

mPhonBench: Evaluating LLMs On Multilingual Phonology

Lennart Keller and Goran Glavaš

Center for Artificial Intelligence and Data Science

University of Würzburg

{firstname}.{lastname}@uni-wuerzburg.de

Abstract

Phonological competence—knowledge of how surface characters encode underlying pronunciation—underpins applications from poetry generation to speech recognition. Although text-only Large Language Models (LLMs) receive no dedicated phonetic supervision, prior work shows that they nonetheless develop such competence in English. It remains, however, unclear (1) whether this generalizes across languages and scripts and (2) how this text-based phonological knowledge interacts with the audio processing competencies of speech-augmented LLMs (SLLMs). In this work, we introduce mPhonBench, a massively multilingual benchmark that probes text-only and speech-augmented LLMs for phonological knowledge, spanning 10 tasks, 58 diverse languages, and 19 scripts. While mPhonBench is text-centric, we complement its nine text-only tasks with word-level phone recognition (PR) to jointly analyze text- and speech-based phonological knowledge in SLLMs. We evaluate a wide selection of open-weight LLMs on mPhonBench, and find that: (i) mid-sized LLMs match or surpass expert systems on grapheme-to-phoneme conversion (G2P) and generalize beyond memorization; (ii) underrepresented languages incur a performance penalty; (iii) while this penalty can be partially offset via in-context learning with a few task examples, models struggle to exploit general phonological descriptions of languages provided in context; and (iv) frontier SLLMs are competitive with expert models on word-level PR, but the capability is not linked to model size, suggesting that training regimes play a key role in developing audio-based phonological competence.

1 Introduction

Scaling Large Language Models (LLMs) in both parameter count and training data has led to a

class of models that have become the go-to solution for a wide range of text domains (Brown et al., 2020; Srivastava et al., 2023) and tasks, and that increasingly integrate other input modalities (Team, 2026). An underinvestigated class of tasks encompasses problems that require *phonological competence*, i.e., knowledge of how surface-level characters encode underlying speech sounds (e.g., poetry or song lyrics generation (Ormazabal et al., 2022; Ding et al., 2025) or poetry analysis (Jannidis et al., 2025)). Acquiring this knowledge from text alone is challenging, and especially so in multilingual settings where the same (sequences of) characters are pronounced differently across languages. Phonology thus provides a compelling test case for LLMs along two complementary dimensions: (a) because text-only LLMs acquire knowledge exclusively from text, phonological tasks reveal whether and how they form representations that extend beyond the orthographic surface forms encountered in pretraining. Recent work suggests that they do, at least for English: LLMs encode structured phonetic representations (McLaughlin et al., 2025) and develop specialized rhyme-planning circuits (Lindsey et al., 2025) despite receiving no dedicated phonetic supervision. (b) beyond the direct applications above, phonological knowledge underpins a broader range of NLP systems: it bridges text and speech in recognition and synthesis (Masumura et al., 2020; The Nguyen et al., 2023), serves as a language-agnostic medium for cross-lingual transfer (Jung et al., 2024), exposes the limits of subword tokenization (Liao and Shi, 2026), and offers a useful signal for machine translation (Liu et al., 2019). This renders investigation of the extent to which the general-purpose LLMs—increasingly used as the backbones of such systems—intrinsically possess phonological competence highly consequential. The evidence from existing efforts is restricted in scope: PhonologyBench (Suvana et al., 2024) finds that LLMs possess phonological capabilities

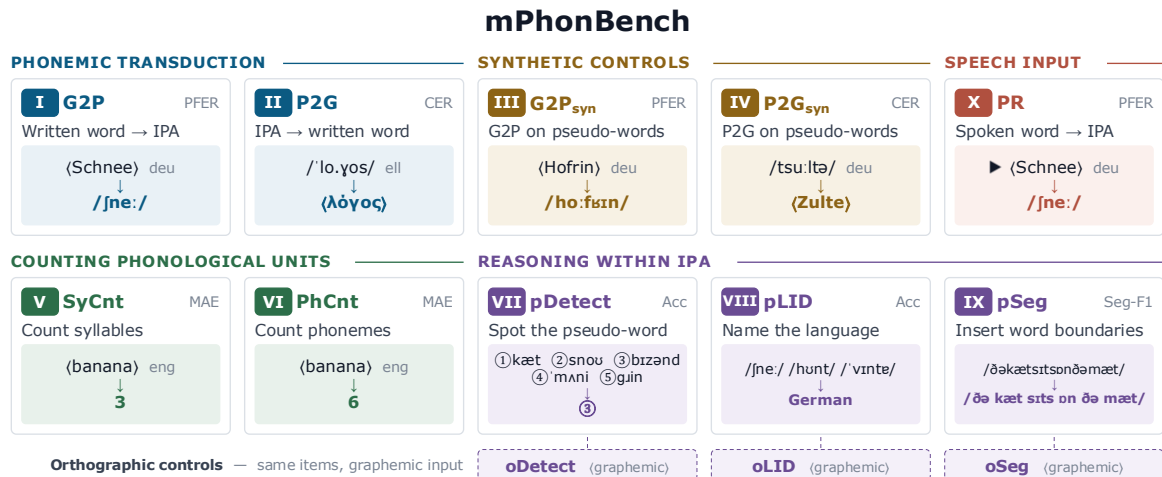


Figure 1: mPhonBench Task Overview

and that these skills scale with model size, but its findings are limited to English. The only massively multilingual phonology benchmark to date, PRiSM (Bharadwaj et al., 2026), focuses on speech as the primary language modality and evaluates dedicated phone-recognition systems rather than general-purpose LLMs. It thus remains unclear whether and how phonological competence in text-only LLMs extends beyond English, to languages with not only orders of magnitude less (pre)training data but also different phonological inventories and orthographic systems. Further, it remains to be seen how phonological knowledge obtained from large text corpora interacts with speech representations of speech-augmented LLMs (SLLMs).

To address these gaps, we introduce the **mPhonBench**: a massively multilingual benchmark for evaluating LLMs spanning 10 tasks, 58 languages, and 19 script systems (see Appendix M for the full language inventory with per-language statistics). Though mPhonBench is text-centric, we complement its nine text-based tasks with a (speech-based) word-level phone recognition task, which we use to probe the interaction between text-based phonological knowledge of LLMs and speech-based competence of speech encoders in SLLMs. We evaluate an extensive suite of state-of-the-art open-weight LLMs across multiple sizes and prompting regimes, and report the following key findings: (i) LLMs are very competitive multilingual phonemic transducers; they seem to encode generalizable phonological rules, with capabilities that scale with model size. (ii) These capabilities are not equally shared across languages: underrepresented languages incur a performance penalty, which can only in part

be offset via in-context learning (ICL) with task-specific examples. (iii) The ICL gains are driven by task examples; despite their “reasoning” abilities, LLMs do not appear able to leverage higher-level descriptions of a language’s phonology to improve performance. (iv) In contrast to the abilities of text-only LLMs, the phone recognition capabilities of state-of-the-art SLLMs do not emerge from scale alone: this suggests that training regimes play a crucial role in developing audio-based phonological competence, requiring further dedicated study. We release mPhonBench with evaluation code, hoping to catalyze research on phonological competence of (S)LLMs.¹

2 Multilingual Phonology Benchmark

2.1 Data Source & Preprocessing

Unless stated otherwise, we source all benchmark data from WikiPron (Lee et al., 2020), an open, massively multilingual repository pairing orthographic word forms with their International Phonetic Alphabet (IPA) transcriptions. IPA transcriptions can express the same phonemic content in multiple ways, and as a crowdsourced resource, WikiPron inherits the idiosyncratic notational conventions of its source Wiktionary entries. This results in substantial cosmetic variation in glyph choice, diacritic ordering, and segment boundaries. We therefore normalize transcriptions using the Broad IPA (BIPA) standard (Anderson et al., 2019), which maps IPA to a fixed segment inventory, canonicalizes diacritic order, and resolves Unicode homoglyphs. We apply

¹<https://github.com/LennartKeller/mPhonBench>

further filters to ensure data quality; see Appendix A for the detailed outline of the pipeline.

2.2 Tasks I & II: Phonemic Transduction

Grapheme-To-Phoneme transduction (G2P) is the task of producing a word’s phonemic representation given its orthographic form. It is arguably the most widely used phonological task, with G2P models being used for synthetic data generation for downstream applications (The Nguyen et al., 2023). G2P is a longstanding task: specialized multilingual models with strong performance have evolved, from static rule- and lookup-based approaches (Mortensen et al., 2018) to neural transducers (Peters et al., 2017; Route et al., 2019; Clematide and Makarov, 2021; Zhu et al., 2022; Dong et al., 2022; Han et al., 2024).

Phoneme-To-Grapheme conversion (P2G) is the inverse task: a phonemic sequence is transduced into the orthographic form. It is applied primarily in speech recognition pipelines, where it converts the phone predictions of an acoustic model into orthographic output (Li et al., 2022; Ma et al., 2025).

Data. We evaluate models on the word level, i.e., we construct G2P and P2G examples for each transcription pair in the WikiPron data.

2.3 Tasks III & IV: Synthetic Transduction

Our G2P and P2G data are derived from a publicly available online source and may have leaked into the pretraining corpora of the evaluated LLMs. With (pre-)training corpora of most open-weight LLMs undisclosed, we cannot measure this contamination directly. We therefore devise an indirect probe: we additionally evaluate G2P and P2G on synthetic words ($G2P_{syn}$, $P2G_{syn}$) that could not have been memorized in pretraining; the gap between the real and synthetic G2P and P2G performance thus approximates the contribution of memorization.

Data. We generate synthetic (grapheme, phoneme) pairs by aligning each language’s WikiPron entries with a lightweight grapheme-to-phoneme aligner (Jiampojamarn et al., 2007), splitting the aligned pairs into position-tagged syllable pairs (*initial*, *medial*, *final*, *mono*), and recombining them into pseudo-words whose length and syllable composition follow the language’s empirical distribution. We discard candidates that match another WikiPron entry of the same language, sampling the same number of instances per language as in tasks I and II. We provide examples in Appendix B.

2.4 Tasks V & VI: Counting Phonological Units

Syllables are the key unit of phonological organization above phonemes. Accordingly, they have been the focus of interest in psycholinguistics (Ukrainetz et al., 2011), speech representation learning (Cho et al., 2024, 2025), and poetry generation (Ormazabal et al., 2022). Following PhonologyBench, we adopt the syllable counting (SyCnt) task and complement it with phoneme counting (PhCnt). Counting syllables from orthography requires both deriving the word’s pronunciation and grouping phoneme segments into syllables. As PhCnt only requires the former, the gap between the two tasks isolates the difficulty of syllabification. For both tasks, we provide the models with a word in its orthographic form and prompt them to return the number of syllables or phonemes.

Data. For PhCnt, the reference count is the number of segments in the BIPA-normalized transcription; the BIPA inventory determines boundaries, so multi-character segments (e.g., affricates, aspirated stops) count as one. For SyCnt, the reference count is the number of syllables delimited by the syllable and stress markers in the IPA transcription. As many WikiPron entries mark these inconsistently, we retain an entry only if its marker-derived count agrees with an estimate of syllable counts based on the principles of Sonority Sequencing and Maximum Onset (Clements, 1990).

2.5 Task VII: Pseudo-Transcript Detection

Pseudo-words (i.e., strings that respect a language’s phonotactic or orthographic patterns without occurring in its lexicon) have a long history as probes of linguistic generalization, from the wug test (Berko, 1958) to recent studies of morphological generalization in LLMs (Weissweiler et al., 2023; Hofmann et al., 2025). In Pseudo-Transcript Detection (pDetect), we test whether a model can identify pseudo IPA transcriptions that are locally well-formed but not attested in the language. We use UniPseudo (New et al., 2024) to generate pseudo transcriptions based on character n -grams and their transitional probabilities. In contrast to the synthetic transcriptions of Section 2.3, UniPseudo-generated forms are not guaranteed to respect syllabic structure and hence only capture local phonotactic regularities, so any higher-order features (e.g., vowel harmony, syllable structure) are unlikely to be preserved. We then pair a single pseudo tran-

scription with four real transcriptions and prompt the model to identify the pseudo transcription by returning its index.

Data. We generate pseudo transcriptions of different lengths, adhering to the real word length distribution of the language. We discard any candidate that (i) appears in the global WikiPron index across all languages or (ii) matches an existing source word or (iii) a previously generated pseudo-word.

2.6 Task VIII: Language Identification

Language identification (LID), including fine-grained variant identification, is a fundamental task in multilingual NLP across both text (Kargaran et al., 2023; Fedorova et al., 2026) and speech modalities (Pratap et al., 2024; Peng et al., 2024). Text-based LID can rely heavily on surface cues, above all script identity. We posit that a phonologically competent model should be able to identify a language from a phonemic transcription, too. Here, without orthographic surface cues, the model must draw on phonotactic patterns and language-specific phoneme inventories. pLID tests this directly: the model is given a phonemic transcription and must name the language.

Data. We construct the dataset by sampling three IPA words from each language. To avoid language ambiguity, we exclude any phonemic form attested in more than one language.

2.7 Task IX: Phonological Segmentation

Segmenting a continuous stream of phones into word(-like) units by observing transitional probabilities is hypothesized to be one of the milestones in infant language acquisition (Saffran et al., 1996), and has been used to study the emergence of similar pathways in LLMs (Goriely et al., 2024; Goriely and Buttery, 2025). In pSeg, we probe whether LLMs acquire this competence as an emergent ability by prompting the model to insert word boundaries in a sentence, presented to the model as one long string of phonemes without whitespaces.

Data. We sample sentences from Wikipedia dumps, excluding those non-trivial to phonemize or tokenize (e.g., containing numbers or abbreviations). We phonemize them with eSpeak NG and normalize the output with BIPA. We discuss the impact of phonemizer choice in §C in the Appendix.

2.8 Task X: Phone Recognition

To handle spoken language, LLMs are commonly augmented with speech encoders (Xu et al., 2025a,b; Microsoft et al., 2025), which encode phonetic features themselves (Choi et al., 2024). To gain insight into how phonetically rich representations from speech encoders interact with phonological knowledge of LLMs, we add Phone Recognition (PR), a speech-based counterpart of G2P. Given a spoken word, the model has to generate its phonemic transcription.

Data. To ensure a fair comparison between the G2P and PR capabilities of models, we construct a word-level speech-to-IPA dataset by pairing WikiPron entries with audio recordings of the same words from the Multilingual Spoken Words Corpus (MSWC) (Mazumder et al., 2021). To avoid over-representing frequent words, we sample from words pronounced by at most 10 different speakers. Due to the smaller language coverage of MSWC, we construct PR pairs for a subset of 19 languages.

2.9 mPhonBench Summary

Orthographic Variants. For pDetect, pLID, and pSeg, we additionally construct orthographic counterparts (i.e., oDetect, oLID, oSeg) as sanity checks and upper-bound performance estimates (see Appendix D for further details).

