transfer work. We apply this intervention to our multilingual text-pretrained checkpoint. Reported as Text→Rom in Tables 2 and 4, we find that, contrary to expectation, it produces large regressions on all pretraining languages across all benchmarks. On unseen languages where the pretrained model lacks script coverage (Greek and Bengali) in the same sense as an English-only model would, the intervention does help, though by much smaller margins than romanized pretraining. Unseen MASSIVE (Table 3) is a partial exception: the benefit holds at the smallest scale but disappears at Medium and Large. We attribute this to MASSIVE being more sensitive to surface-form noise, since intent classification depends on recognizing short trigger phrases that romanization can distort.

Romanized finetuning of a *monolingual* text-pretrained model improves transfer The previous result contradicts prior reports that Text→Rom fine-tuning helps cross-lingual transfer. To further understand the cause of these regressions, we mirror the prior work setup: fine-tuning an English-only pretrained model (Radford et al., 2019) on romanized multilingual downstream data. As shown in Table 7, we do recover the improvements reported by prior work, but improvements achieved through this fine-tuning-time intervention fall well short of what pretraining-time romanization delivers, as reported in previous sections. The gains concentrate in languages with absent or rare scripts. Hindi, Urdu, Tamil, and Malayalam gain 20–30 F1 on MASSIVE at larger scales and move from near-zero to 10–15 on XL-Sum. Latin-scripted languages show small, sometimes negative changes.

These findings qualify Husain et al. (2024)'s results: finetuning helps when the base model lacks coverage of the target script, and degrades performance when such coverage already exists. As most modern open-source and open-weight models are multilingual, covering a wide array of scripts, romanized finetuning will likely lead to uneven results. This asymmetry motivates future work on romanized multilingual pretraining at scale.

Taken together, these results produce a consistent ranking: romanizing at pretraining is best, while romanizing at fine-tuning helps when pretraining lacked script coverage and hurts when it established it. This echoes Xhelili et al. (2024) and Liu et al. (2025b), who find that transliteration's benefits depend on exposing lexical overlap rather than on the operation itself: the same operation supplies overlap when the model lacks script-specific representations and disrupts it when those representations already exist. Characterizing how this plays out in the model's internal representations is a natural next step that we leave to future work.

5 Conclusion

We study input representation as a controlled variable in autoregressive multilingual pretraining, comparing orthographic text, IPA, and Uroman romanization under matched conditions across three scales and eight languages. Romanized pretraining yields the strongest cross-lingual transfer on both seen and unseen languages. IPA improves over text in most settings but only matches romanization on Hindi–Urdu, where phonology is shared, and orthography is disjoint. The widely used recipe of fine-tuning a text-pretrained model on romanized data regresses performance on languages the model already covers, and helps only when the base model lacks script coverage. We recommend that future

		MASSIVE									XL-Sum						
Backbone	**Rep.**	**EN**	**ES**	**PL**	**RU**	**HI**	**UR**	**ML**	**TA**	**Macro**	**EN**	**ES**	**RU**	**HI**	**UR**	**TA**	**Macro**
GPT-2	Text	**54.0**	**30.1**	**28.4**	16.6	9.0	9.3	8.7	5.2	20.2	21.67	14.18	3.35	0.47	0.53	0.00	6.66
	Romanized	44.0	25.1	26.9	**20.2**	**20.5**	**17.1**	**23.5**	**20.3**	**24.7**	**21.94**	**14.57**	**8.46**	**10.52**	**13.84**	**4.53**	**12.31**
GPT-2 Medium	Text	**69.6**	**47.3**	43.9	22.9	14.6	12.8	10.0	11.0	29.0	**24.83**	14.25	3.24	0.43	0.52	0.00	7.18
	Romanized	66.1	43.3	**45.3**	**39.0**	**42.7**	**34.0**	**40.2**	**35.5**	**43.3**	**24.85**	**14.36**	**9.12**	**10.64**	**14.27**	**4.88**	**13.03**
GPT-2 Large	Text	**80.0**	**64.9**	61.5	40.0	32.7	33.1	26.0	21.6	45.0	26.24	14.46	3.38	0.45	0.39	0.00	7.49
	Romanized	77.9	61.6	**62.2**	**59.1**	**56.3**	**53.7**	**60.3**	**56.0**	**60.9**	**26.37**	**15.92**	**8.79**	**10.87**	**14.26**	**4.24**	**13.42**

Table 7: Fine-tuning English-pretrained GPT-2 checkpoints on MASSIVE (F1, %) and XL-Sum (ROUGE-L, %) under the Text and Romanized conditions. Bold marks the better representation per backbone, benchmark, and language. XL-Sum does not include ML.

multilingual pretraining treat input representation as a deliberate design decision alongside data mixture and tokenizer, and adopt romanization when transfer to unseen scripts is a priority.

