

BuzzASR: A Swarm of 100+ Monolingual Speech Recognition Models

Shivam Singh¹ Aditya Yadavalli¹ Catherine Arnett² Alex Warstadt¹

¹UC San Diego ²EleutherAI

{shs046, a1yadavalli, awarstadt}@ucsd.edu catherine@eleuther.ai

Abstract

We introduce BuzzASR, a collection of language-specialized fine-tuned Whisper models adapted for automatic speech recognition (ASR) in 102 languages. Large end-to-end Transformer-based ASR models such as Whisper (Radford et al., 2023) have revolutionized ASR, but most prominent models are highly multilingual. As a result, these models often perform poorly on languages less well-represented in their training set. While it has long been known that effective language adaptation can be achieved through simple fine-tuning on monolingual data, this strategy has only been applied to a small number of languages. We massively scale up this simple approach to 102 languages covered in the FLEURS dataset, while also implementing a more complex language adaptation strategy that integrates monolingual tokenizer replacement and data augmentation using text-only fine-tuning. BuzzASR models outperform Whisper-large-v3 on 77 out of 102 languages, reducing character error rates (CER) by a factor of over 2.8 on average. Our models achieve state-of-the-art CER among open-source systems on 27 of 102 languages on the combined FLEURS and Common Voice test set. Our tokenizer replacement strategy yields an average $3.3\times$ improvement in compression rate (characters per token) over Whisper’s multilingual BPE, with gains of up to $21.7\times$. We release all models, code, and detailed results: <https://lemn-lab.github.io/buzz-asr>.

1 Introduction

Language technologies for the top 200 most widely spoken languages would serve approximately 90% of the global population (Ethnologue, 2025). Currently, automatic speech recognition (ASR) models are highly inaccurate for all but a small fraction of those 200 languages. This is not just due to a lack of data. ASR datasets of ≥ 10 hours of speech recordings with text transcriptions exist for at least 1000 languages (Keren et al., 2025). While this

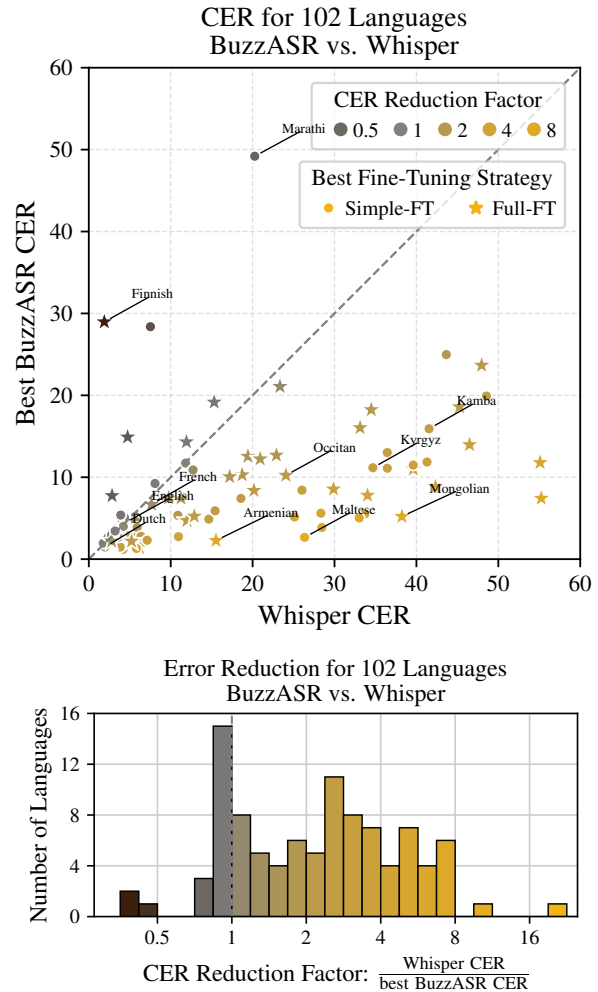


Figure 1: Comparison BuzzASR and Whisper on automatic speech recognition across 102 languages. For each language, we report only the best BuzzASR model. Some outliers are excluded for presentation.

may not seem to be very much data, it can be surprisingly effective when applying domain adaptation to large pretrained multilingual ASR models.

Previous work was able to cut word error rates on low-resource languages by half compared to (at the time) state-of-the-art models, using less than 10 hours of monolingual data (Jimerson et al., 2018; Liu et al., 2024; Özyilmaz et al., 2025; Imam et al.,

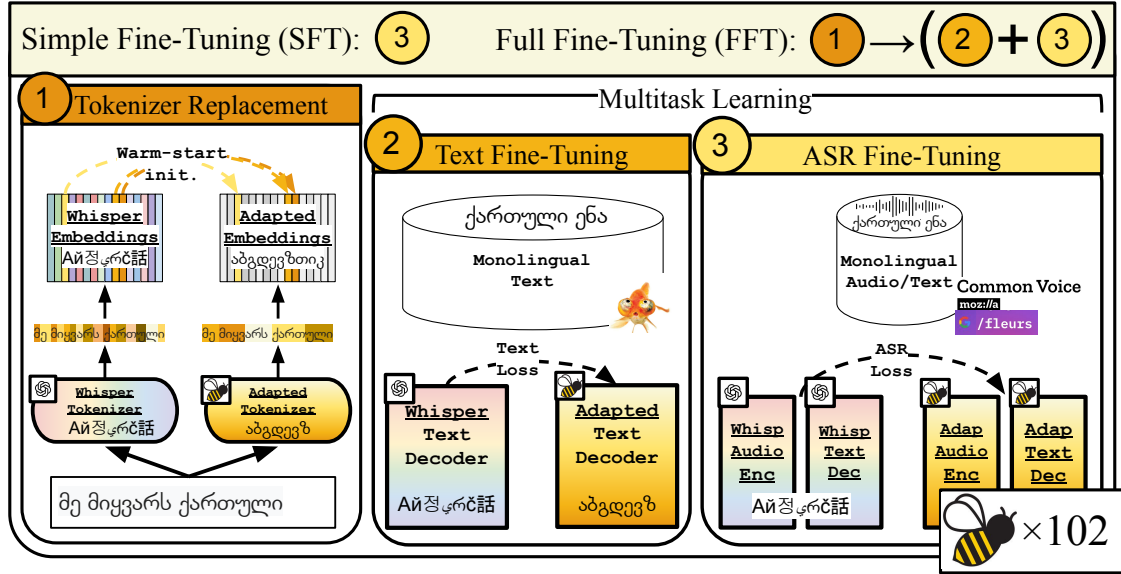


Figure 2: Overview of our two language adaptation approaches. Simple fine-tuning is a minimal approach in which we fine-tune Whisper-large-v3 on monolingual speech recognition data. Full fine-tuning is a 3-stage pipeline involving tokenizer replacement, multitask fine-tuning on text-only and speech recognition, and ending with speech recognition fine-tuning.