Metrics. We evaluate (i) G2P ($G2P_{syn}$) and PR with Phonetic Feature Error Rate (PFER) (Mortensen et al., 2016), (ii) P2G ($P2G_{syn}$) with Character Error Rate (CER), (iii) the counting tasks (SyCnt, PhCnt) with mean absolute error (MAE), (iv) the detection and LID tasks with accuracy, and (v) the segmentation tasks with word-boundary F_1 .

Summary. Figure 1 summarizes the benchmark’s tasks along with illustrative examples. The benchmark comprises 452,996 examples across 10 core tasks (with 3 additional orthographic task variants), covering 58 languages (with two dialectal variants for three languages: English, Portuguese, and Bengali). Phone recognition is supported for 19 of these languages. For pSeg, we include data for 11 more languages (69 in total). See Appendix M for detailed statistics.

3 Evaluation

3.1 Models

LLMs. We evaluate models from five popular LLM families: (i) Gemma-3 (Team et al., 2025),

(ii) Gemma-4, (iii) Qwen3 (Yang et al., 2025), (iv) Qwen3.5 (Qwen Team, 2026), and (v) GPT-OSS. We evaluate across multiple model sizes to test whether phonological abilities emerge with scale. We include both Mixture-Of-Experts (MoE) and dense models.

Speech-augmented LLMs. For PR, we evaluate two of the above models that also support speech input (Gemma-4-E2B-it and Gemma-4-E4B-it) alongside three additional SLLMs: Qwen2.5-Omni-7B (Xu et al., 2025a), Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025b), and Phi-4-multimodal-instruct (Microsoft et al., 2025).

Baselines. With the exception of G2P and PR, most mPhonBench tasks are newly designed tasks with no established baselines. For G2P, we use CharsiuG2P (Zhu et al., 2022), a neural transducer trained on WikiPron data, on which we based the benchmark. This instantiates substantial train-test leakage, which we mitigate by reporting the results on a separate subset of WikiPron data, not used in CharsiuG2P’s training. For PR, we compare against three state-of-the-art multilingual phone recognizers: (i) the Wav2Vec2-XLSR CTC model of Xu et al. (2022), trained on data from 53 languages, (ii) ZIPA (Zhu et al., 2025), trained for 88 languages, and (iii) POWSM (Li et al., 2026), a Whisper-style (Radford et al., 2022) multi-task model trained on the same data as ZIPA. We detail inference configurations for all baselines in Appendix E.

3.2 Prompting

For inference, we resort to several prompting strategies, probing different aspects of phonological reasoning and context use:

Zero- and Few-Shot. We evaluate models in *zero-shot* (ZS) and *few-shot* (FS1, FS5) configurations with zero, one, and five in-context task examples, respectively, prompting the models to emit the final answer *without* intermediate reasoning.

Reasoning. To measure how well Chain-of-Thought (CoT) reasoning compensates for the absence of task examples, we additionally evaluate models in a *zero-shot reasoning* (ZS+R) setup: no task examples, but an explicit prompt to produce a step-by-step reasoning chain before the answer.

Abstract In-Context-Learning. A growing body of work investigates how LLMs leverage *abstract* (i.e., task-agnostic) language descriptions from dictionaries or grammar books (Tanzer et al., 2024;

Zhang et al., 2024a,b; Hus and Anastasopoulos, 2024; Aycock et al., 2025; Pei et al., 2025). Phonological processing is a compelling test case for this kind of transfer-based ICL, because a language’s core phonology (i.e., the phoneme inventory and phonotactic rules) can be concisely described in a single document. We scrape the Wikipedia article on the phonology of each mPhonBench language (or the phonology section of the main language article where no dedicated phonology article exists), prepend it to the prompt, and instruct the model to consult it when generating the reasoning chain and final answer. We refer to this setting as *zero-shot reasoning with context* (ZS+R+C) and evaluate it for G2P, P2G, G2P_{syn}, P2G_{syn}, and PR. We provide the complete templates for all prompt variants in §K in the appendix.

3.3 Inference & Decoding

To maximize inference throughput and use our computational resources efficiently, we use vLLM (Kwon et al., 2023) with continuous batching.

Decoding. Song et al. (2025) render greedy decoding comparable to (or better than) sampling-based decoding in domains with sharp correctness boundaries (e.g., math, code). We argue that mPhonBench tasks also meet this criterion: all queries have a single correct answer (i.e., a transcript or a count). To avoid run-to-run variance of sampling, we thus adopt greedy decoding throughout.

Reasoning. Several LLMs in our selection support *internal* reasoning, i.e., generating a reasoning chain in a special format before emitting the final user-facing response. In preliminary experiments, however, we found that enabling this mode sharply increases degenerate responses (e.g., reasoning chains without a final answer, or repetition loops) across the Gemma-4 and Qwen families, both under greedy decoding and each model’s default decoding strategy. We therefore disable internal reasoning across the board and instead prompt the models to generate a reasoning chain as part of the user-facing response under ZS+R and ZS+R+C.² This mitigates the issue for most families, but some models still produce degenerate responses; where this occurs, we assign task-specific penalty scores rather than excluding the affected instances, to keep the evaluation comparable across models.

²For GPT-OSS models, we set the reasoning effort to *low*, since reasoning cannot be entirely disabled.

4 Results & Analyses

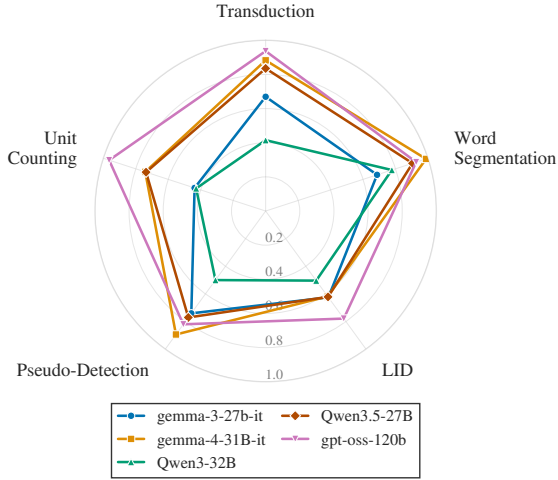


Figure 2: ZS performance of *flagship* models across mPhonBench task families, shown as mean percentile ranks (rescaled to $[0, 1]$; 1 = best), averaged over tasks and languages. Ranks are computed against all evaluated models.

Zero-Shot. We report the ZS results for all models on our nine text-based tasks in Table 1. GPT-OSS-120B emerges as the best model overall, winning 6 of 9 tasks, with Gemma-4-31B-it winning the remaining three. Surprisingly, the smaller (20B) GPT-OSS variant is among the worst models: this is due to its high rate of degenerate responses (see Appendix G for detailed per-task failure rates). Across the other families, we observe a *recency trend*, i.e., a tendency of newer models to outperform their predecessors (Gemma-3/Qwen3 generally trail Gemma-4/Qwen3.5 at comparable scales). This is also reflected in Figure 2, which compares the mean percentile ranks of the *flagship* models (i.e., each family’s largest (dense) model) across task families: Gemma-4 and Qwen3.5 outperform their predecessors on all task families except pLID, where Gemma-3 and Gemma-4 are virtually tied. Within-family comparisons show that tasks generally respond positively to scale, with larger models yielding the largest relative gains on pLID and pSeg. We believe that pLID gains in particular can be credited to increased parametric knowledge of larger models.

Looking across the tasks, we find that only the largest models exceed random performance (0.2) on pDetect by a meaningful margin. pLID, likewise, appears challenging across the board. The results on the respective orthographic counterparts of pLID, pDetect, and pSeg (shown as subscript in corresponding columns of Table 1) let us discern

the contribution of the phonemic modality from general task difficulty. Across all three task pairs, the gaps are substantial: this indicates that the tasks themselves (e.g., language identification) are not inherently difficult—the phonemic input makes them challenging. Comparing the two counting tasks (PhCnt and SyCnt), our assumption that counting phonemes is easier than counting syllables is not supported by the results: all models perform substantially better on SyCnt, with PhCnt being particularly difficult for the Qwen3 family and exhibiting no trends across model sizes. Finally, controlling for total parameter counts, we find that MoE (i.e., sparse) models nearly match the performance of dense counterparts of comparable size.

Few-Shot and Reasoning. Based on the zero-shot results, we select the best-performing model from each family for a targeted evaluation of the two few-shot prompting variants (FS1 and FS5) and the zero-shot reasoning variant without additional context (ZS+R). Table 2 summarizes the results (full results for all models in Appendix J in the appendix). Gemma-4-31B-it benefits the most from in-context examples and overtakes GPT-OSS-120B as the best model at FS5 (best on 6 of 9 tasks vs. 3 of 9 in ZS evaluation). For each task and prompt variant, the best performance belongs to either Gemma or GPT-OSS, which remains strongest on PhCnt and SyCnt (both at FS1) as well as pLID (at ZS+R) and P2G_{syn} (at FS5). Comparing across the prompt variants individually, most models benefit from more in-context examples, albeit to varying degrees. A notable exception is GPT-OSS, whose performance degrades on 6 of 9 tasks, moving from FS1 to FS5. Transduction tasks (G2P, P2G, G2P_{syn}, P2G_{syn}) generally benefit more from in-context examples than from reasoning, whereas for pLID and pDetect we observe the best scores under ZS+R. We hypothesize that these two tasks benefit from reasoning because it allows the model to explore different hypotheses (i.e., in pLID, the model can compare the observed phonotactic signature against candidate languages, and in pDetect it can check each of the five candidates for phonotactic plausibility). In contrast, few-shot examples are most useful for tasks where they reveal repeating patterns, i.e., the input-output token alignments for transduction tasks. Qwen3.5-27B collapses under reasoning, as do other models in the family (see Appendix J). The degradation (i) persists under models’ default decoding configurations and (ii) does not fully man-

	G2P ↓	P2G ↓	G2P _{syn} ↓	P2G _{syn} ↓	pSeg ↑	pLID ↑	pDetect ↑	PhCnt ↓	SyCnt ↓
gemma-3-4b-it	0.143	0.535	0.165	0.582	0.186 _{+0.349}	0.114 _{+0.344}	0.215 _{+0.089}	1.024	0.770
gemma-3-12b-it	0.105	0.441	0.136	0.536	0.329 _{+0.334}	0.132 _{+0.349}	0.222 _{+0.194}	0.843	0.580
gemma-3-27b-it	0.085	0.372	0.122	0.498	0.407 _{+0.385}	0.191 _{+0.383}	0.253 _{+0.193}	1.417	0.440
gemma-4-E2B-it	0.113	0.435	0.139	0.468	0.021 _{+0.247}	0.099 _{+0.301}	0.223 _{+0.052}	1.805	0.292
gemma-4-E4B-it	0.103	0.368	0.132	0.425	0.239 _{+0.340}	0.130 _{+0.328}	0.226 _{+0.185}	1.345	0.258
gemma-4-26B-A4B-it	0.085	0.319	0.126	0.417	0.434 _{+0.312}	0.173 _{+0.380}	0.252 _{+0.246}	0.642	0.187
gemma-4-31B-it	0.075	<u>0.267</u>	<u>0.116</u>	0.381	0.585 _{+0.257}	<u>0.223</u> _{+0.342}	0.314 _{+0.224}	0.895	0.196
Qwen3-4B	0.134	0.557	0.153	0.634	0.269 _{+0.326}	0.111 _{+0.323}	0.201 _{+0.121}	0.917	0.553
Qwen3-8B	0.145	0.475	0.168	0.545	0.364 _{+0.329}	0.128 _{+0.381}	0.212 _{+0.176}	1.072	0.567
Qwen3-14B	0.123	0.457	0.152	0.532	0.416 _{+0.323}	0.149 _{+0.402}	0.216 _{+0.185}	1.446	0.322
Qwen3-30B-A3B	0.112	0.440	0.138	0.512	0.390 _{+0.332}	0.136 _{+0.397}	0.195 _{+0.173}	1.434	0.340
Qwen3-32B	0.119	0.420	0.149	0.516	0.434 _{+0.338}	0.146 _{+0.408}	0.222 _{+0.168}	1.076	0.713
Qwen3.5-0.8B	0.217	0.926	0.225	1.026	0.005 _{+0.109}	0.049 _{+0.304}	0.210 _{-0.021}	7.773	3.840
Qwen3.5-2B	0.170	0.606	0.189	0.679	0.094 _{+0.249}	0.066 _{+0.369}	0.205 _{+0.095}	2.250	0.995
Qwen3.5-4B	0.137	0.438	0.169	0.483	0.141 _{+0.465}	0.098 _{+0.426}	0.209 _{+0.145}	1.777	0.696
Qwen3.5-9B	0.115	0.385	0.148	0.439	0.253 _{+0.432}	0.129 _{+0.435}	0.216 _{+0.184}	1.390	0.320
Qwen3.5-27B	0.086	0.290	0.123	0.368	0.510 _{+0.325}	0.214 _{+0.394}	0.263 _{+0.234}	0.835	0.228
Qwen3.5-35B-A3B	0.086	0.312	0.125	0.393	0.351 _{+0.375}	0.208 _{+0.388}	0.241 _{+0.251}	1.471	0.231
gpt-oss-20b	0.260	0.475	0.279	0.540	0.222 _{+0.240}	0.070 _{+0.407}	0.133 _{+0.208}	1.024	0.305
gpt-oss-120b	<u>0.077</u>	0.231	0.108	0.295	<u>0.524</u> _{+0.285}	0.285 _{+0.300}	<u>0.287</u> _{+0.166}	0.372	0.118

Table 1: Zero-shot performance on mPhonBench tasks, averaged over languages. **Bold** marks the best value in each column; underline marks the second-best. Subscripts for pLID, pDetect, and pSeg indicate the gap w.r.t. performance on orthographic counterparts (oLID, oDetect, oSeg); positive values indicate orthographic task variants are easier.

ifest in pLID; we thus attribute it to a genuine model instability rather than our evaluation setup.

Influence of Context. We next report on the utility of including general-purpose phonological information (rather than task examples) in the models’ context. To this end, we compare the zero-shot reasoning inference with (ZS+R+C) and without (ZS+R) the Wikipedia description of a language’s phonology in the context. Table 3 displays the results. Our best-performing models from the overall evaluation do not benefit from direct access to general phonological knowledge of a language for which they need to carry out a phonological task. We observe either marginal gains or degradations, both well within a single standard deviation. Our finding that LLMs benefit from in-context examples but fail to exploit explicitly provided and useful linguistic knowledge aligns well with results from zero-shot machine translation for low-resource languages, with in-context provided grammar books: Pei et al. (2025) show that LLMs recognize and exploit concrete parallel examples from the book but barely use anything else.