6 Limitations

Scale. Due to our limited compute budget, we pretrained models up to approximately 1B parameters (387M non-embedding), substantially smaller than contemporary multilingual language models, which typically range from 7B to over 100B parameters (Aryabumi et al., 2024; Grattafiori et al., 2024; Qwen et al., 2024; Yang et al., 2025). Future work may extend this work to larger scale settings.

Language coverage. We selected language pairs intended to highlight differences in orthography alongside varying degrees of phonetic overlap: English and Spanish, Russian and Polish, Hindi and Urdu, and Tamil and Malayalam. These languages primarily represent alphabetic or abugida-based writing systems. Many other classes of writing systems were not represented in this work, including logographic systems such as Chinese. Expanding this analysis to additional languages and orthographic systems remains an important direction for future work. Future work should also investigate how varying degrees of linguistic relatedness and shared inheritance, such as language family distance, phonological similarity, and the presence or absence of shared loanwords, affect the extent to which different representations improve transfer.

Transcription Quality. Phonemization is an active area of research. We elected to use the Phonemizer library because it provided a practical balance between transcription quality and throughput at the scale required for our corpus. Although we performed small-scale comparisons of different G2P tools, a more comprehensive evaluation of the computational cost introduced by transcription during both training and inference is needed. Improving phonemization could change our reported trends as it pertains to IPA.

Additionally, the authors of eSpeak, the backend framework powering Phonemizer, note that support for several languages remains incomplete or insufficient, which may affect transcription consistency and overall model quality for low-resource or under-supported languages. During preprocessing, we also observed limitations in using standard Unicode representations for IPA tokenization, particularly with regard to handling compound symbols and diacritic composition, which led to stripping of diacritics and stress markers (§3.3). During our analysis, we found evidence that Unicode may not be the best representation for phonemes for this kind of work; however, exploring alternative tokenization or encoding strategies for IPA representations is, therefore, left to future work.

Finally, our approach for IPA stripping and preserving numerical information during transcription (§3.3) relied on a relatively simple heuristic-based system. More sophisticated approaches may improve robustness.

Converting IPA and romanized representations back to text. To be useful in generative settings, a language modeling-based system must operate in orthographic text, as a typical user might not be able to read romanized text or IPA. There has been prior research exploring phoneme-to-grapheme (P2G) conversion (Lauc, 2024; Ma et al., 2025; Masumura et al., 2020). However, we did not evaluate the performance or practicality of these approaches within our pipeline.

Existing work on P2G also presents several limitations. Both romanization and phonemization are lossy operations. For example, Hermjakob et al. (2018) note that the transformation process is not fully reversible, potentially resulting in informa-

tion loss during reconstruction. Many of these approaches are language-specific (Shibli et al., 2023; Baruah et al., 2024), though some more generalized approaches have been proposed, such as Sequiera et al. (2014). Evaluating the effectiveness, reversibility, and computational overhead of these systems remains an important direction for future work.

References

Divyanshu Aggarwal, Vivek Gupta, and Anoop Kunchukuttan. 2022. IndicXNLI: Evaluating multilingual inference for Indian languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10994–11006, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? Tokenization in the era of commercial language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9923.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Jon Ander Campos, Yi Chern Tan, Kelly Marchisio, Max Bartolo, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Aidan Gomez, Phil Blunsom, Marzieh Fadaee, and 2 others. 2024. Aya 23: Open weight releases to further multilingual progress. *Preprint*, arXiv:2405.15032.

Hemanta Baruah, Sanasam Ranbir Singh, and Priyankoo Sarmah. 2024. AssameseBackTranslit: Back Transliteration of Romanized Assamese Social Media Text. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1627–1637, Torino, Italia. ELRA and ICCL.