2025, *inter alia*). However, these contributions primarily applied this approach to a small number of domains or languages. Thus, the reason why ASR models underperform for most widely spoken languages is not that we lack the ability to train such models; it is simply that the resources that exist have primarily been used to train either massively multilingual models or one-off language-specialized models. Recently, in the text domain, it has been shown that small monolingual text generation models may outperform larger, massively multilingual models on linguistic tasks at the scale of hundreds of languages (Chang et al., 2026). We take a similar approach to ASR and ask whether relatively small models dedicated to a single language can offer better performance, especially for lower-resource languages.

In this paper, we present a suite of 102 language-specific ASR models, which are fine-tuned models based on Whisper (Radford et al., 2023). For 77 of the 102 languages, the best of our fine-tuned models outperforms the original Whisper-Large-v3 on normalized CER, with the median language seeing CER cut by $2.18\times$ and the mean by $2.84\times$. The median absolute CER drops from 15.5% (Whisper zero-shot) to 7.6% (best fine-tuned). Among open-source ASR systems, our models achieve state-of-the-art CER on 27 of 102 languages on a combined evaluation drawn from FLEURS and Common Voice. The gains are greatest where Whisper

is weakest: in Whisper’s 51 worst languages by CER, the median reduction is $3.45\times$, and for ten languages the best fine-tuned model cuts CER by more than $6\times$, led by Amharic, Maltese, and Sorani Kurdish.

While simple fine-tuning is effective, we also explore two improvements to the naïve approach: First, we modify the Whisper tokenizer for each language. The original tokenizer was trained on an imbalanced multilingual dataset, leading to overly long sequence lengths—and therefore worse performance and inference latencies—in poorly represented languages or languages using non-Latin scripts. Second, we fine-tune the Whisper text decoder on monolingual text data, in addition to fine-tuning both the audio encoder and text decoder on ASR data. Many languages are poorly represented in Whisper’s pretraining data, and the limited ASR data is insufficient to learn the target language’s lexicon and grammar. In many such cases, however, sufficient text data is available, allowing us to instill this language-specific knowledge without requiring large amounts of audio data.

The results of this work demonstrate that language-specific models can perform better and more efficiently, making use of relatively small resources. We release a suite of models for 102 languages, 27 of which are, to our knowledge, the state of the art for open-source models.

2 Related Work

2.1 Low-Resource & Multilingual ASR

Earlier work in low-resource ASR primarily focused on developing monolingual or limited multilingual systems for individual low-resource languages (Naing et al., 2015; Popović et al., 2017; Kipyatkova and Karpov, 2016; Bali et al., 2013; Upadhyaya et al., 2017; Ali et al., 2014; Luong and Vu, 2016; Deka et al., 2018, *inter alia*). Later, multilingual pretraining with little to no extra supervision was found to be effective (Watanabe et al., 2017; Toshniwal et al., 2018). Motivated by these findings, Srivastava et al. (2018) and Diwan et al. (2021) released high-quality speech datasets and conducted shared tasks encouraging research in multilingual ASR. Klejch et al. (2021) and Mirishkar et al. (2021) achieved state-of-the-art results on the released shared task datasets, applying various established data-cleaning, augmentation, and multilingual techniques.

In addition to large-scale data efforts, architectural innovations (Vaswani et al., 2017) and advancements in self-supervised learning have made it possible to build powerful scaled-up ASR models like Whisper that achieve state-of-the-art results on several benchmarks (Radford et al., 2023). Notably, wav2vec (Schneider et al., 2019) shows speech representations can be learned from unlabeled audio, leading to reduced dependency on labeled data. Many have scaled this paradigm up in terms of language coverage, number of training hours, and model size (Baevski et al., 2020; Babu et al., 2022; Zhang et al., 2023; Pratap et al., 2024).

At the time of release, Whisper achieved state-of-the-art performance across languages and diverse conditions, leading to widespread adoption (Radford et al., 2023; Olatunji et al., 2023; Bhogale et al., 2023b; Talafha et al., 2023). Whisper also motivated a range of adaptation strategies for new languages, domains, and tasks including fine-tuning (Bhogale et al., 2023b; Yadavalli et al., 2025), parameter-efficient fine-tuning (Kang et al., 2024; Song et al., 2024), prompt-tuning (Ma et al., 2024; Yang et al., 2025), multilingual or multitask fine-tuning (Bhogale et al., 2023b; Olatunji et al., 2023), and other methods (Zhao et al., 2026; Juvekar et al., 2026). Given the broad adoption and the growing body of work on adapting Whisper, we experiment with Whisper models in this study.

2.2 ASR Datasets

Recent efforts in creating and publicly releasing large multilingual ASR corpora have enabled the scaling of multilingual ASR models (Pratap et al., 2020; Wang et al., 2021; Li et al., 2024). However, many of these datasets primarily focus on high-resource European languages. To address this limitation, substantial efforts have been made to collect speech corpora for other languages and regions. One such early effort is BABEL (Gales et al., 2014), a multilingual conversational telephone speech corpus covering 17 low-resource languages. Common Voice (Ardila et al., 2020) is a continuously evolving large-scale crowdsourced effort maintained by Mozilla to collect speech data across a large number of languages. FLEURS (Conneau et al., 2023) is an n -way parallel speech dataset covering 102 languages and is adapted from the FLORES-101 (Goyal et al., 2022) machine translation benchmark. Chen et al. (2024) release an un-transcribed multilingual speech dataset with coverage spanning more than 4000 languages. Most recently, Omnilingual released transcribed data for 348 traditionally under-resourced languages (Keren et al., 2025).