Transduction. Figure 3 contrasts zero-shot G2P and P2G transduction performance on real and synthetic words. All models perform worse on synthetic data, but the gap is roughly constant across scales — indicating that performance does not rely on memorization alone, but rather reflects a degree

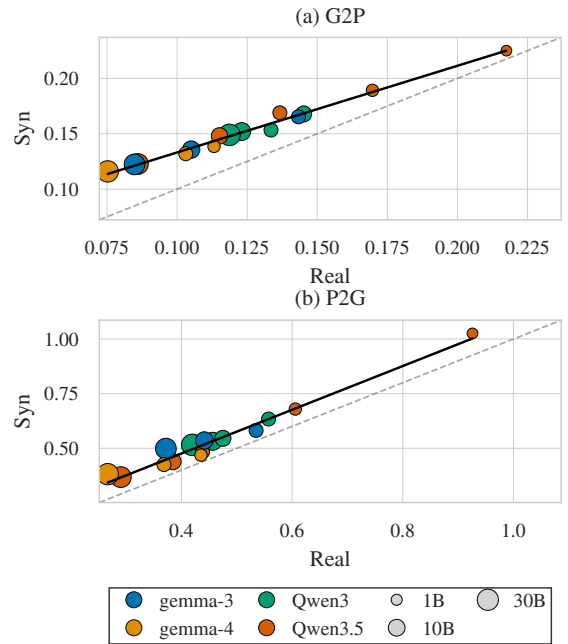


Figure 3: ZS transduction performance on real vs. synthetic data averaged across languages.

of genuine generalization. We further benchmark LLMs against CharsiuG2P on G2P (Table 4), restricted to the subset of WikiPron data not seen in its training. We report two scores: on languages supported by CharsiuG2P (left) and mPhonBench languages not supported by it (right). Three observations stand out. First, all LLMs perform better on languages supported by CharsiuG2P: this

		G2P ↓	P2G ↓	G2P _{syn} ↓	P2G _{syn} ↓	pSeg ↑	pLID ↑	pDetect ↑	PhCnt ↓	SyCnt ↓
gemma-3-27b-it	FS1	0.088	0.295	0.123	0.431	0.537 _{+ .283}	0.206 _{+ .382}	0.257 _{+ .214}	1.074	0.347
	FS5	<u>0.069</u>	0.249	0.107	0.390	0.604 _{+ .231}	0.180 _{+ .407}	0.260 _{+ .216}	0.709	0.247
	ZS+R	0.119	0.331	0.145	0.407	0.403 _{+ .358}	0.194 _{+ .331}	0.242 _{+ .209}	0.717	0.286
gemma-4-31B-it	FS1	0.071	0.226	0.109	0.342	0.643 _{+ .219}	0.187 _{+ .396}	0.323 _{+ .231}	0.786	0.149
	FS5	0.060	0.197	0.095	0.318	0.689 _{+ .185}	0.198 _{+ .375}	<u>0.332</u> _{+ .241}	0.525	0.130
	ZS+R	0.078	0.215	0.113	0.313	0.600 _{+ .248}	<u>0.275</u> _{+ .308}	0.360 _{+ .195}	0.420	0.122
Qwen3-32B	FS1	0.114	0.356	0.134	0.443	0.528 _{+ .269}	0.142 _{+ .429}	0.235 _{+ .192}	0.983	0.305
	FS5	0.096	0.316	0.116	0.415	0.597 _{+ .211}	0.143 _{+ .414}	0.243 _{+ .198}	0.701	0.249
	ZS+R	0.126	0.424	0.148	0.514	0.426 _{+ .339}	0.160 _{+ .371}	0.245 _{+ .187}	0.659	0.306
Qwen3.5-27B	FS1	0.086	0.253	0.117	0.336	0.580 _{+ .268}	0.176 _{+ .428}	0.262 _{+ .251}	0.763	0.207
	FS5	0.073	0.227	<u>0.105</u>	0.319	<u>0.655</u> _{+ .212}	0.179 _{+ .394}	0.263 _{+ .266}	0.637	0.207
	ZS+R	1.003	0.962	1.003	0.999	0.000 _{+ .079}	0.107 _{+ .496}	0.000 _{+ .018}	6.744	2.268
gpt-oss-120b	FS1	0.076	0.228	0.107	<u>0.291</u>	0.533 _{+ .272}	0.263 _{+ .314}	0.281 _{+ .177}	0.369	0.116
	FS5	0.090	<u>0.210</u>	0.116	0.278	0.542 _{+ .265}	0.252 _{+ .326}	0.280 _{+ .180}	0.432	0.132
	ZS+R	0.078	0.253	0.109	0.322	0.530 _{+ .283}	0.299 _{+ .292}	0.289 _{+ .144}	<u>0.378</u>	<u>0.120</u>

Table 2: Per-task performance of the flagship models across prompt variants, averaged over languages.

	G2P	P2G
gemma-3 (27b-it)	-0.025 _{±0.53}	-0.003 _{±0.33}
gemma-4 (31B-it)	-0.016 _{±0.16}	+0.020 _{±0.25}
Qwen3 (32B)	-0.006 _{±0.24}	+0.021 _{±0.33}
GPT-OSS (120B)	- 0.003 _{±0.12}	+ 0.024 _{±0.23}

Table 3: Absolute performance differences between ZS+R and ZS+R+C. We report mean and std. deviation ($\pm$) across all instances (from all languages).

	Supported			Unsupported		
	ZS	FS1	FS5	ZS	FS1	FS5
gemma-3	0.059	0.063	<u>0.053</u>	0.106	0.108	<u>0.082</u>
gemma-4	0.051	0.049	0.043	0.095	0.088	0.074
Qwen3	0.095	0.097	0.084	0.139	0.129	0.107
Qwen3.5	0.062	0.067	0.059	0.106	0.103	0.085
GPT-OSS	0.054	<u>0.055</u>	0.071	<u>0.097</u>	<u>0.094</u>	0.106
CHARSIUG2P	<u>0.053</u>	-	-	0.264	-	-

Table 4: G2P performance (PFER) on WikiPron data unseen by CHARSIUG2P. Left block: the 25 languages supported by CHARSIUG2P. Right block: the remaining 33 unsupported languages.

is because these are generally high(er)-resource languages with large pretraining corpora. Second, Gemma-4 (31B-it) outperforms CharsiuG2P on its own supported languages already under ZS, and widens the margin with few-shot examples. Third, CharsiuG2P shows poor generalization to languages it was not trained on: LLMs degrade in performance much more gracefully on those languages. These results suggest that modern general-purpose LLMs are at the very least competitive with dedicated (i.e., task-specialized) G2P transducers.

Language Resourceness. We next examine how a language’s general “*resourceness*” drives performance on mPhonBench. We correlate each lan-

	PR ↓	G2P ↓
Phi-4-multimodal-instruct	0.632	0.199
Qwen2.5-Omni-7B	0.474	0.333
Qwen3-Omni-30B-A3B-Instruct	0.394	0.161
gemma-4-E2B-it	0.281	0.076
gemma-4-E4B-it	<u>0.198</u>	<u>0.139</u>
POWSM	0.396	-
Wav2Vec2-XLSR	0.194	-
ZIPA	0.233	-

Table 5: Zero-shot PR and paired G2P performance.

guage’s overall benchmark performance (averaged across all tasks except pSeg and PR as a per-task z -score; details in Appendix H) against its word count in FineWeb-2 (Penedo et al., 2025) as a resourceness proxy. Figure 4a plots ZS score against FineWeb-2 word count, and Figure 4b the corresponding FS5 – ZS gain. Language resourceness exhibits a positive correlation ($r=0.52$) with ZS performance, and a *negative* correlation ($r=-0.31$) with the performance difference (i.e., gain) between FS5 and ZS. This is encouraging, as it suggests that, for phonological tasks, LLMs’ inherent representational deficit for low-resource languages can be partially offset with a few in-context examples.

Phone Recognition. Finally, Table 5 reports zero-shot performance of SLLMs and specialized baselines on speech-based word-level PR and the corresponding (text-based) G2P. The best SLLM, Gemma-4-E4B-it, matches the PR performance of the best specialized model, Wav2Vec2-XLSR, outperforming all other specialized speech-to-IPA models. Surprisingly, the smaller Gemma-4-E2B-it outperforms its larger sibling on the corresponding G2P task. The other SLLMs, of similar size or

from WikiPron (Lee et al., 2020), segmentation text from Wikipedia dumps, and audio from the Multilingual Spoken Words Corpus (Mazumder et al., 2021). Audio in MSWC is sourced from Common Voice (Ardila et al., 2020) and is distributed without speaker names or real-world identifiers. A complete inventory of third-party software, datasets, and models, along with their licenses, is provided in Appendix L; all artifacts are used in accordance with their stated terms.

7 Limitations

Probing Implicit Phonological Knowledge. Our benchmark is IPA-centric: 7 of 9 tasks require the model to generate or parse IPA strings, which probes its explicit phonological knowledge. The remaining two counting tasks can likewise be solved by first converting the input into IPA and then applying simple counting rules. This design choice enables broad language coverage and sufficient examples per language for a robust evaluation, but it does not directly probe the implicit phonological knowledge that LLMs encode in their (orthographic) token representations. Tasks that would directly probe this latent information, such as rhyme detection, cannot reliably be derived from our IPA transcriptions. Rhyming is largely a language-specific poetic convention, not a purely phonological predicate (which could be inferred from the IPA). Furthermore, for languages with transparent orthographies, the derived pairs match in their orthographic form, making the task trivial and thereby biasing the evaluation in favor of such languages. Constructing valid rhyme data would thus require language-specific manual curation beyond the scope of this work.

Pseudo Transcriptions At Scale. Our pseudo-word probes ($G2P_{syn}$, $P2G_{syn}$, and $pDetect/oDetect$) enforce no higher-order, language-specific phonotactic constraints. The synthetic transduction pairs therefore likely contain a nontrivial fraction of phonologically implausible forms, which may inflate the gap to real-form performance; for the detection tasks ($pDetect/oDetect$), this looseness is deliberate, as spotting locally well-formed but globally implausible forms is part of the task. However, performance on these tasks scales with model size, so the gap to real forms at least partly reflects incomplete generalization and not just a benchmark artifact.

Pairing Audio with Transcriptions. We create

our PR portion of the benchmark by pairing MSWC audio recordings with external phonemic transcriptions from Wiktionary. While this is a common strategy for creating large-scale multilingual phone recognition datasets (Xu et al., 2022), it comes with the obvious limitation that the actual phonetic realization of the utterance may deviate from the provided transcription. However, since all evaluated SLLMs and baselines are evaluated against the same broad transcriptions, this limitation affects relative model comparisons less than absolute performance estimates.

Reasoning Elicits Degenerate Responses. Despite the mitigations described in our inference setup, the Qwen3.5 family produced a non-trivial number of degenerate responses under both reasoning variants (ZS+R and ZS+R+C); we kept the affected runs and assigned task-specific penalty scores to preserve cross-model comparability. Notably, the breakdown affects all tasks except pLID and oLID. This and the high failure rate of GPT-OSS-20B under zero-shot prompting suggest that IPA in the input or expected output may be implicated, but we leave a systematic investigation to future work.

Using Weakly Curated Data. Our benchmark is derived from Wiktionary, a crowdsourced resource. Despite our best efforts to ensure data quality through IPA normalization and filtering (see Appendix A), we cannot guarantee uniform quality across all languages.

Potential Leakage in Segmentation Data. We source our sentences for the segmentation tasks (pSeg and its orthographic counterpart oSeg) from Wikipedia dumps and they are thus subject to the same contamination risk as our Wiktionary-derived lexical data. Two factors mitigate the impact of such leakage on our conclusions: (i) The potential leakage is limited to the orthographic segmentation task (oSeg), which serves only as a sanity check and upper-bound reference for the phonemic pSeg (Appendix D). (ii) Our preprocessing (lowercasing, removal of punctuation, and deletion of whitespace) ensures that the exact input strings never occur verbatim in pretraining corpora, so models cannot purely rely on pattern matching but must apply lexical knowledge of word forms.

8 Acknowledgements

Inference was conducted using computing resources provided by WestAI on the supercomputer

JURECA (Thörnig, 2021) at Jülich Supercomputing Centre (JSC) and on Julia 2. Julia 2 was funded as a DFG project (“Forschungsgroßgerät nach Art. 91b GG”) under INST 93/1145-1 FUGG.

This work was supported by the Alcatel-Lucent Stiftung and Deutsches Stiftungszentrum through the grant “Equitably Fair and Trustworthy Language Technology” (EQUIFAIR, Grant Nr. T0067/43110/23).

References

- Cormac Anderson, Tiago Tresoldi, Thiago Chacon, Anne-Maria Fehn, Mary Walworth, Robert Forkel, and Johann-Mattis List. 2019. [A cross-linguistic database of phonetic transcription systems](#). *Yearbook of the Poznan Linguistic Meeting*, 4:21–53.
- Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. [Common voice: A massively-multilingual speech corpus](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.
- Seth Aycock, David Stap, Di Wu, Christof Monz, and Khalil Sima'an. 2025. [Can LLMs really learn to translate a low-resource language from one grammar book?](#) In *The Thirteenth International Conference on Learning Representations*.
- Jean Berko. 1958. [The child’s learning of english morphology](#). *WORD*, 14:150–177.
- Mathieu Bernard and Hadrien Titeux. 2021. [Phonemizer: Text to phones transcription for multiple languages in python](#). *Journal of Open Source Software*, 6(68):3958.
- Shikhar Bharadwaj, Chin-Jou Li, Yoonjae Kim, Kwanghee Choi, Eunjung Yeo, Ryan Soh-Eun Shim, Hanyu Zhou, Brendon Boldt, Karen Rosero Jacome, Kalvin Chang, Darsh Agrawal, Keer Xu, Chao-Han Huck Yang, Jian Zhu, Shinji Watanabe, and David R. Mortensen. 2026. [Prism: Benchmarking phone realization in speech models](#). *Preprint*, arXiv:2601.14046.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Cheol Jun Cho, Nicholas Lee, Akshat Gupta, Dhruv Agarwal, Ethan Chen, Alan Black, and Gopala Anumanchipalli. 2025. [Sylber: Syllabic embedding representation of speech from raw audio](#). In *The Thirteenth International Conference on Learning Representations*.
- Cheol Jun Cho, Abdelrahman Mohamed, Shang-Wen Li, Alan W Black, and Gopala K. Anumanchipalli. 2024. [Sd-hubert: Sentence-level self-distillation induces syllabic organization in hubert](#). In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12076–12080.
- Kwanghee Choi, Ankita Pasad, Tomohiko Nakamura, Satoru Fukayama, Karen Livescu, and Shinji Watanabe. 2024. [Self-Supervised Speech Representations are More Phonetic than Semantic](#). In *Interspeech 2024*, pages 4578–4582.
- Simon Clematide and Peter Makarov. 2021. [CLUZH at SIGMORPHON 2021 shared task on multilingual grapheme-to-phoneme conversion: Variations on a baseline](#). In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 148–153, Online. Association for Computational Linguistics.
- G. N. Clements. 1990. *The role of the sonority cycle in core syllabification*, page 283–333. *Papers in Laboratory Phonology*. Cambridge University Press.
- Shuangrui Ding, Zihan Liu, Xiaoyi Dong, Pan Zhang, Rui Qian, Junhao Huang, Conghui He, Dahua Lin, and Jiaqi Wang. 2025. [SongComposer: A large language model for lyric and melody generation in song composition](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7108–7127, Vienna, Austria. Association for Computational Linguistics.
- Lu Dong, Zhi-Qiang Guo, Chao-Hong Tan, Ya-Jun Hu, Yuan Jiang, and Zhen-Hua Ling. 2022. [Neural grapheme-to-phoneme conversion with pre-trained grapheme models](#). In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6202–6206.
- Mariia Fedorova, Nikolay Arefyev, Maja Buljan, Jindřich Helcl, Stephan Oepen, Egil Rønningstad, and Yves Scherrer. 2026. [OpenLID-v3: Improving the precision of closely related language identification – an experience report](#). In *Proceedings of the 13th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 275–292, Rabat, Morocco. Association for Computational Linguistics.
- Zébulon Goriely and Paula Buttery. 2025. [BabyLM’s first words: Word segmentation as a phonological probing task](#). In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pages 522–539, Vienna, Austria. Association for Computational Linguistics.