Mathieu Bernard and Hadrien Titeux. 2021. Phonemizer: Text to phones transcription for multiple languages in python. *Journal of Open Source Software*, 6(68):3958.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, and 1 others. 2022. BLOOM: A 176B-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*.

Bastian Bunzeck, Daniel Duran, Leonie Schade, and Sina Zarrieß. 2024. Graphemes vs. phonemes: Battling it out in character-based language models. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 54–64.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin 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, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*.

Benedikt Ebing, Lennart Keller, and Goran Glavaš. 2026. One script instead of hundreds? on pretraining romanized encoder language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 38291–38307.

eSpeak. espeak languages. `https://espeak.sourceforge.net/languages.html`. Accessed: 2026-05-24.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2023. MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4277–4302, Toronto, Canada. Association for Computational Linguistics.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Zébulon Goriely and Paula Buttery. 2025. Ipa childes & g2p+: Feature-rich resources for cross-lingual phonology and phonemic language modeling. In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pages 502–521.

Zébulon Goriely, Richard Diehl Martinez, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2024. From babble to words: Pre-training language models on continuous streams of phonemes. In *Proceedings of the BabyLM Challenge at the 28th Conference on Computational Natural Language Learning (CoNLL)*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Ulf Hermjakob, Jonathan May, and Kevin Knight. 2018. Out-of-the-box universal Romanization tool uroman. In *Proceedings of ACL 2018, System Demonstrations*, pages 13–18, Melbourne, Australia. Association for Computational Linguistics.

Hugging Face. 2019. Tokenizers: Fast state-of-the-art tokenizers. Accessed 2026-02-15.

Jaavid Aktar Husain, Raj Dabre, Aswanth Kumar, Jay Gala, Thanmay Jayakumar, Ratish Puduppully, and Anoop Kunchukuttan. 2024. RomanSetu: Efficiently unlocking multilingual capabilities of large language models via romanization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

International Phonetic Association. 1999. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press.

Keller Jordan, Jeremy Bernstein, Brendan Rappazzo, @fernbear.bsky.social, Boza Vlado, You Jiacheng, Franz Cesista, Braden Koszarsky, and @Grad62304977. 2024. modded-nanogpt: Speedrunning the nanogpt baseline.

Haeji Jung, Jinju Kim, Kyungjin Kim, Youjeong Roh, and David R Mortensen. 2026. Happiness is sharing a vocabulary: A study of transliteration methods. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7797–7816.

Haeji Jung, Changdae Oh, Jooeon Kang, Jimin Sohn, Kyungwoo Song, Jinkyu Kim, and David R. Mortensen. 2024. Mitigating the linguistic gap with phonemic representations for robust cross-lingual transfer. In *Proceedings of the 4th Workshop on Multi-lingual Representation Learning (MRL) at EMNLP*.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. *Preprint*, arXiv:1912.07840.

Andrej Karpathy. 2022. NanoGPT. `https://github.com/karpathy/nanoGPT`.

Davor Lauc. 2024. Polyipa – multilingual phoneme-to-grapheme conversion model. *Preprint*, arXiv:2412.09102.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran 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)*.

Tomasz Limisiewicz, Jiří Balhar, and David Mareček. 2023. Tokenization impacts multilingual language modeling: Assessing vocabulary allocation and overlap across languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5661–5681, Toronto, Canada. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, and 2 others. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yihong Liu, Chunlan Ma, Haotian Ye, and Hinrich Schuetze. 2024. Translico: A contrastive learning framework to address the script barrier in multilingual pretrained language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2476–2499.

Yihong Liu, Chunlan Ma, Haotian Ye, and Hinrich Schütze. 2025a. Transmi: A framework to create strong baselines from multilingual pretrained language models for transliterated data. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 469–495.

Yihong Liu, Mingyang Wang, Amir Hossein Kargaran, Ayyoob ImaniGooghari, Orgest Xhelili, Haotian Ye, Chunlan Ma, François Yvon, and Hinrich Schütze. 2025b. How transliterations improve crosslingual alignment. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2417–2433.

Te Ma, Min Bi, Saierdaer Yusuyin, Hao Huang, and Zhijian Ou. 2025. Llm-based phoneme-to-grapheme for phoneme-based speech recognition. *Preprint*, arXiv:2506.04711.