Several region-specific initiatives have also focused on improving speech resources for underrepresented languages. Project Vaani (Pulikodan et al., 2026) released more than 30,000 hours of culturally relevant speech data spanning over 100 languages in the Indian subcontinent while ensuring regional diversity. Shrutilipi (Bhogale et al., 2023a) released more than 6,000 hours of transcribed Indic speech data mined from in-the-wild sources. Similarly, African Next Voices (Marivate et al., 2026), NaijaVoices (Emezue et al., 2025), and WAXAL (Diack et al., 2026) provide high-quality culturally rich speech datasets for several African languages. In this work, we use Common Voice (Ardila et al., 2020) and FLEURS (Conneau et al., 2023) because they provide high-quality multilingual speech data with broad language coverage, making them well-suited for our work.

2.3 Curse of Multilinguality

While joint multilingual training can lead to impressive results due to crosslingual transfer for ASR (Pratap et al., 2024; Keren et al., 2025), multilingual pretraining may also be limited by the “curse of multilinguality” (Conneau et al., 2020; Mei et al., 2026; Zhao et al., 2026). This refers to the phenomenon where training on a very large number of

language leads to degraded performance for many of the languages. This has been studied for text generation models, and has been found to be driven by limited model capacity and negative interference from unrelated languages (Chang et al., 2024).

3 Adaptation strategies

In this paper, we hope to challenge the reliance on the one-model-to-rule-them-all paradigm and advocate for language- and context-specific training to maximize performance for each language. We consider two approaches to adapting pretrained Whisper to a target language: **Simple fine-tuning** entails fine-tuning the entire Whisper model on aligned monolingual speech–text data using the ASR objective. **Full fine-tuning** is a more complex pipeline in which we replace Whisper’s multilingual tokenizer with a language-specialized one and fine-tune on a mixture of monolingual text-only and speech–text data.

Referring to Figure 2, we introduce the basic building blocks of these pipelines below: **tokenizer replacement**, **ASR fine-tuning**, and **text fine-tuning**. Additionally, **multitask fine-tuning** combines text fine-tuning and ASR fine-tuning.

3.1 Tokenizer Replacement

Whisper uses a byte-level byte-pair encoding (BPE) tokenizer (Sennrich et al., 2016; Radford et al., 2019) with a vocabulary of 51,865 tokens: 50,257 BPE merges and 1,608 special tokens, including 99 language tokens. The Whisper tokenizer was trained on a corpus heavily skewed towards English (Radford et al., 2023). As a result, the tokenizer is well-adapted only to the most highly represented languages, and is especially poorly suited to languages with non-Latin and unique scripts. Replacing the vocabulary allows the models to maximally and efficiently benefit from the language-specific adaptation (Minixhofer et al., 2022, *inter alia*).

To perform tokenizer replacement for a target language, we begin by training a new BPE tokenizer on a monolingual text corpus. While we could simply randomly initialize a new embedding matrix for the new vocabulary, this approach fails to leverage existing structure in the pretrained model. Instead, we adopt a **warm-start initialization** of the embedding matrix, resembling Gee et al. (2022). First, embeddings for any entries appearing in both tokenizers’ vocabularies are initialized using the original Whisper embedding. Second, for

any entries appearing only in the new tokenizer’s vocabulary, we tokenize that entry according to the original tokenizer, and initialize the new embedding as the average of the original tokenizer’s embeddings.

For example, the German compound *Geschwindigkeit* (“speed”) receives its own dedicated entry in our per-language German BPE. Whisper’s original tokenizer, lacking a single merge for this compound, decomposes it into four subword tokens: G + esch + wind + igkeit. The new *Geschwindigkeit* embedding is initialized as the mean of those four subword embeddings, preserving whatever lexical information Whisper had already encoded at the level of static embeddings.

We apply the identical warm-start procedure to the output (unembedding) projection, initializing each row from (the average of) the corresponding Whisper output embeddings, so that both the input and output representations of a new token inherit Whisper’s encoded information. Throughout, we keep the token-to-index mapping consistent between the embedding and unembedding matrices.

3.2 ASR Fine-Tuning

During ASR fine-tuning, the entire Whisper encoder–decoder model is fine-tuned on aligned speech–text data using an ASR objective. Our simple fine-tuning pipeline consists only of this step.

3.3 Text Fine-Tuning

During text fine-tuning, the Whisper text decoder is fine-tuned on text-only data using the autoregressive language modeling objective (next-token prediction). As Whisper’s decoder incorporates cross-attention from the encoder embeddings, we pass dummy inputs to the encoder in the form of random embeddings and freeze the encoder.

Text fine-tuning is primarily useful insofar as text-only data is more abundant than speech–text data for the target language. Generally, there are several orders of magnitude difference in dataset size. For some languages, the only ASR data available is from FLEURS, comprising only about 40,000 English words, while we have at least 1M tokens of text data for each language—and in many cases much more data. While in most cases we would prefer ASR fine-tuning given sufficient data, fortunately, there are two areas in which we speculate that training the decoder alone on text data is sufficient:

First, the syntactic priors of the model can conceivably be improved through text fine-tuning. The decoder alone is responsible for outputting grammatical text, while the encoder may specialize in acoustic processing and phonological representations. Chang et al. (2026) showed that even with less than 1GB of text data, small models can learn syntactic information as well as much larger models trained on orders of magnitude more data. Not only is the syntax largely dependent on the text alone, but syntax differs substantially across languages. Text-generation models exhibit worse cross-lingual transfer when they have less similar syntax (Chang et al., 2024). If syntax is more language-specific and harder to learn, we reason that the decoder stands to benefit from relatively abundant text-only input.

Second, if we are performing tokenizer replacement, the model must adapt to the new vocabulary. As this is likely to require extensive training, and the tokenizer is used only by the decoder, text-only inputs are particularly valuable at this stage to prevent tokens in the new vocabulary from being under-trained.

3.4 Multitask Learning

Multitask fine-tuning is a single fine-tuning stage in which each minibatch contains a mixture of ASR examples (aligned speech-text pairs) and text-only examples, with the fraction of ASR examples controlled by a hyperparameter $\alpha \in (0, 1]$, the *ASR proportion*. Each example in the batch contributes its own loss term: the ASR loss for aligned pairs, computed against the full encoder-decoder output; and the language-modeling loss for text-only examples, computed by passing random embeddings as encoder input and freezing the encoder, as described in Section 3.3. Setting $\alpha = 1.0$ recovers pure ASR fine-tuning; setting $\alpha = 0$ recovers pure text fine-tuning.