- 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 *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 37–53, Miami, FL, USA. Association for Computational Linguistics.
- Dongrui Han, Mingyu Cui, Jiawen Kang, Xixin Wu, Xunying Liu, and Helen Meng. 2024. [Improving grapheme-to-phoneme conversion through in-context knowledge retrieval with large language models](#). In *2024 IEEE 14th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 631–635.
- Valentin Hofmann, Leonie Weissweiler, David R. Mortensen, Hinrich Schütze, and Janet B. Pierrehumbert. 2025. [Derivational morphology reveals analogical generalization in large language models](#). *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.
- Jonathan Hus and Antonios Anastasopoulos. 2024. [Back to school: Translation using grammar books](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20207–20219, Miami, Florida, USA. Association for Computational Linguistics.
- Fotis Jannidis, Rabea Kleymann, Julian Schröter, and Heike Zinsmeister. 2025. [Do large language models understand literature? case studies and probing experiments on german poetry](#). *Journal of Computational Literary Studies*, 4(1).
- Sittichai Jiampojarn, Grzegorz Kondrak, and Tarek Sherif. 2007. [Applying many-to-many alignments and hidden Markov models to letter-to-phoneme conversion](#). In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 372–379, Rochester, New York. Association for Computational Linguistics.
- Haeji Jung, Changdae Oh, Joeeon 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 Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pages 200–211, Miami, Florida, USA. Association for Computational Linguistics.
- Amir Hossein Kargaran, Ayyoob Imani, François Yvon, and Hinrich Schuetze. 2023. [GlotLID: Language identification for low-resource languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6155–6218, Singapore. Association for Computational Linguistics.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with pagedattention](#). In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Jackson L. Lee, Lucas F.E. Ashby, M. Elizabeth Garza, Yeonju Lee-Sikka, Sean Miller, Alan Wong, Arya D. McCarthy, and Kyle Gorman. 2020. [Massively multilingual pronunciation modeling with WikiPron](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4223–4228, Marseille, France. European Language Resources Association.
- Chin-Jou Li, Calvin Chang, Shikhar Bharadwaj, Eunjung Yeo, Kwanghee Choi, Jian Zhu, David Mortensen, and Shinji Watanabe. 2026. [Powsm: A phonetic open whisper-style speech foundation model](#). *Preprint*, arXiv:2510.24992.
- Xinjian Li, Florian Metze, David R. Mortensen, Alan W. Black, and Shinji Watanabe. 2022. [ASR2K: Speech Recognition for Around 2000 Languages without Audio](#). In *Interspeech 2022*, pages 4885–4889. ISCA.
- Disen Liao and Freda Shi. 2026. [How tokenization limits phonological knowledge representation in language models and how to improve them](#). In *Tokenization Workshop*.
- Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermy, Andy Jones, and 8 others. 2025. [On the biology of a large language model](#). *Transformer Circuits Thread*.
- Hairong Liu, Mingbo Ma, Liang Huang, Hao Xiong, and Zhongjun He. 2019. [Robust neural machine translation with joint textual and phonetic embedding](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3044–3049, Florence, Italy. Association for Computational Linguistics.
- Te Ma, Min Bi, Saierdaer Yusuyin, Hao Huang, and Zhi-jian Ou. 2025. [LLM-based phoneme-to-grapheme for phoneme-based speech recognition](#). In *Interspeech 2025*, pages 559–563. ISCA.
- 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.
- Mark Mazumder, Sharad Chitlangia, Colby Banbury, Yiping Kang, Juan Manuel Ciro, Keith Achorn, Daniel Galvez, Mark Sabini, Peter Mattson, David Kanter, Greg Diamos, Pete Warden, Josh Meyer, and Vijay Janapa Reddi. 2021. [Multilingual spoken words corpus](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

- Oliver McLaughlin, Jack Merullo, and Arjun Khurana. 2025. [I have no mouth, and i must rhyme: Uncovering internal phonetic representations in LLaMA 3.2](#). In *ICML 2025 Workshop on Assessing World Models*.
- Microsoft, :, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Junkun Chen, Weizhu Chen, Yen-Chun Chen, Yi ling Chen, Qi Dai, and 57 others. 2025. [Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras](#). Preprint, arXiv:2503.01743.
- 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).
- David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. 2016. [PanPhon: A resource for mapping IPA segments to articulatory feature vectors](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484, Osaka, Japan. The COLING 2016 Organizing Committee.
- Boris New, Jessica Bourgin, Julien Barra, and Christophe Pallier. 2024. [UniPseudo: A universal pseudoword generator](#). *Quarterly Journal of Experimental Psychology*, 77(2):278–286.
- OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, and 108 others. 2025. [gpt-oss-120b & gpt-oss-20b model card](#). Preprint, arXiv:2508.10925.
- Aitor Ormazabal, Mikel Artetxe, Manex Agirrezabal, Aitor Soroa, and Eneko Agirre. 2022. [PoeLM: A meter- and rhyme-controllable language model for unsupervised poetry generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3655–3670, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Renhao Pei, Yihong Liu, Peiqin Lin, François Yvon, and Hinrich Schuetze. 2025. [Understanding in-context machine translation for low-resource languages: A case study on Manchu](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8767–8788, Vienna, Austria. Association for Computational Linguistics.
- 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](#). In *Second Conference on Language Modeling*.
- Yifan Peng, Yui Sudo, Muhammad Shakeel, and Shinji Watanabe. 2024. [OWSM-CTC: An open encoder-only speech foundation model for speech recognition, translation, and language identification](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10192–10209, Bangkok, Thailand. Association for Computational Linguistics.
- 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, Copenhagen, Denmark. Association for Computational Linguistics.
- Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2024. [Scaling speech technology to 1,000+ languages](#). *Journal of Machine Learning Research*, 25(97):1–52.
- Qwen Team. 2026. [Qwen3.5: Towards native multimodal agents](#).
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. [Robust speech recognition via large-scale weak supervision](#). *arXiv preprint*.
- James Route, Steven Hillis, Isak Czeresnia Etinger, Han Zhang, and Alan W Black. 2019. [Multimodal, multilingual grapheme-to-phoneme conversion for low-resource languages](#). In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 192–201, Hong Kong, China. Association for Computational Linguistics.
- Jenny R. Saffran, Richard N. Aslin, and Elissa L. Newport. 1996. [Statistical learning by 8-month-old infants](#). *Science*, 274(5294):1926–1928.
- Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. 2025. [The good, the bad, and the greedy: Evaluation of LLMs should not ignore non-determinism](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4195–4206, Albuquerque, New Mexico. Association for Computational Linguistics.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 431 others. 2023. [Beyond](#)

- the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*. Featured Certification.
- Ashima Suvarna, Harshita Khandelwal, and Nanyun Peng. 2024. [PhonologyBench: Evaluating phonological skills of large language models](#). In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 1–14, Bangkok, Thailand. Association for Computational Linguistics.
- Garrett Tanzer, Mirac Suzgun, Eline Visser, Dan Jurafsky, and Luke Melas-Kyriazi. 2024. [A benchmark for learning to translate a new language from one grammar book](#). In *The Twelfth International Conference on Learning Representations*.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. [Gemma 3 technical report](#). *Preprint*, arXiv:2503.19786.
- Qwen Team. 2026. [Qwen3.5-omni technical report](#). *Preprint*, arXiv:2604.15804.
- Linh The Nguyen, Thinh Pham, and Dat Quoc Nguyen. 2023. [XPhoneBERT: A Pre-trained Multilingual Model for Phoneme Representations for Text-to-Speech](#). In *Interspeech 2023*, pages 5506–5510. ISCA.
- Philipp Thörnig. 2021. [Jureca: Data centric and booster modules implementing the modular supercomputing architecture at jülich supercomputing centre](#). *Journal of large-scale research facilities JLSRF*, 7.
- Teresa A. Ukrainetz, Janae J. Nuspl, Kimberly Wilkerson, and Sarah Rose Beddes. 2011. [The effects of syllable instruction on phonemic awareness in preschoolers](#). *Early Childhood Research Quarterly*, 26(1):50–60.
- Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. [ESPnet: End-to-end speech processing toolkit](#). In *Proceedings of Interspeech*, pages 2207–2211.
- Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kbra, Atharva Kulkarni, Abhishek Vijayakumar, Haofer Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. [Counting the bugs in ChatGPT’s wugs: A multilingual investigation into the morphological capabilities of a large language model](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6508–6524, Singapore. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. 2025a. [Qwen2.5-omni technical report](#). *Preprint*, arXiv:2503.20215.
- Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfu Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, and 19 others. 2025b. [Qwen3-omni technical report](#). *Preprint*, arXiv:2509.17765.
- Qiantong Xu, Alexei Baevski, and Michael Auli. 2022. [Simple and Effective Zero-shot Cross-lingual Phoneme Recognition](#). In *Interspeech 2022*, pages 2113–2117.
- 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.
- Chen Zhang, Xiao Liu, Jiaheng Lin, and Yansong Feng. 2024a. [Teaching large language models an unseen language on the fly](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8783–8800, Bangkok, Thailand. Association for Computational Linguistics.
- Kexun Zhang, Yee Choi, Zhenqiao Song, Taiqi He, William Yang Wang, and Lei Li. 2024b. [Hire a linguist!: Learning endangered languages in LLMs with in-context linguistic descriptions](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15654–15669, Bangkok, Thailand. Association for Computational Linguistics.
- Jian Zhu, Farhan Samir, Eleanor Chodroff, and David R. Mortensen. 2025. [ZIPA: A family of efficient models for multilingual phone recognition](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19568–19585, Vienna, Austria. Association for Computational Linguistics.
- Jian Zhu, Cong Zhang, and David Jurgens. 2022. [ByT5 model for massively multilingual grapheme-to-phoneme conversion](#). In *Interspeech 2022*, pages 446–450. ISCA.

Language	Orthographic	IPA
English (US)	bizand	/bɪzənd/
	garmenine	/gɑɪməniːn/
	manlent	/mænlənt/
	antrosive	/æntɹəsɪv/
	Carbly	/kɑɪbli/
German	Zulte	/tsuːltə/
	Quarin	/kvaxɪn/
	Alrapie	/alɾapiː/
	Hofrin	/hɔːfɪn/
	Sandach	/zantax/
Arabic	داماش	/ʃaːmaːd/
	طارنل	/lɪtaraːtˤ/
	ساسطلق	/qalʔusˤaːsa/
	ةجابت	/tabaːdʒa/
Hindi	ليحوشق	/qafuːhiːl/
	अधाना	/əd̪ʱaːnaː/
	चाकाले	/tʃʌːkaːlɛː/
	दहलान	/d̪əfɪlaːn/
Greek	बलागी	/bəlɑːgiː/
	मंगी	/məŋgiː/
	Φιλιππος	/filitiːstos/
	πεμελιν	/pemeˈlas/
	λτηκα	/altika/
	ατηλαψη	/atilaˈpsi/
	εποπιροο	/epopisoˈro/

Table 6: Examples of generated pseudo-pairs for the synthetic G2P_{syn}/ P2G_{syn} tasks.

A Preprocessing Pipeline

We apply a single filtering pass to the WikiPron tables before any task-specific data generation, gating which entries reach the downstream pipeline. We retain only broad transcriptions, since the benchmark targets phonemic IPA, and drop within-language duplicates of (grapheme, phoneme) pairs to avoid over-representing common forms. After BIPA normalization (see Section 2.2), we require each transcription to carry explicit syllable-boundary markers (.) and exclude entries where the marker-implied syllable count disagrees with the count produced by syllabifying the transcription under the Sonority Sequencing and Maximum Onset principles (as used for SyCnt in Section 2.4). At the language level, we retain only languages with at least 200 entries, ensuring that every language has sufficient material for a meaningful evaluation (and data generation), and cap the per-language test pool at 1000 to keep the inference compute manageable. Finally, we drop word forms that occur in the filtered tables of more than one language, as these are often loanwords or proper names with identical orthography and unclear phonemic form across languages, making them unreliable for language-specific evaluation.

B Pseudo Generation

B.1 G2P_{syn} & P2G_{syn}

Here we provide the implementation details of the synthetic transcription generation for G2P_{syn} and P2G_{syn} of Section 2.3:

Alignment. For each language, we train a lightweight grapheme-to-phoneme alignment model (Jiampojamarn et al., 2007) on all WikiPron transcriptions.

Position-indexed syllable inventory. We segment each aligned pair at IPA syllable boundaries and tag the resulting (grapheme, phoneme) syllable pairs by position — *initial*, *medial*, *final*, or *mono*-syllabic — retaining only pairs that occur at least twice in the source corpus.

Recombination. For each language, we sample as many synthetic pairs as there are real test items: a pseudo-word of length n is composed from one *initial*, $n - 2$ *medial*, and one *final* syllable (or one *mono* entry for $n = 1$), with syllables drawn proportional to their corpus frequency and n drawn from the language’s empirical syllable-count distribution.

Filtering and language coverage. We discard any candidate that coincides with an existing WikiPron entry and deduplicate the generated set. Languages with fewer than 50 valid (grapheme, phoneme) pairs, a monosyllabic-dominant token distribution, or an inventory too sparse to support recombination are skipped; per-language coverage is reported in Appendix M.

We provide samples of generated pairs in Table 6.

B.2 pDetect & oDetect

Next, we describe the UniPseudo-based generator used for both the IPA-side (pDetect) (Section 2.5) and orthographic-side (oDetect) pseudo-word items in detail:

n -gram estimation. We run UniPseudo (New et al., 2024) in trigram mode (seed length 3, key length 2): each candidate is seeded with a length-3 substring drawn from a real form, and every subsequent symbol is sampled from the empirical distribution of continuations after the last two symbols. Statistics are estimated from the union of the language’s WikiPron test and development splits, with the development split capped at 2,500 entries to bound the size of the n -gram tables.