Ryo Masumura, Naoki Makishima, Mana Ihori, Akihiko Takashima, Tomohiro Tanaka, and Shota Orihashi. 2020. Phoneme-to-Grapheme Conversion Based Large-Scale Pre-Training for End-to-End Automatic Speech Recognition. In *Interspeech 2020*, pages 2822–2826. ISCA.

Francois Meyer and Jan Buys. 2024. A systematic analysis of subwords and cross-lingual transfer in multilingual translation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2194–2200, Mexico City, Mexico. Association for Computational Linguistics.

Milan Miletić, Julie Kallini, and Ekaterina Shutova. 2026. Phonemes to the rescue: Multilingual tokenization based on international phonetic alphabet. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 40323–40349.

Ibraheem Muhammad Moosa, Mahmud Elahi Akhter, and Ashfia Binte Habib. 2023. Does transliteration help multilingual language modeling? In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 670–685, Dubrovnik, Croatia. Association for Computational Linguistics.

David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*.

Hoang H. Nguyen, Chenwei Zhang, Tao Zhang, Eugene Rohrbaugh, and Philip S. Yu. 2023a. Enhancing cross-lingual transfer via phonemic transcription integration. In *Findings of the Association for Computational Linguistics: ACL 2023*.

Linh The Nguyen, Thinh Pham, and Dat Quoc Nguyen. 2023b. Xphonebert: A pre-trained multilingual model for phoneme representations for text-to-speech. *arXiv preprint arXiv:2305.19709*.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. 2025. Fineweb2: One pipeline to scale them all–adapting pre-training data processing to every language. *arXiv preprint arXiv:2506.20920*.

Ben Peters, Jon Dehdari, and Josef van Genabith. 2017. Massively multilingual neural grapheme-to-phoneme conversion. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 19–26.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. *Advances in neural information processing systems*, 36:36963–36990.

Telmo Pires, Eva Schlinger, and Dan 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.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Sukannya Purkayastha, Sebastian Ruder, Jonas Pfeiffer, Iryna Gurevych, and Ivan Vulić. 2023. Romanization-based large-scale adaptation of multilingual language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2024. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI*. Accessed: 2024-11-15.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Royal Denzil Sequiera, Shashank S Rao, and B. R. Shambavi. 2014. Word-Level Language Identification and Back Transliteration of Romanized Text. In *Proceedings of the 6th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '14, pages 70–73, New York, NY, USA. Association for Computing Machinery.

G. M. Shahariar Shibli, Md. Tanvir Rouf Shawon, Anik Hassan Nibir, Md. Zabed Miandad, and Nibir Chandra Mandal. 2023. Automatic back transliteration of Romanized Bengali (Banglish) to Bengali. *Iran Journal of Computer Science*, 6(1):69–80.

David Stap, Vlad Niculae, and Christof Monz. 2023. Viewing knowledge transfer in multilingual machine translation through a representational lens. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14973–14987, Singapore. Association for Computational Linguistics.

Ahmet Üstün, Viraat Aryabumi, Zheng Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, and 1 others. 2024. Aya model: An instruction finetuned open-access multilingual language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15894–15939.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP) at ACL*.

Orgest Xhelili, Yihong Liu, and Hinrich Schuetze. 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.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. 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.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. Pmlr.

Zhanchao Zhou, Tianyi Wu, Zhiyun Jiang, Fares Obeid, and Zhenzhong Lan. 2025. Value residual learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28341–28356, Vienna, Austria. Association for Computational Linguistics.

A Tokenizer Overlap Computation Details

We compute cross-lingual subword overlap as follows. Given the multilingual tokenizer and two languages L_1 and L_2, we tokenize each language's monolingual corpus separately and let T_{L_i} denote the set of unique tokens observed in L_i. The simplest measure of shared vocabulary is the Jaccard index over these token sets:

$$J(L_1, L_2) = \frac{|T_{L_1} \cap T_{L_2}|}{|T_{L_1} \cup T_{L_2}|}. \quad (1)$$

Higher values indicate greater shared vocabulary and, by the argument in section 3.4, greater potential for transfer through shared embeddings (Conneau et al., 2020).