4 Experiments

We study and compare two fine-tuning strategies (simple fine-tuning and full fine-tuning) to adapt a multilingual model to 102 target languages. All our experiments are done using the 1.55B parameter Whisper-large-v3 model. For each fine-tuning strategy and language, we conduct three fine-tuning runs with different hyperparameter configurations, resulting in 612 fine-tuned models. We report our hyperparameters in Section B.

4.1 Data and Languages

Text-Audio Data We draw aligned text-audio data from two sources: FLEURS (Conneau et al., 2023) and CommonVoice v25 (Ardila et al., 2020). We limit all our experiments and models to the 102 languages represented in FLEURS. Due to compute limitations, we restrict speech-text data to 100 hours (in English, this is approximately 1 million words) for higher-resource languages.

FLEURS is an n -way parallel corpus consisting of English Wikipedia data expert-translated into 102 languages with audio recordings by 3 native speakers. The training set contains approximately 10 hours per language, and the development and test splits contain on the order of 200–1000 samples per language depending on language coverage.

CommonVoice is a crowdsourced dataset with variable coverage of almost 300 languages. We use only data from the 77 languages also in FLEURS. Due to vast differences in data quantities per language and computational constraints, we cap the CommonVoice training data at 90 hours per language. To equalize compute across languages with very large test sets (e.g., English, Russian, Mandarin), we cap each language’s test data at 2,000 utterances. This cap is reached for 36 of the 77 languages that are included in CommonVoice.¹ The remaining 25 out of 102 languages did not have any CommonVoice data, so we use only FLEURS data.

FLEURS provides normalized transcripts, which we use as-is. For CommonVoice, we apply only lightweight text normalization: lowercasing, punctuation removal with a shared regular expression, and whitespace normalization. We keep the preprocessing minimal so as to avoid altering transcript content beyond surface formatting.

For all datasets, audio is resampled to 16 kHz, and model inputs are limited to at most 30 seconds of audio. We use the provided test splits for both datasets for evaluation.

Text Data We sample 500k lines from the fish-food corpus (Chang et al., 2026). Only one FLEURS language (Kamba) had no data in fish-food, so we use the FLEURS training transcripts as text-only data. Text-only training corpora are NFC-normalized,² lowercased, punctuation-stripped, fil-

¹Languages whose CV-test split is below 2,000 are evaluated on all available samples.

²This is a standard unicode normalization that combines separate diacritics and base characters into a single, precom-

tered to require $\geq 70\%$ characters in the language’s primary script, and deduplicated based on exact string match.

17-Language Subset For some supplementary experiments, we study a subset of the 102 FLEURS languages, selected heuristically to represent a wide variety of language families, scripts, and resource levels. These languages, along with key metadata, are listed in Table 2.

4.2 Tokenizer Replacement

For each language, we conduct tokenizer replacement as described in Section 3.1. We train a 51,865-token byte-level BPE tokenizer per language, matching Whisper’s original vocabulary size. This allows us to apply warm-start initialization without resizing the special-token block or the decoder’s unembedding head. We use the same pre-tokenizer as Whisper. Each tokenizer is trained on the same per-language fish-food corpus used for text fine-tuning, so the tokenizer’s vocabulary matches the distribution the decoder will be fine-tuned on.

4.3 Multitask Learning

For multitask fine-tuning experiments, we interleave text-only and ASR samples within a single fine-tuning stage. At each optimization step, we sample either an aligned speech-text sample with probability α or a text-only sample. In a supplementary experiment, we tested different mixing ratios of text-only and ASR samples (10%, 20%, and 50% text data; Section C) with a subset of languages, and found that 50% text data was optimal. Therefore, this is the mix that we used.

4.4 Experimental Configurations

For each of the 102 target languages and each fine-tuning strategy, we train a small set of learning-rate configurations and select, per language, the one with the lowest word error rate on the FLEURS development split. These configurations (**cfg A**, **cfg B**, and **cfg C**) set absolute learning rates for simple fine-tuning. For full fine-tuning, the configurations set the embedding- and decoder-learning-rate multipliers on a fixed base rate. We determined this set of configurations heuristically following a hyperparameter search with a subset of 17 languages (Section B).

Optimization. All runs use AdamW (Loshchilov and Hutter, 2019) with a per-device batch size of 2 and patience-based early stopping on FLEURS-validation WER, evaluated every 100 steps (patience 3 for simple fine-tuning, 5 for full fine-tuning), for at most 6 epochs. Simple fine-tuning uses a 150-step warmup and weight decay $\approx 4 \times 10^{-5}$. Full fine-tuning uses a 200-step warmup and weight decay 0.01 (per-configuration values in Table 3). We compute cross-entropy loss over the label tokens, masking padding positions with -100 . For text-only batches in full fine-tuning, the encoder receives random embeddings and is frozen during the backward pass, so gradients flow only through the decoder.

4.5 Evaluation

Our primary evaluation metric is normalized character error rate (CER). We report both CER and WER in Section J, but focus our discussion on CER since word length varies widely across languages and makes WER less comparable. All reported numbers are on the combined FLEURS + Common Voice test split; for the 25 languages not covered by Common Voice, we report FLEURS results only. CER and WER are computed with jiwer.³

We also report Whisper zero-shot as a baseline. Of the 102 target languages, 83 are supported by Whisper, which we decode using the corresponding Whisper language token. The remaining 19 languages are not supported by Whisper; for these we prompt the model with the most closely related supported language (e.g., Asturian with Spanish, Kyrgyz with Kazakh, Oriya with Bengali). The full mapping is given in Section A. All systems including the Whisper zero-shot baseline are decoded identically with greedy search (num_beams=1, no_repeat_ngram_size=3, repetition_penalty=1.2) and no external language-model fusion.

4.6 Baselines

We compare our methods to the original Whisper-large-v3 models and several recently released multilingual ASR models: Omnilingual 1B and 7B (Keren et al., 2025); Massively Multilingual Speech (MMS, Pratap et al., 2024); Qwen3-ASR 1.7B (Shi et al., 2026); and Cohere Transcribe 2B (Mack et al., 2026). We report raw CER/WER for Qwen3-ASR and Cohere Transcribe, which do not provide

posed character.