Sampling. Generation is length-stratified: for each attested form length, we sample candidates in proportion to its frequency in the source pool, so the

pseudo length distribution mirrors the attested one. Lengths with fewer than five source words are excluded, since their n -gram tables would be dominated by a single form.

Filtering. We discard any candidate that (i) appears in our WikiPron data, (ii) coincides with one of the source words used to estimate its n -gram statistics, or (iii) duplicates a previously generated pseudo; the global-index check catches loanword-style overlaps that a same-language filter alone would miss.

C pSeg Phonemizer Variants

We report the IPA word-segmentation results for all four variants of the pSeg task, comparing the two phonemizers (Epitrans vs. eSpeak) and the two IPA inventories (raw vs. standardized) in Table 7. We chose to report scores for the best variant (eSpeak-std) in the main body, but because the margin between Epitrans and eSpeak-std is small, we include all four variants in the benchmark release.

D Orthographic Task Variants

We additionally evaluate orthographic counterparts to three of our IPA-side tasks: pseudoword detection (oDetect), language identification (oLID), and word segmentation (oSeg). Each variant reuses its phonemic counterpart’s pipeline verbatim but takes the orthographic word form as input rather than the IPA transcription.

oDetect: Detecting Pseudo-Words. Each item contains five orthographic word forms in a shuffled order: four attested and one distractor, generated by UniPseudo on the grapheme side (mirroring the IPA-side construction in Section 2.5); the model returns the index of the distractor.

oLID: Orthographic LID. Same setup as pLID (Section 2.6) but with three orthographic word forms as input rather than IPA transcriptions. We apply the same cross-language overlap filter in all our WikiPron languages.

oSeg: Segmentation of Unspaced Text. We use the same Wikipedia sentences as for pSeg. To avoid leaking word-boundary cues through punctuation or capitalization, we lowercase the sentence and strip punctuation before concatenating the words into a single unspaced string.

Table 8 reports performance on the three orthographic variants for the flagship cohort, with a subscript indicating the gain over the matching phone-

mic task.

E Baseline Inference Configurations

Model	Checkpoint
CHARSIUG2P	charsiu/g2p_multilingual_byT5_small_100
WAV2VEC2-XLSR	facebook/wav2vec2-xlsr-53-espeak-cv-ft
ZIPA	anyspeech/zipa-large-crctc-ns-800k
POWSM	espnet/powsm

Table 9: Hugging Face checkpoints used for the G2P and PR baselines.

We report the exact Hugging Face Hub Repo IDs for all baselines in Table 9.

G2P. We use the largest available CHARSIUG2P model via the transformers framework (Wolf et al., 2020). Following the authors’ methodology, we use greedy decoding and set the language code to unk for unseen languages.

PR. We run WAV2VEC2-XLSR via the transformers implementation, ZIPA via the ONNX implementation released by the authors, and POWSM via the ESPnet2 inference frontend (Watanabe et al., 2018). We use greedy decoding for WAV2VEC2-XLSR and ZIPA; for POWSM, we set the task token to <pr> and use beam search with beam_size=5.

F Computational Resources

All experiments were conducted on NVIDIA H100 GPUs. We estimate the total compute footprint of this work at approximately 2,500 GPU-hours, including pilot runs, development, and preliminary analyses.

G Failure Rates

We report the zero-shot failure rates for all models and tasks in Table 10.

H Details on the Benchmark Aggregate Score

Benchmark tasks report scores in heterogeneous units (PFER, CER, F_1 , MAE, accuracy), so directly averaging them into a per-(model, language) summary would let the metric with the largest natural range dominate. For the resourceness regression (Figure 4a), we therefore standardize each task’s per-language scores before cross-task aggregation, producing a single z -aggregated proxy score per (model, prompt-variant, language). For every task,

	Epitran	Epitran-std	eSpeak	eSpeak-std $\star$
gemma-3-4b-it	0.188	0.183	0.129	0.186
gemma-3-12b-it	<u>0.326</u>	0.323	0.171	0.329
gemma-3-27b-it	<u>0.397</u>	0.388	0.249	0.407
gemma-4-E2B-it	0.026	<u>0.025</u>	0.016	0.021
gemma-4-E4B-it	0.241	<u>0.232</u>	0.102	0.239
gemma-4-26B-A4B-it	0.436	0.429	0.339	<u>0.434</u>
gemma-4-31B-it	<u>0.583</u>	0.566	0.487	0.585
Qwen3-4B	0.260	<u>0.265</u>	0.159	0.269
Qwen3-8B	0.364	<u>0.357</u>	0.255	0.364
Qwen3-14B	0.411	0.404	0.283	0.416
Qwen3-30B-A3B	<u>0.379</u>	<u>0.379</u>	0.331	0.390
Qwen3-32B	<u>0.424</u>	0.415	0.289	0.434
Qwen3.5-0.8B	0.007	<u>0.005</u>	0.003	<u>0.005</u>
Qwen3.5-2B	<u>0.089</u>	<u>0.089</u>	0.028	0.094
Qwen3.5-4B	0.123	<u>0.128</u>	0.060	0.141
Qwen3.5-9B	0.236	<u>0.239</u>	0.170	0.253
Qwen3.5-27B	<u>0.488</u>	0.481	0.410	0.510
Qwen3.5-35B-A3B	<u>0.339</u>	0.337	0.283	0.351
gpt-oss-20b	0.214	<u>0.215</u>	0.149	0.222
gpt-oss-120b	0.526	0.520	0.318	<u>0.524</u>
Average	<u>0.303</u>	0.299	0.212	0.309

Table 7: IPA word-segmentation F_1 per variant, macro-averaged over languages in the zero-shot regime (ZS). Columns: Epitran vs. eSpeak, raw IPA vs. BIPA-standardized IPA. The variant marked $\star$ is the one used in the main evaluation.

we first collapse example-level metrics to a single per-model, prompt-variant, language score. We then standardize these scores against the cohort, $z = (x - \mu)/\sigma$, and sign-flip lower-is-better metrics so that positive z means “better than the typical (model, prompt-variant, language) triple”. We collapse near-identical task variants (e.g. real and synthetic G2P) onto the same families used for the headline mPhonBench score, take a per-language mean within each family, and average over the six families with lexical coverage (G2P, P2G, pLID, PhCnt, SyCnt, pDetect); we omit pSeg and PR due to their differing language coverage.

I Detailed Results: PR

Per-Language Results. We report per-language PFER for the PR task in Table 11.

Per-Prompt-Variant Results. Table 12 reports per-prompt-variant PFER for PR. In stark contrast to the text-side trend, few-shot prompting and reasoning consistently *degrade* performance relative to zero-shot, often dramatically (e.g., gemma-4-E2B-it jumps from 0.281 to 1.394 under FS5). The exception is gemma-4-E4B-it, which shows an inconsistent trajectory: it degrades slightly under FS1 but improves under FS5, outperforming the zero-shot baseline. Reasoning has only a marginal

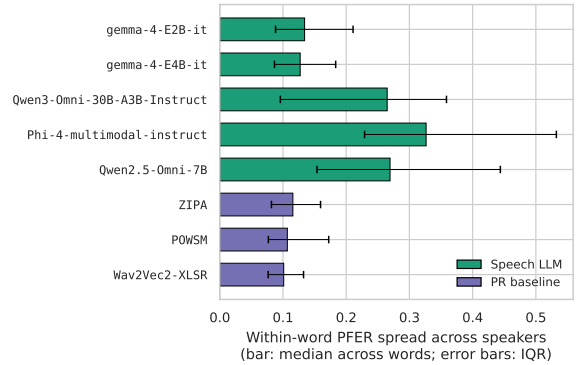


Figure 5: Within-word PR PFER spread across speakers under ZS prompting. Bars show the median across words; error bars show the inter-quartile range (IQR). Lower spread indicates more consistent performance across speakers of the same word.

effect without additional context (ZS+R) and degrades performance substantially with Wikipedia context (ZS+R+C). Taken together, these results suggest that the ICL benefits observed on the text side do not transfer cleanly to the audio-side PR, and that current SLLMs cannot reliably leverage in-context examples or reasoning chains for phonemic transcription.

Speaker Variance. Figure 5 reports across-speaker variance on the PR test set. All expert

		oLID $\Delta_{\text{pLID}} \uparrow$	oDetect $\Delta_{\text{pDetect}} \uparrow$	oSeg $\Delta_{\text{pSeg}} \uparrow$
gemma-3-27b-it	ZS	0.574 _{+0.383}	0.445 _{+0.193}	0.792 _{+0.385}
	FS1	0.588 _{+0.382}	0.470 _{+0.214}	0.820 _{+0.283}
	FS5	0.587 _{+0.407}	0.475 _{+0.216}	0.835 _{+0.231}
	ZS+R	0.525 _{+0.331}	0.450 _{+0.209}	0.761 _{+0.358}
gemma-4-31B-it	ZS	0.564 _{+0.342}	0.536 _{+0.224}	0.842 _{+0.257}
	FS1	0.583 _{+0.396}	0.552 _{+0.231}	0.862 _{+0.219}
	FS5	0.574 _{+0.375}	0.572 _{+0.241}	0.874 _{+0.185}
	ZS+R	0.583 _{+0.308}	<u>0.553</u> _{+0.195}	0.849 _{+0.248}
Qwen3-32B	ZS	0.554 _{+0.408}	0.389 _{+0.168}	0.772 _{+0.338}
	FS1	0.571 _{+0.429}	0.426 _{+0.192}	0.796 _{+0.269}
	FS5	0.557 _{+0.414}	0.440 _{+0.198}	0.808 _{+0.211}
	ZS+R	0.532 _{+0.371}	0.431 _{+0.187}	0.765 _{+0.339}
Qwen3.5-27B	ZS	0.608 _{+0.394}	0.496 _{+0.234}	0.836 _{+0.325}
	FS1	<u>0.604</u> _{+0.428}	0.512 _{+0.251}	0.847 _{+0.268}
	FS5	0.573 _{+0.394}	0.527 _{+0.266}	<u>0.867</u> _{+0.212}
	ZS+R	0.603 _{+0.496}	0.018 _{+0.018}	0.079 _{+0.079}
gpt-oss-120b	ZS	0.585 _{+0.300}	0.452 _{+0.166}	0.809 _{+0.285}
	FS1	0.577 _{+0.314}	0.458 _{+0.177}	0.805 _{+0.272}
	FS5	0.578 _{+0.326}	0.460 _{+0.180}	0.808 _{+0.265}
	ZS+R	0.591 _{+0.292}	0.433 _{+0.144}	0.813 _{+0.283}

Table 8: Performance on the orthographic task variants (oLID, oDetect, oSeg) for the largest dense model of each flagship family, macro-averaged over languages. The subscript reports the gain Δ over the matching phonemic task (pLID, pDetect, pSeg), computed on languages that contribute to both tasks. A positive Δ means the orthographic baseline is the easier task.

baselines show consistently lower variance than the SLLMs, suggesting that SLLM performance is more sensitive to speaker characteristics.

	G2P	P2G	G2P _{syn}	P2G _{syn}	pSeg	pLID	PhCnt	SyCnt	pDetect	oSeg	oLID	oDetect	<i>Mean</i>
gemma-3-4b-it	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
gemma-3-12b-it	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
gemma-3-27b-it	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
gemma-4-E2B-it	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
gemma-4-E4B-it	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
gemma-4-26B-A4B-it	0.0	0.3	0.1	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1
gemma-4-31B-it	0.0	0.2	0.1	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.1
Qwen3-4B	0.0	0.6	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
Qwen3-8B	1.8	0.4	1.8	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5
Qwen3-14B	0.8	0.0	0.7	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2
Qwen3-30B-A3B	0.2	0.1	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
Qwen3-32B	0.1	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen3.5-0.8B	2.3	1.4	2.4	1.3	0.3	0.0	17.1	17.9	1.5	0.1	0.2	23.9	4.9
Qwen3.5-2B	0.6	1.5	0.6	2.2	0.4	0.0	0.0	0.0	0.0	0.1	0.0	0.6	0.6
Qwen3.5-4B	0.3	0.0	0.3	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
Qwen3.5-9B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen3.5-27B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen3.5-35B-A3B	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
gpt-oss-20b	18.7	24.9	18.8	27.5	43.6	72.3	7.7	4.5	48.1	37.6	24.5	18.5	29.6
gpt-oss-120b	0.0	0.0	0.0	0.0	1.9	0.8	0.0	0.0	0.1	0.1	0.2	0.0	0.3
<i>Average</i>	1.2	1.5	1.3	1.7	2.3	3.7	1.2	1.1	2.5	1.9	1.2	2.2	1.8

Table 10: Per-model and -task failure rate (%) under ZS-prompting.