Raw Jaccard, however, is sensitive to several confounds: rare tokens contribute equally to common ones, punctuation and whitespace artifacts inflate apparent overlap, code-switched lines leak tokens across languages, and unequal corpus sizes bias the unique-token counts. We therefore complement raw Jaccard with an adjusted variant that applies four corrections. First, we filter for contamination by retaining only documents whose language tag matches the target language and removing code-switched or noisy lines (motivated by language-switch warnings observed during preprocessing). Second, we exclude punctuation- and whitespace-only tokens, which would otherwise dominate the intersection. Third, we replace set-based Jaccard with a frequency-aware weighted variant that weights each token by its actual usage in the two corpora:

$$J_w(L_1, L_2) = \frac{\sum_t \min(c_{L_1}(t), c_{L_2}(t))}{\sum_t \max(c_{L_1}(t), c_{L_2}(t))}, \quad (2)$$

where $c_{L_i}(t)$ is the count of token t in the corpus of L_i. This downweights tokens that are technically shared but rare in one language. Fourth, we control for corpus-size imbalance by sampling matched token budgets per language and averaging the resulting overlap over multiple random samples.

Representation	Small	Medium	Large
Text	1.365	1.043	0.868
IPA	1.143	0.899	0.818
Romanized	1.087	0.862	0.775

Table 8: Multilingual pretraining bits per character by representation across scales. Values are not directly comparable across representations.

Rep.	ENG–SPA	HIN–URD	RUS–POL	TAM–MAL
IPA (stripped)	0.989	0.897	0.893	0.872
Text (orig.)	0.994	0.703	0.997	1.044
Phonemes	0.992	**0.650**	**0.786**	**0.712**
Romanized	**0.945**	0.735	0.937	0.808

Table 9: Pretraining bits-per-character (BPC) across input representations and language pairs at the **Small** scale. Bold indicates the best representation per language pair. Lower is better.

B Pretraining details

Training procedure. We use a dual-optimizer setup: AdamW (β_1=0.8, β_2=0.95) for embedding, head, and scalar parameters, and Muon (momentum=0.95) for hidden-layer matrix parameters, with a linear cooldown learning-rate schedule (peak Muon learning rate as 0.025). Each training step processes 524,288 tokens across 8 NVIDIA H100 GPUs. We train each model for 3 epochs over the training split of the pretraining corpus, and select the checkpoint with the lowest validation loss on the held-out split for downstream evaluation.

Bits per character. We report multilingual pretraining bits per character (BPC) by representation across model scales in Table 8, and bilingual pretraining BPC at the per-pair level in Table 9 and Table 10 for small and medium models, respectively. We caution that BPC values normalized by representation-specific characters are not directly comparable across representations with different vocabularies and surface forms, since the denominator differs across configurations. We report these numbers for completeness and as evidence that all configurations trained successfully; we do not draw cross-representation conclusions from them, and rely instead on downstream task performance (§4.2) as the operational measure of pretraining quality.

Pretraining-corpus sequence lengths. Figure 5 reports average tokens per document on our pretraining corpus, FineWeb-2.

Compared to the four Latin- and Cyrillic-script languages, Hindi, Urdu, Tamil, and Malayalam tokens inflate under text by factors of roughly 2.5–3.5 times their romanized counterparts, with the largest gaps for Tamil and Malayalam. IPA and romanization compress them back into the same range as the others. Under a fixed per-step token budget, text models therefore cover 2.5–3.5 times fewer documents per step in these four languages than IPA or romanized models do, and romanization and IPA close this gap during training.

Rep.	ENG–SPA	HIN–URD	RUS–POL	TAM–MAL
IPA (stripped)	0.838	0.765	0.723	0.759
Text (orig.)	0.857	**0.509**	0.823	0.784
Phonemes	0.828	0.556	**0.650**	**0.618**
Romanized	**0.812**	0.649	0.783	0.704

Table 10: Pretraining bits-per-character (BPC) across input representations and language pairs at the **Medium** scale. Bold indicates the best representation per language pair. Lower is better.

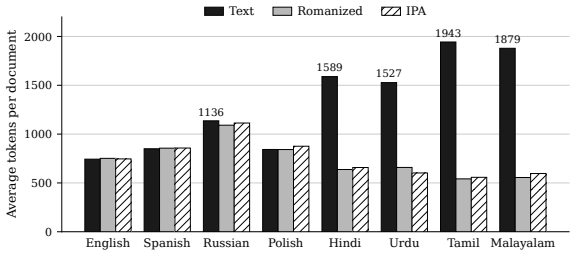

Figure 5: Average sequence length (tokens per document) by language and representation on the pretraining corpus.