³<https://github.com/jitsi/jiwer>

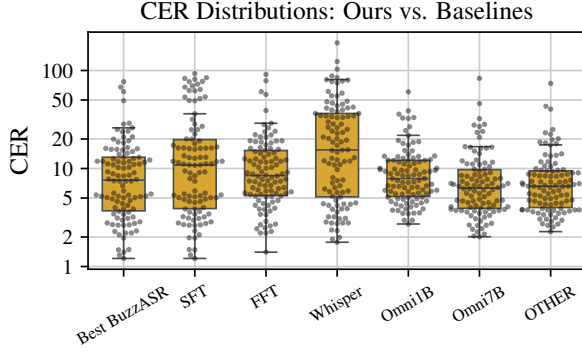


Figure 3: CERs for 102 languages for Whisper, BuzzASR models, Omnilingual models, and all other baseline models (only the best per language is shown).

Baseline	Simple-FT		Full-FT		Best BuzzASR	
	CER	WER	CER	WER	CER	WER
Whisper-ZS	65	65	69	72	77	78
Omni-1B	52	60	44	65	60	76
Omni-7B	36	40	31	43	42	53
MMS-1B	41	54	39	58	53	70
Qwen3-ASR	91	99	91	99	94	100
Cohere-T	99	102	93	100	99	102
All	19	22	19	34	27	39

Table 1: Win rates against external baselines on the combined FLEURS+CV25 test set. Each cell is the number of languages (of 102) where the given BuzzASR strategy achieves lower CER/WER than the baseline. Best is the better of Simple-FT and Full-FT per language.

normalized scores. We evaluate on all 102 languages, although not all languages are supported by every model.

5 Results

In this section, we compare the performance of BuzzASR to Whisper and other pretrained ASR models, as well as reporting the efficacy of tokenizer adaptation. The complete CER and WER rates for BuzzASR and other pretrained models on all 102 languages are reported in Table 7 and Table 8, respectively. Complete tokenizer performance metrics for the adapted tokenizers and the original Whisper tokenizer are reported in Table 6.

5.1 ASR Results

As the BuzzASR models are fine-tuned from Whisper-large-v3, we first compare the performance between the two. Figure 1 compares CER on the combined test split for Whisper-large-v3 vs. BuzzASR. We report the best performance among our two fine-tuning strategies. We find that the BuzzASR models outperform Whisper on 77

languages, reducing CER by a factor of 2.8 on average. The scatter plot (Figure 1; top) shows that not only do most points fall below the diagonal, indicating that BuzzASR generally improves on Whisper-large-v3, but most points are well below the diagonal, showing substantial improvement. Improvements are especially pronounced for languages where Whisper CER is high. The majority of languages where we see little or no improvement are languages where Whisper CER is below 10, including English, Russian, French, Dutch, and Swedish. This suggests that fine-tuning is most valuable for languages under-served by Whisper’s original training distribution, while for high-resource languages, Whisper can remain competitive. For a small set of languages, such as Finnish and Marathi, we find that fine-tuning leads to substantially *worse* performance. We infer that fine-tuning can sometimes be unstable, and leave open the possibility that additional fine-tuning runs could still benefit these languages.

To determine the magnitude of these improvements, we calculate the CER reduction factor for each language, which we define as the ratio between Whisper zero-shot CER and the lower CER achieved by either simple fine-tuning or full fine-tuning. For example, a value of 2 indicates that BuzzASR achieves half the CER of Whisper-large-v3, and any value greater than 1 indicates an improvement. The distribution of CER reduction factors is visualized in Figure 1 (bottom). Across 102 languages, the median reduction factor is 2.2, meaning that the best fine-tuned system more than halves the CER for the median language. On the other hand, the mean reduction factor is much higher at 2.8, reflecting large gains for several languages with very poor zero-shot Whisper performance. In the strongest cases, such as Amharic, Armenian, Maltese, Assamese, Uzbek, and Kabuverdianu, fine-tuning reduces CER by more than $6\times$ relative to Whisper zero-shot.

In Table 1, we show the win rates for BuzzASR models vs. all external baselines. We find that simple fine-tuning improves the CER for 65 languages, while full fine-tuning is better for 69 languages. This suggests that neither fine-tuning method is optimal for all languages, but instead full fine-tuning is more suitable for languages where the base model has a high error rate and simple fine-tuning is more appropriate for languages the model already makes fewer errors on.

Next, we compare to other open-weight models.

Importantly, BuzzASR beats all baselines on CER for 27 languages reaching SOTA performance.⁴ Omni-7B is the strongest baseline, with our best model outperforming it on only 42 languages, using CER. However, against the more comparably sized Omni-1B, BuzzASR performs better for 60 languages. The best BuzzASR model per language outperforms Qwen3-ASR and Cohere Transcribe on nearly every language (94 and 99, respectively). The full distribution of CERs for most models are visualized in Figure 3.

5.2 Tokenizer-quality metrics

Next, we analyze the impact of tokenizer adaptation. Figure 4 (top) compares the compression rate, measured as characters per token, of our tokenizer and Whisper’s tokenizer across all the evaluated languages. The density plot shows that our models have higher compression rates for the vast majority of languages. The median compression rate increases from 2.38 characters per token for Whisper to 5.04 for our tokenizers, resulting in a median gain of $2.18\times$. The effect is especially pronounced for non-Latin scripts, where Whisper’s tokenization is highly fragmented the median compression gain is $3.80\times$ for non-Latin scripts versus $1.93\times$ for Latin scripts. This reduction in fragmentation shortens sequence lengths, thus making multilingual speech recognition more efficient. We refer the readers to Table 5, which shows the script-wise improvement in ASR latency due to our improved tokenizers. Furthermore, the distribution of compression rates across languages appears roughly normal, which suggests that the compression rates are more consistent cross-linguistically.

We also examine how the improvements in tokenization efficiency are associated with downstream ASR gains using Figure 4 (bottom). The x -axis shows the compression rate ratio between our tokenizer and Whisper’s tokenizer, while the y -axis shows the CER reduction factor obtained by the BuzzASR model with full fine-tuning relative to Whisper zero-shot. The regression line shows a general positive relationship. This indicates that languages for which our tokenizer provides larger compression gains also tend to show larger CER reductions after fine-tuning. However, this trend is moderate rather than deterministic, indicating that tokenization is only one of the many factors affecting the ASR performance. The effect is especially

⁴BuzzASR outperforms all baselines on 39 languages when using WER as an evaluation metric.