	Phi-4-multimodal-instruct	Qwen2.5-Omni-7B	Qwen3-Omni-30B-A3B-Instruct	gemma-4-E2B-it	gemma-4-E4B-it	POWSM	Wav2Vec2-XLSR	ZIPA
<i>Phone Recognition (PR) ↓</i>								
ara_arab	0.673 $\pm$ 0.18	0.666 $\pm$ 0.17	0.302 $\pm$ 0.04	0.207 $\pm$ 0.03	0.250 $\pm$ 0.04	0.296 $\pm$ 0.04	0.259 $\pm$ 0.02	<u>0.249</u> $\pm$ 0.02
asm_beng	0.509 $\pm$ 0.06	0.741 $\pm$ 0.14	1.813 $\pm$ 0.86	0.537 $\pm$ 0.02	<u>0.389</u> $\pm$ 0.10	2.282 $\pm$ 1.03	0.343 $\pm$ 0.08	0.426 $\pm$ 0.03
deu_latn	0.806 $\pm$ 0.04	0.349 $\pm$ 0.02	0.240 $\pm$ 0.01	<u>0.170</u> $\pm$ 0.01	0.185 $\pm$ 0.01	0.194 $\pm$ 0.01	0.162 $\pm$ 0.00	0.185 $\pm$ 0.01
ell_grek	0.658 $\pm$ 0.08	0.778 $\pm$ 0.09	0.574 $\pm$ 0.17	0.531 $\pm$ 0.05	<u>0.224</u> $\pm$ 0.04	0.302 $\pm$ 0.11	0.163 $\pm$ 0.02	0.267 $\pm$ 0.03
eng_latn_uk	0.499 $\pm$ 0.02	0.208 $\pm$ 0.01	0.189 $\pm$ 0.01	0.158 $\pm$ 0.01	0.168 $\pm$ 0.01	0.172 $\pm$ 0.01	<u>0.155</u> $\pm$ 0.00	0.153 $\pm$ 0.00
eng_latn_us	0.572 $\pm$ 0.03	0.197 $\pm$ 0.01	0.210 $\pm$ 0.01	0.167 $\pm$ 0.01	0.166 $\pm$ 0.01	0.172 $\pm$ 0.01	<u>0.155</u> $\pm$ 0.00	0.153 $\pm$ 0.00
fas_arab	0.634 $\pm$ 0.03	0.579 $\pm$ 0.03	0.424 $\pm$ 0.03	0.248 $\pm$ 0.01	<u>0.223</u> $\pm$ 0.01	0.397 $\pm$ 0.02	0.200 $\pm$ 0.00	0.259 $\pm$ 0.01
fra_latn	0.545 $\pm$ 0.03	0.312 $\pm$ 0.03	0.793 $\pm$ 0.02	0.163 $\pm$ 0.01	<u>0.171</u> $\pm$ 0.01	0.183 $\pm$ 0.02	0.195 $\pm$ 0.01	0.186 $\pm$ 0.01
hau_latn	0.798 $\pm$ 0.10	0.777 $\pm$ 0.09	0.364 $\pm$ 0.10	0.567 $\pm$ 0.12	0.197 $\pm$ 0.04	1.112 $\pm$ 0.50	<u>0.216</u> $\pm$ 0.03	0.365 $\pm$ 0.06
ind_latn	0.501 $\pm$ 0.06	0.407 $\pm$ 0.05	0.216 $\pm$ 0.03	<u>0.136</u> $\pm$ 0.02	0.126 $\pm$ 0.02	0.276 $\pm$ 0.04	0.234 $\pm$ 0.02	0.225 $\pm$ 0.02
ita_latn	0.475 $\pm$ 0.03	0.264 $\pm$ 0.02	0.165 $\pm$ 0.01	0.138 $\pm$ 0.01	0.096 $\pm$ 0.01	<u>0.108</u> $\pm$ 0.01	0.128 $\pm$ 0.00	0.155 $\pm$ 0.01
lit_latn	0.797 $\pm$ 0.13	0.442 $\pm$ 0.06	<u>0.179</u> $\pm$ 0.07	0.471 $\pm$ 0.09	0.195 $\pm$ 0.03	0.213 $\pm$ 0.03	0.169 $\pm$ 0.03	0.262 $\pm$ 0.03
mlt_latn	0.634 $\pm$ 0.04	0.652 $\pm$ 0.13	0.270 $\pm$ 0.03	0.591 $\pm$ 0.04	0.288 $\pm$ 0.05	0.553 $\pm$ 0.11	0.200 $\pm$ 0.01	<u>0.242</u> $\pm$ 0.02
nld_latn	0.626 $\pm$ 0.07	0.366 $\pm$ 0.03	0.340 $\pm$ 0.04	<u>0.162</u> $\pm$ 0.01	0.145 $\pm$ 0.01	<u>0.162</u> $\pm$ 0.01	0.174 $\pm$ 0.01	0.183 $\pm$ 0.01
pol_latn	0.705 $\pm$ 0.04	0.472 $\pm$ 0.05	0.418 $\pm$ 0.04	0.263 $\pm$ 0.01	0.283 $\pm$ 0.01	<u>0.212</u> $\pm$ 0.02	0.207 $\pm$ 0.01	0.240 $\pm$ 0.01
por_latn_bz	0.662 $\pm$ 0.16	0.380 $\pm$ 0.09	0.210 $\pm$ 0.02	0.172 $\pm$ 0.02	<u>0.176</u> $\pm$ 0.02	0.242 $\pm$ 0.03	0.200 $\pm$ 0.01	0.239 $\pm$ 0.01
por_latn_po	0.682 $\pm$ 0.10	0.340 $\pm$ 0.03	0.233 $\pm$ 0.02	0.173 $\pm$ 0.02	<u>0.190</u> $\pm$ 0.02	0.242 $\pm$ 0.03	0.200 $\pm$ 0.01	0.239 $\pm$ 0.01
ron_latn	0.634 $\pm$ 0.16	0.487 $\pm$ 0.07	0.285 $\pm$ 0.04	0.246 $\pm$ 0.03	<u>0.147</u> $\pm$ 0.03	0.180 $\pm$ 0.05	0.144 $\pm$ 0.01	0.207 $\pm$ 0.02
tur_latn	0.592 $\pm$ 0.07	0.583 $\pm$ 0.04	0.255 $\pm$ 0.02	0.230 $\pm$ 0.02	0.150 $\pm$ 0.01	0.221 $\pm$ 0.02	0.192 $\pm$ 0.01	<u>0.185</u> $\pm$ 0.01
<i>Grapheme-to-Phoneme (G2P, paired) ↓</i>								
ara_arab	0.328 $\pm$ 0.03	0.487 $\pm$ 0.05	0.181 $\pm$ 0.01	0.117 $\pm$ 0.01	<u>0.126</u> $\pm$ 0.08	—	—	—
asm_beng	0.148 $\pm$ 0.02	0.373 $\pm$ 0.08	0.115 $\pm$ 0.00	<u>0.040</u> $\pm$ 0.00	0.031 $\pm$ 0.00	—	—	—
deu_latn	0.236 $\pm$ 0.02	0.293 $\pm$ 0.02	0.103 $\pm$ 0.01	<u>0.109</u> $\pm$ 0.01	0.133 $\pm$ 0.02	—	—	—
ell_grek	0.128 $\pm$ 0.04	0.285 $\pm$ 0.05	<u>0.037</u> $\pm$ 0.01	0.087 $\pm$ 0.02	0.028 $\pm$ 0.00	—	—	—
eng_latn_uk	0.250 $\pm$ 0.01	0.219 $\pm$ 0.01	0.100 $\pm$ 0.01	<u>0.070</u> $\pm$ 0.00	0.066 $\pm$ 0.00	—	—	—
eng_latn_us	0.238 $\pm$ 0.01	0.234 $\pm$ 0.01	0.153 $\pm$ 0.01	<u>0.072</u> $\pm$ 0.00	0.069 $\pm$ 0.00	—	—	—
fas_arab	0.223 $\pm$ 0.01	0.413 $\pm$ 0.03	<u>0.125</u> $\pm$ 0.00	0.111 $\pm$ 0.00	0.588 $\pm$ 0.02	—	—	—
fra_latn	0.170 $\pm$ 0.01	0.355 $\pm$ 0.03	1.000 $\pm$ 0.00	<u>0.053</u> $\pm$ 0.01	0.051 $\pm$ 0.00	—	—	—
hau_latn	0.144 $\pm$ 0.04	0.229 $\pm$ 0.10	0.140 $\pm$ 0.02	<u>0.009</u> $\pm$ 0.00	0.008 $\pm$ 0.02	—	—	—
ind_latn	0.177 $\pm$ 0.02	0.266 $\pm$ 0.04	0.093 $\pm$ 0.02	0.064 $\pm$ 0.01	<u>0.073</u> $\pm$ 0.01	—	—	—
ita_latn	0.107 $\pm$ 0.01	0.200 $\pm$ 0.01	<u>0.068</u> $\pm$ 0.01	0.040 $\pm$ 0.01	0.072 $\pm$ 0.02	—	—	—
lit_latn	<u>0.080</u> $\pm$ 0.02	0.429 $\pm$ 0.65	0.085 $\pm$ 0.07	0.045 $\pm$ 0.00	0.083 $\pm$ 0.04	—	—	—
mlt_latn	0.207 $\pm$ 0.02	0.467 $\pm$ 0.06	<u>0.186</u> $\pm$ 0.03	0.108 $\pm$ 0.01	0.195 $\pm$ 0.04	—	—	—
nld_latn	0.202 $\pm$ 0.02	0.319 $\pm$ 0.03	0.094 $\pm$ 0.01	0.068 $\pm$ 0.00	<u>0.093</u> $\pm$ 0.02	—	—	—
pol_latn	0.243 $\pm$ 0.01	0.325 $\pm$ 0.03	<u>0.144</u> $\pm$ 0.01	0.125 $\pm$ 0.02	0.304 $\pm$ 0.05	—	—	—
por_latn_bz	0.268 $\pm$ 0.04	0.451 $\pm$ 0.10	<u>0.134</u> $\pm$ 0.02	0.087 $\pm$ 0.02	0.211 $\pm$ 0.04	—	—	—
por_latn_po	0.290 $\pm$ 0.03	0.366 $\pm$ 0.04	0.099 $\pm$ 0.01	<u>0.111</u> $\pm$ 0.02	0.213 $\pm$ 0.09	—	—	—
ron_latn	0.168 $\pm$ 0.02	0.203 $\pm$ 0.02	<u>0.100</u> $\pm$ 0.03	0.066 $\pm$ 0.01	0.177 $\pm$ 0.06	—	—	—
tur_latn	0.180 $\pm$ 0.02	0.415 $\pm$ 0.08	<u>0.111</u> $\pm$ 0.01	0.063 $\pm$ 0.01	0.119 $\pm$ 0.05	—	—	—

Table 11: Per-language PFER (lower is better) for PR. Top block: phoneme recognition. Bottom block: paired G2P. Subscript denotes the Standard Error of the Mean across speakers.

	ZS ↓	FS1 ↓	FS5 ↓	ZS+R ↓	ZS+R+C ↓
Phi-4-multimodal-instruct	0.632	1.251	1.371	1.476	1.657
Qwen2.5-Omni-7B	0.474	0.570	0.413	1.133	1.328
Qwen3-Omni-30B-A3B-Instruct	0.394	0.502	<u>0.388</u>	1.000	1.000
gemma-4-E2B-it	0.281	<u>0.247</u>	1.394	<u>0.371</u>	<u>0.464</u>
gemma-4-E4B-it	<u>0.198</u>	0.210	0.170	0.196	0.268
ZIPA	0.233	—	—	—	—
POWSM	0.396	—	—	—	—
Wav2Vec2-XLSR	0.194	—	—	—	—

Table 12: Per-prompt-variant PFER for PR, macro-averaged across languages (lower is better).

J Detailed Results: Text-only Tasks

	ZS ↓	FS1 ↓	FS5 ↓	ZS+R ↓	ZS+R+C ↓
gemma-3-4b-it	0.143	0.142	0.131	0.200	0.330
gemma-3-12b-it	0.105	0.097	0.083	0.127	0.156
gemma-3-27b-it	0.085	0.088	<u>0.069</u>	0.119	0.138
gemma-4-E2B-it	0.113	0.121	0.111	0.125	0.212
gemma-4-E4B-it	0.103	0.102	0.094	0.106	0.136
gemma-4-26B-A4B-it	0.085	0.080	<u>0.069</u>	<u>0.086</u>	<u>0.089</u>
gemma-4-31B-it	0.075	0.071	0.060	0.078	<u>0.089</u>
Qwen3-4B	0.134	0.154	0.142	0.163	0.159
Qwen3-8B	0.145	0.134	0.115	0.163	0.165
Qwen3-14B	0.123	0.119	0.102	0.121	0.145
Qwen3-30B-A3B	0.112	0.116	0.108	0.157	0.142
Qwen3-32B	0.119	0.114	0.096	0.126	0.132
Qwen3.5-0.8B	0.217	0.235	0.212	0.300	0.264
Qwen3.5-2B	0.170	0.187	0.165	0.335	0.575
Qwen3.5-4B	0.137	0.125	0.113	1.000	1.002
Qwen3.5-9B	0.115	0.102	0.091	1.001	1.001
Qwen3.5-27B	0.086	0.086	0.073	1.003	1.001
Qwen3.5-35B-A3B	0.086	0.086	0.077	1.003	1.001
gpt-oss-20b	0.260	0.196	0.202	0.309	0.356
gpt-oss-120b	<u>0.077</u>	<u>0.076</u>	0.090	0.078	0.075

Table 13: Per-prompt-variant PFER for G2P, macro-averaged across languages (lower is better).

	ZS ↓	FS1 ↓	FS5 ↓	ZS+R ↓	ZS+R+C ↓
gemma-3-4b-it	0.535	0.487	0.433	0.557	0.592
gemma-3-12b-it	0.441	0.362	0.318	0.416	0.401
gemma-3-27b-it	0.372	0.295	0.249	0.331	0.326
gemma-4-E2B-it	0.435	0.396	0.357	0.556	0.656
gemma-4-E4B-it	0.368	0.333	0.289	0.393	0.449
gemma-4-26B-A4B-it	0.319	0.286	0.257	0.270	0.261
gemma-4-31B-it	<u>0.267</u>	0.226	0.197	0.215	0.189
Qwen3-4B	0.557	0.472	0.430	0.512	0.474
Qwen3-8B	0.475	0.408	0.371	0.501	0.426
Qwen3-14B	0.457	0.375	0.331	0.469	0.464
Qwen3-30B-A3B	0.440	0.384	0.342	0.394	0.386
Qwen3-32B	0.420	0.356	0.316	0.424	0.397
Qwen3.5-0.8B	0.926	0.603	0.563	0.671	0.765
Qwen3.5-2B	0.606	0.530	0.473	0.632	0.697
Qwen3.5-4B	0.438	0.383	0.353	1.000	1.002
Qwen3.5-9B	0.385	0.330	0.296	0.982	1.004
Qwen3.5-27B	0.290	0.253	0.227	0.962	1.015
Qwen3.5-35B-A3B	0.312	0.279	0.252	0.966	1.010
gpt-oss-20b	0.475	0.453	0.478	0.499	0.509
gpt-oss-120b	0.231	<u>0.228</u>	<u>0.210</u>	<u>0.253</u>	<u>0.220</u>

Table 14: Per-prompt-variant CER for P2G, macro-averaged across languages (lower is better).

	ZS ↓	FS1 ↓	FS5 ↓	ZS+R ↓	ZS+R+C ↓
gemma-3-4b-it	0.165	0.158	0.147	0.213	0.347
gemma-3-12b-it	0.136	0.124	0.111	0.145	0.186
gemma-3-27b-it	0.122	0.123	0.107	0.145	0.163
gemma-4-E2B-it	0.139	0.144	0.131	0.149	0.251
gemma-4-E4B-it	0.132	0.128	0.116	0.130	0.159
gemma-4-26B-A4B-it	0.126	0.194	0.106	0.119	0.121
gemma-4-31B-it	<u>0.116</u>	<u>0.109</u>	0.095	<u>0.113</u>	<u>0.116</u>
Qwen3-4B	0.153	0.162	0.149	0.168	0.171
Qwen3-8B	0.168	0.150	0.130	0.174	0.173
Qwen3-14B	0.152	0.138	0.122	0.151	0.157
Qwen3-30B-A3B	0.138	0.130	0.119	0.165	0.161
Qwen3-32B	0.149	0.134	0.116	0.148	0.159
Qwen3.5-0.8B	0.225	0.234	0.214	0.303	0.268
Qwen3.5-2B	0.189	0.195	0.170	0.334	0.558
Qwen3.5-4B	0.169	0.146	0.131	1.001	1.002
Qwen3.5-9B	0.148	0.126	0.114	1.000	1.003
Qwen3.5-27B	0.123	0.117	<u>0.105</u>	1.003	1.003
Qwen3.5-35B-A3B	0.125	0.123	0.109	1.001	1.001
gpt-oss-20b	0.279	0.218	0.229	0.317	0.361
gpt-oss-120b	0.108	0.107	0.116	0.109	0.104

Table 15: Per-prompt-variant PFER for G2P_{syn}, macro-averaged across languages (lower is better).