C Fine-tuning Details

C.1 Training Procedure

For each *(task, representation, scale)* cell, we perform a grid search over learning rate $\in \{5\times10^{-5}, 1\times10^{-4}, 2\times10^{-4}, 3\times10^{-4}\}$ and batch size $\in \{128, 256\}$, training for three epochs with linear warmup and decay. We select hyperparameters based on the lowest validation loss. For seen languages, we fine-tune jointly across all languages in a single multilingual run; for unseen languages, we fine-tune a separate model per language.

C.2 Bilingual Downstream Results

Table 11 reports cross-lingual transfer performance for bilingual models at a small scale. These results informed the design of our multilingual experiments presented in the main body.

Romanization or IPA outperforms text on seven

Table 11: Bilingual cross-lingual transfer F1 at the small model scale. Each cell reports the best F1 across transfer directions and hyperparameters. Bold marks the best representation per language pair.

Dataset	Pair	Text	IPA	Rom.
XNLI	Eng–Spa	0.85	0.81	**0.86**
	Hin–Urd	0.34	**0.64**	0.51
	Rus–Pol	0.35	0.40	**0.43**
IndicXNLI	Tam–Mal	0.68	0.68	**0.71**
MASSIVE	Eng–Spa	**0.588**	0.332	0.578
	Hin–Urd	0.030	**0.400**	0.040
	Tam–Mal	0.213	0.500	**0.573**
	Rus–Pol	0.490	0.480	**0.610**

of eight (dataset, pair) settings; text wins only on MASSIVE Eng–Spa, the one pair that already shares Latin script. The gaps are largest for related-language pairs with disjoint scripts, where text often collapses to near-zero F1 while IPA and romanization recover substantial transfer. Romanization is the most consistent winner across settings, while IPA's largest gains concentrate on Hin–Urd, the pair with the closest phonological overlap and the most disjoint scripts.

D In-Context Learning

D.1 Full k-shot results

Table 12 reports the same evaluations as the main-text figure across three scales (Small, Medium, Large) and four shot counts (k=0, 1, 2, 4). Three patterns hold across the full table. First, few-shot demonstrations provide little reliable improvement over zero-shot: within-scale variation across k is small and often non-monotonic, while increasing model scale yields consistent gains for representations trained from scratch in the target surface form. Second, IPA's gap relative to text and romanization persists across k values, indicating that its ICL deficit is a property of the representation rather than an artifact of the zero-shot setting. Third, the text→rom rows—text-pretrained models evaluated on romanized inputs at inference time—generally underperform the same models evaluated on text, and gain less from added demonstrations or scale than the matched-representation configurations do; an inference-time surface-form mismatch is not rescued by in-context examples.

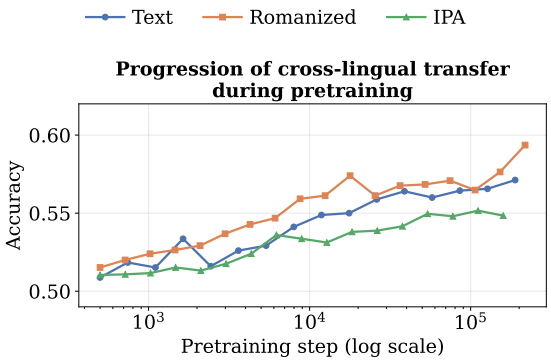

Figure 6: Zero-shot cross-lingual transfer accuracy on XStoryCloze and XCOPA at intermediate pretraining checkpoints (medium models, log-scale x-axis).

D.2 Progression of cross-lingual transfer during pretraining

To test whether a shared surface form speeds convergence rather than only raising final accuracy, we evaluate cross-lingual transfer at intermediate pretraining checkpoints (Figure 6). The ordering is stable almost from the outset: romanized > text > IPA at nearly every checkpoint. The gaps do not close as training proceeds. Romanization's advantage is persistent.

E Impact of the Tokenizer Integer-Overflow Bug

All results in the main paper are reported with tokenizers trained using the `tokenizers` library. The `tokenizers` library (Hugging Face, 2019) accumulates byte-pair frequencies in a signed 32-bit integer during BPE training, so pairs occurring more than 2^{31} times overflow and are dropped or mis-ranked.[8] We patched the accumulator to 64-bit, verified byte-identical output below the threshold, and retrained the tokenizers and affected models. Then we re-evaluated on MASSIVE fine-tuning and zero-shot prompting. Re-running the remaining evaluations (XNLI, XL-Sum) was beyond our compute budget.