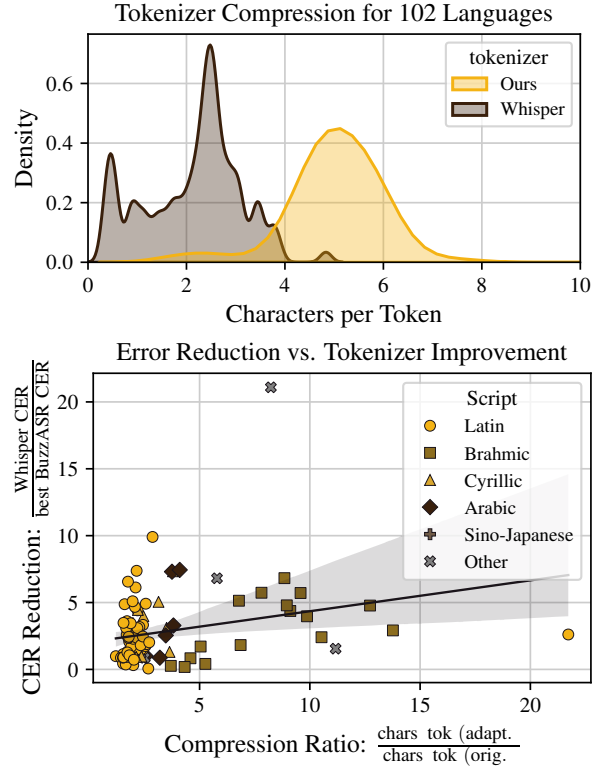


Figure 4: **Top:** Distribution of tokenizer compression rates for all languages with original and adapted tokenizers. **Bottom:** Relationship between improvement in CER between Whisper and full fine-tuning by writing system and change in compression rate after tokenizer adaptation.

pronounced for non-Latin script languages—they have a higher median compression ratio than Latin-script languages and also tend to obtain larger CER reductions. We report additional tokenizer metrics, such as UTF-8 coverage, vocab utilization, and boundary crossing, in Section I.

5.3 Supplementary Results

In the Appendix, we report several supplementary experiments. We briefly summarize those findings below.

For multitask fine-tuning, higher proportions of ASR training are better. In Section C, we explore variants of the multitask fine-tuning training hyperparameters. Among the settings we compare, 50% ASR data is optimal, and increasing the amount of text-only data yields modest improvements.

Warm-start initialization improves tokenizer adaptation. In Section D, we compare alternative strategies for tokenizer adaptation, including a tokenizer replacement strategy where we randomly

initialize new language-specific tokens and a tokenizer adaptation strategy where we append new tokens to the end of the original tokenizer merge list. The warm-start tokenizer replacement strategy used in our main full fine-tuning experiments performs best.

Out-of-domain evaluation hurts performance slightly. In Section E, we test out-of-domain generalization of models trained on FLEURS alone. We find that out-of-domain evaluation hurts performance slightly. This suggests that some (but probably not all) of the performance gains we observe over baseline models are due to better domain alignment of our models’ train and test data, rather than language-specific fine-tuning.

Multilingual fine-tuning harms performance slightly. In Section F, we compare bilingual fine-tuning runs to monolingual runs. For nearly all language pairs (including related and unrelated language pairs), we find that monolingual fine-tuning is more effective.

Multitask fine-tuning is not helpful without tokenizer replacement. In Section G, we disentangle the effects of tokenizer replacement and multitask fine-tuning, which are conflated in our full fine-tuning experiments. We find that full fine-tuning is more effective than multitask fine-tuning without tokenizer replacement.

6 Discussion and Conclusion

BuzzASR models provide SOTA or near-SOTA automatic speech recognition for dozens of languages. Given their Whisper-based architecture, BuzzASR models can be easily deployed in any pipeline already implemented for Whisper where strong monolingual performance is needed on one of our covered languages. While BuzzASR models are outperformed by Omni-7B on the majority of languages, we note that BuzzASR generally outperforms Omni-1B, which has a more comparable number of parameters. Thus, in cases where Omni-7B is too large to deploy, BuzzASR offers competitive performance for the majority of languages we cover.

In addition to providing a new open-source model suite, our experiments demonstrate the efficacy of language-specialized fine-tuning at scale. Given the benefits of language-specialized models to the global community, our work motivates similar approaches with other ASR model architectures,

as well as with text models and multimodal models.

Finally, our tokenizer replacement approach reveals the extent to which multilingual tokenizers under-deliver, particularly for non-Latin scripts. We show that language-adapted tokenizers not only improve ASR performance disproportionately in cases where the original tokenizer was ill-suited, they also decrease inference latency and compute by reducing sequence lengths.

Future Work There are significant opportunities for future improvements to BuzzASR. First, we believe through further hyperparameter-tuning for vocabulary adaptation and multi-task learning we could achieve better performance. We focused on a limited set of design choices to optimize, and most significantly improved performance. By further tuning the multitask learning protocol, we could better make use of abundant text-only data.

Second, there is a wide array of established optimizations for ASR that we do not yet exploit. These include data augmentation through the addition of noise or text-to-speech, interpolation with monolingual text-only LMs, and training simultaneously on closely related languages.

Finally, we focus on a relatively small set of languages. As we note above, there is sufficient fine-tuning data for roughly 1000 languages. We hope to expand our suite of BuzzASR models, especially to include under-represented and under-studied languages. As ASR can be used to increase text data for training language models, creating better tools for these languages can lead to improvements in other language technologies.

Limitations

In this paper, we only conduct fine-tuning experiments on Whisper. However, Omni-7B still shows state-of-the-art performance for most languages. We anticipate that fine-tuning Omnilingual would result in even better performance. We focused on Whisper because it is very widely used and integrated into many existing pipelines. Replacing Whisper with one of our models would require minimal adjustments for users. Furthermore, Whisper-large-v3 is significantly smaller than Omni-7B, with only 1.6B parameters. Focusing on smaller models results in more efficient and economical models, and we note that BuzzASR models outperform the similarly sized Omni-1B in the vast majority of cases. While we aimed to produce performant models, the main goal of the paper was

to demonstrate the efficacy of our fine-tuning approach for models optimized for a single language. Future work could explore using other models as starting points.

We used only a small fraction of the existing ASR data. We limited the data we use to 100 hours of data, primarily due to computational resource limitations. For the languages we fine-tuned models for in this paper, we could achieve even higher performance by using more of the available data. There are also many languages which we did not train models for, but for which there is a significant amount of data. We hope this approach can be applied to a wider set of languages.