	ZS ↓	FS1 ↓	FS5 ↓	ZS+R ↓	ZS+R+C ↓
gemma-3-4b-it	0.582	0.542	0.498	0.593	0.678
gemma-3-12b-it	0.536	0.466	0.425	0.486	0.469
gemma-3-27b-it	0.498	0.431	0.390	0.407	0.391
gemma-4-E2B-it	0.468	0.442	0.401	0.600	0.736
gemma-4-E4B-it	0.425	0.393	0.352	0.455	0.540
gemma-4-26B-A4B-it	0.417	0.386	0.366	0.374	0.344
gemma-4-31B-it	0.381	0.342	<u>0.318</u>	0.313	0.263
Qwen3-4B	0.634	0.552	0.521	0.557	0.526
Qwen3-8B	0.545	0.492	0.460	0.552	0.480
Qwen3-14B	0.532	0.456	0.414	0.547	0.538
Qwen3-30B-A3B	0.512	0.465	0.428	0.437	0.433
Qwen3-32B	0.516	0.443	0.415	0.514	0.465
Qwen3.5-0.8B	1.026	0.638	0.609	0.714	0.810
Qwen3.5-2B	0.679	0.613	0.535	0.681	0.749
Qwen3.5-4B	0.483	0.447	0.417	1.000	1.003
Qwen3.5-9B	0.439	0.393	0.369	0.998	1.010
Qwen3.5-27B	<u>0.368</u>	<u>0.336</u>	0.319	0.999	1.010
Qwen3.5-35B-A3B	0.393	0.370	0.351	0.998	1.012
gpt-oss-20b	0.540	0.517	0.542	0.557	0.571
gpt-oss-120b	0.295	0.291	0.278	<u>0.322</u>	<u>0.283</u>

Table 16: Per-prompt-variant CER for P2G_{syn}, macro-averaged across languages (lower is better).

	ZS $\uparrow$	FS1 $\uparrow$	FS5 $\uparrow$	ZS+R $\uparrow$
gemma-3-4b-it	0.186	0.336	0.418	0.189
gemma-3-12b-it	0.329	0.447	0.530	0.348
gemma-3-27b-it	0.407	0.537	0.604	0.403
gemma-4-E2B-it	0.021	0.147	0.295	0.277
gemma-4-E4B-it	0.239	0.386	0.466	0.351
gemma-4-26B-A4B-it	0.434	0.488	0.520	0.332
gemma-4-31B-it	0.585	0.643	0.689	0.600
Qwen3-4B	0.269	0.385	0.466	0.227
Qwen3-8B	0.364	0.448	0.527	0.314
Qwen3-14B	0.416	0.473	0.565	0.395
Qwen3-30B-A3B	0.390	0.469	0.534	0.323
Qwen3-32B	0.434	0.528	0.597	0.426
Qwen3.5-0.8B	0.005	0.054	0.155	0.005
Qwen3.5-2B	0.094	0.171	0.248	0.112
Qwen3.5-4B	0.141	0.269	0.417	0.000
Qwen3.5-9B	0.253	0.412	0.498	0.000
Qwen3.5-27B	0.510	<u>0.580</u>	<u>0.655</u>	0.000
Qwen3.5-35B-A3B	0.351	0.462	0.549	0.000
gpt-oss-20b	0.222	0.303	0.307	0.189
gpt-oss-120b	<u>0.524</u>	0.533	0.542	<u>0.530</u>

Table 17: Per-prompt-variant Word Boundary F_1 for pSeg (eSpeak-std), macro-averaged across languages (higher is better).

	ZS $\uparrow$	FS1 $\uparrow$	FS5 $\uparrow$	ZS+R $\uparrow$
gemma-3-4b-it	0.114	0.110	0.045	0.109
gemma-3-12b-it	0.132	0.127	0.084	0.162
gemma-3-27b-it	0.191	<u>0.206</u>	0.180	0.194
gemma-4-E2B-it	0.099	0.086	0.078	0.097
gemma-4-E4B-it	0.130	0.115	0.098	0.134
gemma-4-26B-A4B-it	0.173	0.164	0.143	0.183
gemma-4-31B-it	<u>0.223</u>	0.187	<u>0.198</u>	<u>0.275</u>
Qwen3-4B	0.111	0.074	0.039	0.098
Qwen3-8B	0.128	0.118	0.092	0.118
Qwen3-14B	0.149	0.135	0.118	0.142
Qwen3-30B-A3B	0.136	0.129	0.132	0.154
Qwen3-32B	0.146	0.142	0.143	0.160
Qwen3.5-0.8B	0.049	0.004	0.006	0.040
Qwen3.5-2B	0.066	0.051	0.056	0.052
Qwen3.5-4B	0.098	0.057	0.064	0.068
Qwen3.5-9B	0.129	0.106	0.094	0.067
Qwen3.5-27B	0.214	0.176	0.179	0.107
Qwen3.5-35B-A3B	0.208	0.161	0.173	0.055
gpt-oss-20b	0.070	0.069	0.058	0.057
gpt-oss-120b	0.285	0.263	0.252	0.299

Table 18: Per-prompt-variant Accuracy for pLID, macro-averaged across languages (higher is better).

	ZS $\uparrow$	FS1 $\uparrow$	FS5 $\uparrow$	ZS+R $\uparrow$
gemma-3-4b-it	0.215	0.207	0.193	0.212
gemma-3-12b-it	0.222	0.233	0.241	0.241
gemma-3-27b-it	0.253	0.257	0.260	0.242
gemma-4-E2B-it	0.223	0.209	0.210	0.213
gemma-4-E4B-it	0.226	0.229	0.219	0.230
gemma-4-26B-A4B-it	0.252	0.259	0.269	0.275
gemma-4-31B-it	0.314	0.323	0.332	0.360
Qwen3-4B	0.201	0.202	0.199	0.208
Qwen3-8B	0.212	0.212	0.218	0.220
Qwen3-14B	0.216	0.221	0.227	0.224
Qwen3-30B-A3B	0.195	0.197	0.200	0.217
Qwen3-32B	0.222	0.235	0.243	0.245
Qwen3.5-0.8B	0.210	0.215	0.214	0.110
Qwen3.5-2B	0.205	0.210	0.201	0.161
Qwen3.5-4B	0.209	0.207	0.205	0.000
Qwen3.5-9B	0.216	0.220	0.210	0.000
Qwen3.5-27B	0.263	0.262	0.263	0.000
Qwen3.5-35B-A3B	0.241	0.246	0.249	0.000
gpt-oss-20b	0.133	0.154	0.152	0.068
gpt-oss-120b	<u>0.287</u>	<u>0.281</u>	<u>0.280</u>	<u>0.289</u>

Table 19: Per-prompt-variant Accuracy for pDetect, macro-averaged across languages (higher is better).

	ZS $\downarrow$	FS1 $\downarrow$	FS5 $\downarrow$	ZS+R $\downarrow$
gemma-3-4b-it	0.770	0.534	0.457	1.052
gemma-3-12b-it	0.580	0.365	0.310	0.459
gemma-3-27b-it	0.440	0.347	0.247	0.286
gemma-4-E2B-it	0.292	0.245	0.253	0.316
gemma-4-E4B-it	0.258	0.224	0.231	0.205
gemma-4-26B-A4B-it	<u>0.187</u>	0.168	0.153	0.130
gemma-4-31B-it	0.196	<u>0.149</u>	0.130	<u>0.122</u>
Qwen3-4B	0.553	0.422	0.319	0.552
Qwen3-8B	0.567	0.344	0.279	0.410
Qwen3-14B	0.322	0.300	0.272	0.329
Qwen3-30B-A3B	0.340	0.274	0.255	0.313
Qwen3-32B	0.713	0.305	0.249	0.306
Qwen3.5-0.8B	3.840	1.009	0.884	4.018
Qwen3.5-2B	0.995	0.823	0.529	0.872
Qwen3.5-4B	0.696	0.443	0.397	2.800
Qwen3.5-9B	0.320	0.296	0.238	2.659
Qwen3.5-27B	0.228	0.207	0.207	2.268
Qwen3.5-35B-A3B	0.231	0.214	0.201	2.438
gpt-oss-20b	0.305	0.274	0.299	0.395
gpt-oss-120b	0.118	0.116	<u>0.132</u>	0.120

Table 20: Per-prompt-variant Mean-Absolute-Error (MAE) for SyCnt, macro-averaged across languages (lower is better).

	ZS ↓	FS1 ↓	FS5 ↓	ZS+R ↓
gemma-3-4b-it	1.024	1.026	0.828	1.224
gemma-3-12b-it	0.843	0.911	0.651	0.861
gemma-3-27b-it	1.417	1.074	0.709	0.717
gemma-4-E2B-it	1.805	2.079	1.400	0.644
gemma-4-E4B-it	1.345	1.085	1.026	0.546
gemma-4-26B-A4B-it	<u>0.642</u>	<u>0.632</u>	0.577	<u>0.398</u>
gemma-4-31B-it	0.895	0.786	<u>0.525</u>	0.420
Qwen3-4B	0.917	1.055	0.761	0.935
Qwen3-8B	1.072	1.166	0.663	0.935
Qwen3-14B	1.446	1.171	0.727	0.788
Qwen3-30B-A3B	1.434	1.229	1.016	0.762
Qwen3-32B	1.076	0.983	0.701	0.659
Qwen3.5-0.8B	7.773	2.018	1.740	3.696
Qwen3.5-2B	2.250	1.278	0.773	1.744
Qwen3.5-4B	1.777	1.303	0.971	6.772
Qwen3.5-9B	1.390	1.267	0.770	6.770
Qwen3.5-27B	0.835	0.763	0.637	6.744
Qwen3.5-35B-A3B	1.471	1.081	0.792	6.753
gpt-oss-20b	1.024	0.933	1.005	1.313
gpt-oss-120b	0.372	0.369	0.432	0.378

Table 21: Per-prompt-variant Mean-Absolute-Error (MAE) for PhCnt, macro-averaged across languages (lower is better).

	ZS ↑	FS1 ↑	FS5 ↑	ZS+R ↑
gemma-3-4b-it	0.303	0.330	0.324	0.355
gemma-3-12b-it	0.415	0.438	0.433	0.432
gemma-3-27b-it	0.445	0.470	0.475	0.450
gemma-4-E2B-it	0.275	0.340	0.323	0.347
gemma-4-E4B-it	0.411	0.420	0.419	0.413
gemma-4-26B-A4B-it	<u>0.497</u>	<u>0.515</u>	0.523	<u>0.519</u>
gemma-4-31B-it	0.536	0.552	0.572	0.553
Qwen3-4B	0.322	0.325	0.334	0.349
Qwen3-8B	0.387	0.360	0.371	0.397
Qwen3-14B	0.400	0.425	0.443	0.425
Qwen3-30B-A3B	0.367	0.396	0.397	0.405
Qwen3-32B	0.389	0.426	0.440	0.431
Qwen3.5-0.8B	0.189	0.242	0.219	0.177
Qwen3.5-2B	0.300	0.298	0.282	0.259
Qwen3.5-4B	0.354	0.377	0.382	0.011
Qwen3.5-9B	0.400	0.431	0.430	0.022
Qwen3.5-27B	0.496	0.512	<u>0.527</u>	0.018
Qwen3.5-35B-A3B	0.491	0.509	<u>0.527</u>	0.025
gpt-oss-20b	0.341	0.361	0.286	0.293
gpt-oss-120b	0.452	0.458	0.460	0.433

Table 22: Per-prompt-variant Accuracy for oDetect, macro-averaged across languages (higher is better).

	ZS $\uparrow$	FS1 $\uparrow$	FS5 $\uparrow$	ZS+R $\uparrow$
gemma-3-4b-it	0.536	0.611	0.636	0.546
gemma-3-12b-it	0.663	0.742	0.763	0.640
gemma-3-27b-it	0.792	0.820	0.835	0.761
gemma-4-E2B-it	0.268	0.570	0.612	0.577
gemma-4-E4B-it	0.579	0.712	0.730	0.703
gemma-4-26B-A4B-it	0.746	0.768	0.781	0.726
gemma-4-31B-it	0.842	0.862	0.874	0.849
Qwen3-4B	0.595	0.659	0.683	0.605
Qwen3-8B	0.693	0.723	0.739	0.645
Qwen3-14B	0.739	0.764	0.787	0.733
Qwen3-30B-A3B	0.722	0.740	0.760	0.665
Qwen3-32B	0.772	0.796	0.808	0.765
Qwen3.5-0.8B	0.114	0.360	0.445	0.114
Qwen3.5-2B	0.343	0.491	0.545	0.323
Qwen3.5-4B	0.606	0.657	0.683	0.011
Qwen3.5-9B	0.685	0.738	0.766	0.018
Qwen3.5-27B	<u>0.836</u>	<u>0.847</u>	<u>0.867</u>	0.079
Qwen3.5-35B-A3B	<u>0.726</u>	<u>0.783</u>	<u>0.798</u>	0.043
gpt-oss-20b	0.461	0.494	0.485	0.453
gpt-oss-120b	0.809	0.805	0.808	<u>0.813</u>

Table 23: Per-prompt-variant Word Boundary F_1 for oSeg, macro-averaged across languages (higher is better).

	ZS $\uparrow$	FS1 $\uparrow$	FS5 $\uparrow$	ZS+R $\uparrow$
gemma-3-4b-it	0.458	0.480	0.468	0.462
gemma-3-12b-it	0.481	0.535	0.516	0.507
gemma-3-27b-it	0.574	0.588	0.587	0.525
gemma-4-E2B-it	0.400	0.400	0.393	0.383
gemma-4-E4B-it	0.458	0.452	0.457	0.469
gemma-4-26B-A4B-it	0.554	0.552	0.547	0.553
gemma-4-31B-it	0.564	0.583	0.574	0.583
Qwen3-4B	0.434	0.446	0.465	0.438
Qwen3-8B	0.509	0.521	0.515	0.508
Qwen3-14B	0.551	0.544	0.538	0.542
Qwen3-30B-A3B	0.532	0.537	0.549	0.541
Qwen3-32B	0.554	0.571	0.557	0.532
Qwen3.5-0.8B	0.353	0.334	0.356	0.314
Qwen3.5-2B	0.436	0.419	0.415	0.358
Qwen3.5-4B	0.523	0.492	0.495	0.482
Qwen3.5-9B	0.564	0.564	0.546	0.549
Qwen3.5-27B	0.608	0.604	0.573	0.603
Qwen3.5-35B-A3B	<u>0.596</u>	<u>0.589</u>	0.575	<u>0.602</u>
gpt-oss-20b	0.477	0.488	0.477	0.468
gpt-oss-120b	0.585	0.577	<u>0.578</u>	0.591

Table 24: Per-prompt-variant Accuracy for oLID, macro-averaged across languages (higher is better).