Tables 13 and 14 compare the two settings. MASSIVE macro-F1 changes by at most 0.46 points and zero-shot average accuracy by at most 0.007. In both cases, the changes are significantly smaller than the gaps between the representations. Every pairwise ordering of the three representations is preserved at every scale.

Cross-lingual transfer is stable as well.

[8] https://github.com/huggingface/tokenizers/issues/2058

Task	Lang	Repr.	Small				Medium				Large-std			
			k=0	k=1	k=2	k=4	k=0	k=1	k=2	k=4	k=0	k=1	k=2	k=4
XStoryCloze	en	text	.532	.540	**.552**	.524	.594	.596	**.600**	.594	.600	.606	.610	**.612**
		romanized	.546	.550	**.552**	.522	.614	.602	**.618**	.604	**.640**	.612	.622	.622
		ipa	**.524**	.520	**.524**	.514	.566	.564	.570	**.572**	.598	.604	.590	**.612**
XStoryCloze	es	text	**.516**	.508	.514	.502	**.566**	.552	.556	.538	.566	.568	.568	**.570**
		romanized	**.526**	.520	.520	.486	.572	.576	.568	**.590**	**.604**	.592	**.604**	.592
		ipa	**.528**	.520	.526	.520	.556	**.572**	.562	.566	.572	**.590**	**.590**	.584
		text→rom	.514	.506	**.516**	.492	**.550**	.540	.542	.542	.576	.558	.566	**.580**
XStoryCloze	ru	text	.494	**.498**	.488	.496	.556	**.560**	**.560**	.554	.584	.580	.582	**.588**
		romanized	**.534**	.520	.520	.494	.572	.568	**.574**	.562	.622	.618	.622	**.624**
		ipa	.476	.472	**.478**	.460	**.518**	.510	.508	.514	.512	.514	**.520**	.516
		text→rom	**.456**	**.456**	**.456**	.454	.458	.464	.456	**.472**	.448	.464	.462	**.478**
XStoryCloze	hi	text	**.528**	.502	.504	.470	.560	.570	.570	**.584**	**.592**	.568	.568	.566
		romanized	.536	**.540**	.536	.526	.570	**.572**	.566	.558	**.590**	.568	.576	.580
		ipa	**.506**	.504	.486	.476	.560	**.568**	.556	.564	.574	**.584**	.566	.576
		text→rom	**.476**	**.476**	.464	.462	.476	.464	**.482**	.456	.478	**.480**	.472	.470
XCopa	ta	text	**.542**	.540	.536	.532	.566	.572	**.580**	.574	.586	.582	**.592**	.578
		romanized	.530	.524	**.534**	.530	.562	.562	.564	**.574**	.564	**.582**	.580	.576
		ipa	.554	.556	.554	**.558**	.556	**.580**	.576	.570	.596	**.606**	.580	.590
		text→rom	.564	**.570**	**.570**	.564	.558	.554	.566	**.572**	**.564**	.550	**.566**	.550

Table 12: Accuracy on XStoryCloze and XCOPA across scales and k-shot settings. Bold marks the best score per row within each scale block.

Frequency-weighted subword overlap within each language pair, computed as described in Appendix A, changes at most 0.012 (Table 15).

Therefore, our conclusions are unchanged under either setting.

F AI Disclosure

We used generative AI tools for limited writing and coding assistance, including grammar, clarity, organization suggestions, and debugging support. All research ideas, experimental design decisions, code, results, analysis, and final text were reviewed and verified by the authors.