We use the test splits from our training datasets (FLEURS and Common Voice) for evaluation. Due to data limitations, we do not evaluate performance across different domains or dialects, as such datasets are not available for many of the languages we consider in this paper. A supplementary analysis Section E reveals that out-of-domain generalization may be a challenge for adapted ASR models resembling the BuzzASR releases.

We also note that monolingual models may provide better performance in controlled evaluation, but might not be appropriate for all applications. Monolingual models are best when one already knows the language that is being used, which may limit their utility. If coverage of multiple languages is required, deploying multiple monolingual models will require substantially more compute resources than deploying a single multilingual model. This is especially true given we perform full fine-tuning, rather than parameter-efficient fine-tuning, which improves memory efficiency in exchange for some losses in performance on ASR (Liu et al., 2024). Additionally, multilingual models may better be able to handle code switching or multi-dialectal speech, though this may require additional specialized training or fine-tuning.

Acknowledgments

This work used resources available through the National Research Platform (NRP) at the University of California, San Diego (Weitzel et al., 2025). NRP has been developed, and is supported in part, by funding from National Science Foundation, from awards 1730158, 1540112, 1541349, 1826967, 2112167, 2100237, and 2120019, as well as additional funding from community partners. We also thank CoreWeave for providing compute for

this project.

References

- Ahmed Ali, Yifan Zhang, Patrick Cardinal, Najim Dahak, Stephan Vogel, and James Glass. 2014. [A complete KALDI recipe for building Arabic speech recognition systems](#). In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 525–529.
- 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.
- Arun Babu, Changan Wang, Andros Tjandra, Kushal Lakhota, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2022. [XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale](#). In *Interspeech 2022*, pages 2278–2282.
- Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. [wav2vec 2.0: A framework for self-supervised learning of speech representations](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.
- Kalika Bali, Sunayana Sitaram, Sebastien Cuendet, and Indrani Medhi. 2013. [A Hindi speech recognizer for an agricultural video search application](#). In *Proceedings of the 3rd ACM Symposium on Computing for Development, ACM DEV ’13*, New York, NY, USA. Association for Computing Machinery.
- Kaushal Bhogale, Abhigyan Raman, Tahir Javed, Sumanth Doddapaneni, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2023a. [Effectiveness of mining audio and text pairs from public data for improving ASR systems for low-resource languages](#). In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.
- Kaushal Bhogale, Sai Sundaresan, Abhigyan Raman, Tahir Javed, Mitesh M. Khapra, and Pratyush Kumar. 2023b. [Vistaar: Diverse Benchmarks and Training Sets for Indian Language ASR](#). In *Interspeech 2023*, pages 4384–4388.
- Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin Bergen. 2026. [Goldfish: Monolingual language models for 350 languages](#). *Proceedings of the Fifteenth Language Resources and Evaluation Conference*, pages 3750–3781.
- Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K. Bergen. 2024. [When is multilinguality a curse? Language modeling for 250 high- and](#)

- low-resource languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4074–4096, Miami, Florida, USA. Association for Computational Linguistics.
- William Chen, Wangyou Zhang, Yifan Peng, Xinjian Li, Jinchuan Tian, Jiatong Shi, Xuankai Chang, Soumi Maiti, Karen Livescu, and Shinji Watanabe. 2024. [Towards robust speech representation learning for thousands of languages](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10205–10224, Miami, Florida, USA. Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2023. [FLEURS: Few-shot learning evaluation of universal representations of speech](#). In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805.
- Barsha Deka, S R Nirmala, and Samudravijaya K. 2018. [Development of Assamese Continuous Speech Recognition System](#). In *6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, pages 220–224.
- Abdoulaye Diack, Perry Nelson, Kwaku Agbesi, Angela Nakalembe, MohamedElfatih MohamedKhair, Vusumuzi Dube, Tavonga Siyavora, Subhashini Venugopalan, Jason Hickey, Uche Okonkwo, Abhishek Bapna, Isaac Wiafe, Raynard Dodzi Helegah, Elikem Doe Atsakpo, Charles Nutrokor, Fiifi Baffoe Payin Winful, Kafui Kwashie Solaga, Jamal-Deen Abdulai, Akon Obu Ekpezu, and 24 others. 2026. [WAXAL: A large-scale multilingual African language speech corpus](#).
- Anuj Diwan, Rakesh Vaideeswaran, Sanket Shah, Ankita Singh, Srinivasa Raghavan, Shreya Khare, Vinit Unni, Saurabh Vyas, Akash Rajpuria, Chiranjeevi Yarra, Ashish Mittal, Prasanta Kumar Ghosh, Preethi Jyothi, Kalika Bali, Vivek Seshadri, Sunayana Sitaram, Samarth Bharadwaj, Jai Nanavati, Raoul Nanavati, and Karthik Sankaranarayanan. 2021. [MUCS 2021: Multilingual and Code-Switching ASR Challenges for Low Resource Indian Languages](#). In *Interspeech 2021*, pages 2446–2450.
- Chris Emezue, The NaijaVoices Community, Busayo Awobade, Abraham Toluwase Owodunni, Handel Emezue, Gloria Monica Tobechukwu Emezue, Nefertiti Nneoma Emezue, Sewade Ogun, Bunmi Akinremi, David Ifeoluwa Adelani, and Chris Pal. 2025. [The NaijaVoices Dataset: Cultivating Large-Scale, High-Quality, Culturally-Rich Speech Data for African Languages](#). In *Interspeech 2025*, pages 1338–1342.
- Ethnologue. 2025. What are the top 200 most spoken languages? <https://www.ethnologue.com/insights/ethnologue200/>. Ethnologue Insights.
- Mark J. F. Gales, Kate M. Knill, Anton Ragni, and Shakti P. Rath. 2014. [Speech recognition and keyword spotting for low-resource languages: Babel project research at CUED](#). In *4th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2014)*, pages 16–23.
- Leonidas Gee, Andrea Zugarini, Leonardo Rigutini, and Paolo Torrioni. 2022. [Fast vocabulary transfer for language model compression](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 409–416, Abu Dhabi, UAE. Association for Computational Linguistics.
- Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc’Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. [The Flores-101 evaluation benchmark for low-resource and multilingual machine translation](#). *Transactions of the Association for Computational Linguistics*, 10:522–538.
- Sukairaj Hafiz Imam, Babangida Sani, Dawit Ketema Gete, Bedru Yimam Ahmed, Ibrahim Said Ahmad, Idris Abdulmumin, Seid Muhie Yimam, Muhammad Yahuza Bello, and Shamsuddeen Hassan Muhammad. 2025. [Automatic speech recognition for African low-resource languages: Challenges and future directions](#). In *Proceedings of the Sixth Workshop on African Natural Language Processing (AfricaNLP 2025)*, pages 89–94, Vienna, Austria. Association for Computational Linguistics.
- Robert Jimerson, Karthik Simha, Raymond Ptucha, and Emily Prudhommeaux. 2018. [Improving ASR output for endangered language documentation](#). In *Proceedings of the 6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, pages 187–191. International Speech Communication Association.
- Kush Juvekar, Kavya Manohar, Aditya Srinivas Menon, Arghya Bhattacharya, and Kumarmanas Nethil. 2026. [Vividh-ASR: A complexity-tiered benchmark and optimization dynamics for robust Indic speech recognition](#). *Preprint*, arXiv:2605.13087.
- Ji-Hun Kang, Jae-Hong Lee, Mun-Hak Lee, and Joon-Hyuk Chang. 2024. [Whisper Multilingual Downstream Task Tuning Using Task Vectors](#). In *Interspeech 2024*, pages 2385–2389.
- Gil Keren, Artyom Kozhevnikov, Yen Meng, Christophe Ropers, Matthew Setzler, Skyler Wang, Ife Adenbara, Michael Auli, Can Balioglu, Kevin Chan,