K Prompt Templates

Below, we provide the complete set of prompt templates across all tasks and prompting variants. Few-shot prompts are rendered as a multi-turn chat history, reusing the single-turn template for each turn.

G2P/ G2P_{syn}

Provide the phonemic (broad) IPA transcription for the following
`{language}{dialect_info}` word written in `{script}` script.
Just return the final answer in the format `IPA: <transcription>` without any explanation.

Word: `{graphemes}`

P2G/ P2G_{syn}

Convert IPA phonemic transcriptions into standard `{language}{dialect_info}` orthography
using `{script}` script.
Just return the final answer in the format `Orthography: <word>` without any explanation.

IPA: `{phonemes}`

pSeg

Below is an IPA transcription of a `{language}` sentence written as a continuous string
without word boundaries. Segment this IPA string into individual words by inserting
spaces at word boundaries. Do not change, remove, or add IPA symbols; only insert
spaces.
Just return the final answer in the format `Segmented: <segmented IPA>` without any
explanation.

Input:
`{phonemes}`

oSeg

Below is a `{language}` sentence written in `{script}` script as a continuous string without
spaces. Segment this string into individual words by inserting spaces at word
boundaries. Do not change, remove, or add any characters; only insert spaces.
Just return the final answer in the format `Segmented: <segmented sentence>` without any
explanation.

Input:
`{graphemes}`

pLID

Below are `{n}` words in IPA phonetic notation, all from the same language. Identify the
language and return the name in English.
Just return the final answer in the format `Language: <language>` without any explanation.

IPA:
`{examples}`

oLID

Below are {n} words written in {script} script, all from the same language. Identify the language and return the name in English.
Just return the final answer in the format `Language: <language>` without any explanation.

Words:
{examples}

pDetect

Below are {n} numbered items in IPA phonetic notation representing {language}{dialect_info} words. Exactly one item is artificially generated and is not an attested word in the language.
Just return the final answer in the format `Pseudo-Word Index: <number>` without any explanation.

IPA words:
{examples}

oDetect

Below are {n} numbered {language}{dialect_info} words written in {script} script. Exactly one item is artificially generated and is not an attested word in the language.
Just return the final answer in the format `Pseudo-Word Index: <number>` without any explanation.

Words:
{examples}

PhCnt

Count the total number of phonemes in the canonical pronunciation of the following {language}{dialect_info} word written in {script} script.
Just return the final answer in the format `Number of phonemes: <number>` without any explanation.

Word: {graphemes}

SyCnt

Count the total number of syllables in the canonical pronunciation of the following {language}{dialect_info} word written in {script} script.
Just return the final answer in the format `Number of syllables: <number>` without any explanation.

Word: {graphemes}

PR

Provide the phonemic (broad) IPA transcription for the following audio recording of a {language}{dialect_info} word.
Just return the final answer in the format `IPA: <transcription>` without any explanation.

Audio:

ZS+R

{task instruction}

You may structure your response in any way you like (e.g. provide step-by-step reasoning),
but the final line must contain only {target
format} and nothing else.

{test input}

ZS+R+C

Context Information

The following text is the content of a Wikipedia article about the phonology of a
specific language. Use this information to assist in answering the subsequent
question:

{wikipedia article}

Task:

{task instruction}

You may structure your response in any way you like (e.g. provide step-by-step reasoning),
but the final line must contain only {target
format} and nothing else.

{test input}

L Software, Data, and Model Licences

We summarize the third-party software, datasets, and models used in this work, together with their licenses in Tables 25, 26, and 27 respectively. All artifacts were used in a manner consistent with their stated licenses and intended-use statements.

Tool	Citation	Licence	Use
vLLM	Kwon et al. (2023)	Apache 2.0	Batched LLM inference.
transformers	Wolf et al. (2020)	Apache 2.0	HuggingFace inference for the Wav2Vec2-XLSR (PR) and CHARSIU-G2P (G2P) baselines.
onnxruntime	—	Apache 2.0	ONNX inference engine for the ZIPA PR baseline.
ESPnet2	Watanabe et al. (2018)	Apache 2.0	Speech2Text inference frontend for the POWSM PR baseline.
panphon	Mortensen et al. (2016)	MIT	PFER via articulatory-feature edit distance.
Epitran	Mortensen et al. (2018)	MIT	IPA transcription (one of two phonemizer variants for pSeg).
eSpeak NG	—	GPL-3.0	IPA transcription (second phonemizer variant for pSeg).
phonemizer	Bernard and Titeux (2021)	GPL-3.0	Python wrapper invoking eSpeak NG.
pyclts	Anderson et al. (2019)	Apache 2.0	Phoneme-inventory normalization (library).

Table 25: Third-party software used in benchmark construction and evaluation.

Dataset	Citation	Licence	Use
WikiPron	Lee et al. (2020)	Apache 2.0	(Grapheme, IPA) pairs for G2P, P2G, PhCnt, SyCnt, and IPA segmentation gold.
wikimedia/wikipedia (2023-11-01 dump)	—	CC-BY-SA 3.0	Source sentences for oSeg and pSeg (26 languages).
Wikipedia (Phonology articles, live scrape)	—	CC-BY-SA 4.0	Context supplied in the ZS+R+C prompt variant.
MSWC	Mazumder et al. (2021)	CC-BY 4.0	Audio data for PR.
FineWeb-2	Penedo et al. (2025)	ODC-BY 1.0	Per-language token counts as resourceness proxy.
CLTS database	Anderson et al. (2019)	CC-BY 4.0	Phoneme-inventory standardization (data).

Table 26: Datasets used to construct and evaluate the benchmark.

Family / system	Checkpoints	Licence	Citation
<i>Text-side LLMs</i>			
Gemma-3	4B-it, 12B-it, 27B-it	Gemma Terms of Use	Team et al. (2025)
Gemma-4	E2B-it, E4B-it, 26B-A4B-it, 31B-it	Apache 2.0	—
Qwen3	4B, 8B, 14B, 30B-A3B, 32B	Apache 2.0	Yang et al. (2025)
Qwen3.5	0.8B, 2B, 4B, 9B, 27B, 35B-A3B	Apache 2.0	Qwen Team (2026)
gpt-oss	20b, 120b	Apache 2.0	OpenAI et al. (2025)
<i>Speech-side LLMs</i>			
Phi-4-multimodal	Phi-4-multimodal-instruct	MIT	Microsoft et al. (2025)
Qwen2.5-Omni	7B	Apache 2.0	Xu et al. (2025a)
Qwen3-Omni	30B-A3B-Instruct	Apache 2.0	Xu et al. (2025b)
<i>Non-LLM baselines</i>			
Wav2Vec2-XLSR	facebook/wav2vec2-xlsr-53-espeak-cv-ft	Apache 2.0	Xu et al. (2022)
ZIPA	anyspeech/zipa-large-crctc-ns-800k	MIT	Zhu et al. (2025)
POWSM	espnet/powsm	CC-BY 4.0	Li et al. (2026)
CharsiuG2P	charsiu/g2p_multilingual_byT5_small_100	MIT	Zhu et al. (2022)

Table 27: Models evaluated in this work and the licenses under which their weights are released. All weights were obtained from the publishers’ Hugging Face repositories.

M Benchmark Statistics

ISO	G2P	G2P _{syn}	P2G	P2G _{syn}	PhCnt	SyCnt	pLID	oLID	pDetect	oDetect	Total
acw_arab	409	409	409	409	409	409	112	112	265	305	3,248
afr_latn	233	233	233	233	233	233	76	76	85	165	1,800
ajp_arab	624	624	624	624	624	624	156	156	482	455	4,993
akk_latn	535	535	535	535	535	535	175	175	347	445	4,352
ang_latn	1,000	1,000	1,000	1,000	1,000	1,000	333	333	1,000	1,000	8,666
ara_arab	1,000	1,000	1,000	1,000	1,000	1,000	298	298	1,000	1,000	8,596
ary_arab	671	671	671	671	671	671	205	205	528	605	5,569
asm_beng	1,000	1,000	1,000	1,000	1,000	1,000	269	269	1,000	1,000	8,538
ben_beng_dhaka	795	795	795	795	795	795	246	246	474	713	6,449
ben_beng_rarh	795	795	795	795	795	795	246	246	474	713	6,449
bod_tibt	970	970	970	970	970	970	317	317	698	875	8,027
cic_latn	242	242	242	242	242	242	80	80	75	175	1,862
cnk_latn	210	210	210	210	210	210	69	69	115	135	1,648
csb_latn	1,000	1,000	1,000	1,000	1,000	1,000	327	327	1,000	1,000	8,654
deu_latn	511	511	511	511	511	511	169	169	205	425	4,034
div_thaa	998	998	998	998	998	998	323	323	784	1,000	8,418
dsb_latn	221	221	221	221	221	221	72	72	66	155	1,691
ell_grek	1,000	1,000	1,000	1,000	1,000	1,000	320	320	1,000	–	7,640
eng_latn_uk	987	987	987	987	987	987	327	327	1,000	1,000	8,576
eng_latn_us	987	987	987	987	987	987	327	327	1,000	1,000	8,576
fas_arab	1,000	1,000	1,000	1,000	1,000	1,000	294	294	1,000	1,000	8,588
fra_latn	997	997	997	997	997	997	329	329	1,000	1,000	8,640
got_goth	521	521	521	521	521	521	171	171	313	425	4,206
grc_grek	997	997	997	997	997	997	326	326	1,000	14	7,648
guj_gujr	1,000	1,000	1,000	1,000	1,000	1,000	301	301	1,000	1,000	8,602
guw_latn	402	402	402	402	402	402	124	124	101	315	3,076
hau_latn	925	925	925	925	925	925	303	303	–	–	6,156
haw_latn	1,000	1,000	1,000	1,000	1,000	1,000	326	326	998	1,000	8,650
heb_hebr	401	401	401	401	401	401	116	116	259	305	3,202
hin_deva	1,000	1,000	1,000	1,000	1,000	1,000	304	304	1,000	1,000	8,608
ido_latn	585	585	585	585	585	585	192	192	406	525	4,825
ind_latn	888	888	888	888	888	888	289	289	1,000	1,000	7,906
isl_latn	253	253	253	253	253	253	84	84	139	205	2,030
ita_latn	1,000	1,000	1,000	1,000	1,000	1,000	332	332	1,000	1,000	8,664
jav_java	1,000	1,000	1,000	1,000	1,000	1,000	215	215	763	705	7,898
kaw_latn	459	459	459	459	459	459	100	100	259	405	3,618
khb_talu	201	201	201	201	201	201	60	60	108	135	1,569
khm_khmr	1,000	1,000	1,000	1,000	1,000	1,000	316	316	1,000	1,000	8,632
lim_latn	278	278	278	278	278	278	91	91	100	215	2,165
lit_latn	291	291	291	291	291	291	97	97	126	205	2,271
mar_deva	1,000	1,000	1,000	1,000	1,000	1,000	307	307	1,000	1,000	8,614
mlt_latn	983	983	983	983	983	983	324	324	1,000	1,000	8,546
mqs_latn	337	337	337	337	337	337	110	110	231	285	2,758
nld_latn	998	998	998	998	998	998	332	332	1,000	1,000	8,652
nno_latn	329	329	329	329	329	329	107	107	189	284	2,661
nya_latn	984	984	984	984	984	984	324	324	818	945	8,315
pan_arab	386	386	386	386	386	386	87	87	193	295	2,978
pan_guru	530	530	530	530	530	530	122	122	322	435	4,181
pol_latn	1,000	1,000	1,000	1,000	1,000	1,000	332	332	1,000	1,000	8,664
por_latn_bz	1,000	1,000	1,000	1,000	1,000	1,000	329	329	1,000	1,000	8,658
por_latn_po	1,000	1,000	1,000	1,000	1,000	1,000	329	329	1,000	1,000	8,658
ron_latn	1,000	1,000	1,000	1,000	1,000	1,000	330	330	1,000	1,000	8,660
san_deva	1,000	1,000	1,000	1,000	1,000	1,000	308	308	1,000	1,000	8,616
scn_latn	589	589	589	589	589	589	193	193	391	515	4,826
sga_latn	1,000	1,000	1,000	1,000	1,000	1,000	332	332	1,000	1,000	8,664
szl_latn	1,000	1,000	1,000	1,000	1,000	1,000	322	322	900	1,000	8,544
tha_thai	1,000	1,000	1,000	1,000	1,000	1,000	325	325	1,000	1,000	8,650
tur_latn	999	999	999	999	999	999	329	329	1,000	1,000	8,652
urd_arab	1,000	1,000	1,000	1,000	1,000	1,000	211	211	1,000	1,000	8,422
wln_latn	998	998	998	998	998	998	319	319	1,000	1,000	8,626
yor_latn	1,000	1,000	1,000	1,000	1,000	1,000	321	321	1,000	1,000	8,642
Total	46,519	46,519	46,519	46,519	46,519	46,519	14,510	14,510	40,214	41,379	389,727

Table 28: Per-language test-item counts for the word-level tasks (G2P, G2P_{syn}, P2G, P2G_{syn}, PhCnt, SyCnt, pLID, oLID, pDetect, oDetect).

ISO	pSeg	oSeg	Total
afr_latn	995	995	1,990
ara_arab	995	995	1,990
ben_beng	995	995	1,990
deu_latn	995	995	1,990
eng_latn	995	995	1,990
est_latn	995	995	1,990
fin_latn	995	995	1,990
fra_latn	995	995	1,990
hin_deva	995	995	1,990
ind_latn	995	995	1,990
kat_geor	995	995	1,990
mal_mlym	995	995	1,990
mlt_latn	995	995	1,990
nld_latn	995	995	1,990
pol_latn	995	995	1,990
por_latn	995	995	1,990
ron_latn	995	995	1,990
slv_latn	995	995	1,990
spa_latn	995	995	1,990
sqi_latn	995	995	1,990
swe_latn	995	995	1,990
tam_taml	995	995	1,990
tel_telu	995	995	1,990
tur_latn	995	995	1,990
uig_arab	995	995	1,990
urd_arab	995	995	1,990
Total	25,870	25,870	51,740

Table 29: Per-language test-item counts for the segmentation tasks (pSeg, oSeg).

ISO	PR
ara_arab	98
asm_beng	7
deu_latn	1,243
ell_grek	103
eng_latn_uk	1,240
eng_latn_us	1,240
fas_arab	3,363
fra_latn	1,235
hau_latn	51
ind_latn	109
ita_latn	645
lit_latn	37
mlt_latn	237
nld_latn	413
pol_latn	498
por_latn_bz	214
por_latn_po	214
ron_latn	151
tur_latn	431
Total	11,529

Table 30: Per-language test-item counts for the phone recognition task (PR).