Size	Rep.	Tok.	EN	ES	HI	UR	RU	PL	TA	ML	Macro
Small	Text	orig	**75.56**	**72.62**	**71.13**	**63.32**	**73.10**	71.02	65.58	**66.16**	**69.81**
		i64	74.38	72.41	71.04	63.17	72.36	**71.63**	**66.31**	65.92	69.65
	IPA	orig	**76.87**	74.72	**74.08**	71.63	71.75	**72.68**	70.17	**69.21**	72.64
		i64	76.69	**75.38**	73.27	**72.51**	**71.84**	72.42	**70.91**	68.57	**72.70**
	Romanized	orig	79.90	**77.80**	75.70	**73.70**	77.80	**77.40**	69.80	71.90	75.50
		i64	**80.73**	77.62	**75.91**	73.48	**78.46**	76.35	**69.92**	**72.71**	**75.65**
Medium	Text	orig	81.98	**80.31**	80.60	75.31	**80.95**	78.49	77.64	80.05	79.42
		i64	**82.33**	79.14	**80.71**	**75.38**	80.21	**79.26**	**78.17**	**80.84**	**79.51**
	IPA	orig	82.77	**79.19**	80.97	**78.68**	80.10	**80.65**	76.57	79.15	79.76
		i64	**83.61**	79.04	**81.02**	77.91	**81.03**	79.84	**77.26**	**79.68**	**79.92**
	Romanized	orig	83.04	81.40	80.06	**77.36**	**82.54**	81.45	76.84	**78.21**	80.11
		i64	**83.17**	**81.56**	**80.14**	76.47	82.38	**82.13**	**77.51**	77.83	**80.15**
Large	Text	orig	83.99	**82.21**	84.67	79.89	83.59	**82.95**	82.28	83.96	82.94
		i64	**85.06**	81.48	**85.22**	**80.94**	**84.17**	82.26	**83.61**	**84.44**	**83.40**
	IPA	orig	**86.75**	**83.96**	**85.81**	82.48	84.80	**84.57**	82.55	**84.67**	**84.45**
		i64	86.21	83.76	85.31	**83.15**	84.80	84.06	82.55	84.40	84.28
	Romanized	orig	86.42	**85.88**	**86.28**	**82.72**	**86.28**	**85.27**	82.28	84.30	**84.93**
		i64	**86.45**	84.77	85.51	82.35	85.78	84.87	**82.41**	**85.24**	84.67

Table 13: Fine-tuning F1 (%) on MASSIVE with the original (*orig*) and i64-fixed (*i64*) tokenizers. Bold marks the better setting per language and scale; *Macro* averages the eight languages.

			Small		Medium		Large	
Task	Lang	Repr.	orig	i64	orig	i64	orig	i64
XSC	en	text	**.532**	.528	**.594**	.588	**.600**	.584
		rom.	**.546**	.542	.614	.614	**.640**	.634
		ipa	.524	**.528**	**.566**	.562	**.598**	.588
XSC	es	text	**.516**	.508	.566	**.572**	**.566**	.560
		rom.	**.526**	.522	**.572**	.568	**.604**	.600
		ipa	**.528**	.518	**.556**	.546	.572	.572
XSC	ru	text	.494	**.506**	.556	**.562**	.584	**.586**
		rom.	.534	**.536**	.572	**.578**	.622	.622
		ipa	**.476**	.470	**.518**	.516	.512	**.522**
XSC	hi	text	**.528**	.526	.560	**.570**	**.592**	.584
		rom.	.536	**.542**	**.570**	.562	**.590**	.586
		ipa	.506	**.516**	**.560**	.554	.574	**.582**
XCOPA	ta	text	**.542**	.530	**.566**	.558	**.586**	.582
		rom.	.530	**.552**	.562	.562	.564	.564
		ipa	.554	**.564**	.556	**.564**	**.596**	.592
Avg	–	text	**.522**	.520	.568	**.570**	**.586**	.579
		rom.	.534	**.539**	**.578**	.577	**.604**	.601
		ipa	.518	**.519**	**.551**	.548	.570	**.571**

Table 14: Zero-shot accuracy on XStoryCloze (XSC) and XCOPA (ta) with the original (*orig*) and i64-fixed (*i64*) tokenizers. Bold marks the better setting per scale; *Avg* averages the five task–language cells.

	Text		IPA		Romanized	
Pair	orig	i64	orig	i64	orig	i64
Tam–Mal	0.004	0.004	0.127	0.125	0.195	0.191
Rus–Pol	0.066	0.067	0.104	0.103	0.200	0.199
Eng–Spa	0.149	0.149	0.059	0.064	0.154	0.153
Hin–Urd	0.007	0.007	0.243	0.231	0.066	0.065

Table 15: Frequency-weighted subword overlap within each language pair under the original (*orig*) and i64-fixed (*i64*) tokenizers.