- Chierh Cheng, Joe Chuang, Caley Droof, Mark Dupenthaler, Paul-Ambroise Duquenne, Alexander Erben, Cynthia Gao, Gabriel Mejia Gonzalez, Kehan Lyu, and 13 others. 2025. [Omnilingual ASR: Open-source multilingual speech recognition for 1600+ languages](#). *Preprint*, arXiv:2511.09690.
- Irina Kipyatkova and Alexey Karpov. 2016. [DNN-based acoustic modeling for Russian speech recognition using Kaldi](#). In *Speech and Computer*, pages 246–253, Cham. Springer International Publishing.
- Ondřej Klejch, Electra Wallington, and Peter Bell. 2021. [The CSTR System for Multilingual and Code-Switching ASR Challenges for Low Resource Indian Languages](#). In *Interspeech 2021*, pages 2881–2885.
- Song Li, Yongbin You, Xuezhi Wang, Zhengkun Tian, Ke Ding, and Guanglu Wan. 2024. [MSR-86K: An Evolving, Multilingual Corpus with 86,300 Hours of Transcribed Audio for Speech Recognition Research](#). In *Interspeech 2024*, pages 1245–1249.
- Yunpeng Liu, Xukui Yang, and Dan Qu. 2024. [Exploration of Whisper fine-tuning strategies for low-resource ASR](#). *EURASIP Journal on Audio, Speech, and Music Processing*, 2024(1):31.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Hieu-Thi Luong and Hai-Quan Vu. 2016. [A non-expert Kaldi recipe for Vietnamese speech recognition system](#). In *Proceedings of the Third International Workshop on Worldwide Language Service Infrastructure and Second Workshop on Open Infrastructures and Analysis Frameworks for Human Language Technologies (WLSI/OIAF4HLT2016)*, pages 51–55, Osaka, Japan. The COLING 2016 Organizing Committee.
- Hao Ma, Zhiyuan Peng, Mingjie Shao, Jing Li, and Ju Liu. 2024. [Extending Whisper with prompt tuning to target-speaker ASR](#). In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12516–12520.
- Julian Mack, Ekagra Ranjan, Walter Beller-Morales, Bharat Venkitesh, and Pierre Richemond. 2026. [cohere-transcribe-03-2026 \(revision d96e814\)](#).
- Vukosi Marivate, Kayode Olaleye, Sitwala Mundia, Andinda Bakainga, Unarine Netshifhefhe, Mahmooda Milanzie, Tsholofelo Hope Mogale, Thapelo Sindane, Zainab Abdulrasaq, Kesego Mokgosi, Chijoke Okorie, Nia Zion Van Wyk, Graham Morrissey, Dale Dunbar, Francois Smit, Tsosheletso Chidi, Rooweither Mabuya, Andiswa Bukula, Respect Mlambo, and 4 others. 2026. [Swivuriso: The South African next voices multilingual speech dataset](#). *Preprint*, arXiv:2512.02201.
- Yuxiang Mei, Delai Qiu, Shengping Liu, Jiaen Liang, and Yanhua Long. 2026. [Zipper-LoRA: Dynamic parameter decoupling for speech-LLM based multilingual speech recognition](#). *Preprint*, arXiv:2603.17558.
- Benjamin Minixhofer, Fabian Paischer, and Navid Rekasaz. 2022. [WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.
- Ganesh Mirishkar, Aditya Yadavalli, and Anil Kumar Vuppala. 2021. [An investigation of hybrid architectures for low resource multilingual speech recognition system in Indian context](#). In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 205–212, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLP AI).
- Hay Mar Soe Naing, Aye Mya Hlaing, Win Pa Pa, Xinhui Hu, Ye Kyaw Thu, Chiori Hori, and Hisashi Kawai. 2015. [A Myanmar large vocabulary continuous speech recognition system](#). In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 320–327.
- Tobi Olatunji, Tejumade Afonja, Aditya Yadavalli, Chris Chinenye Emezue, Sahib Singh, Bonaventure F. P. Dossou, Joanne Osuchukwu, Salomey Osei, Atnafu Lambebo Tonja, Naome Etori, and Clinton Mbataku. 2023. [AfriSpeech-200: Pan-African accent speech dataset for clinical and general domain ASR](#). *Transactions of the Association for Computational Linguistics*, 11:1669–1685.
- Branislav Popović, Edvin Pakoci, and Darko Pekar. 2017. [End-to-end large vocabulary speech recognition for the Serbian language](#). In *Speech and Computer*, pages 343–352, Cham. Springer International Publishing.
- Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaocheng 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.
- Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. [MLS: A Large-Scale Multilingual Dataset for Speech Research](#). In *Interspeech 2020*, pages 2757–2761.
- Sujith Pulikodan, Abhayjeet Singh, Agneeth Basu, Nihar Desai, Pavan Kumar J, Pranav D Bhat, Raghu Dharmaraju, Ritika Gupta, Sathvik Udupa, Saurabh Kumar, Sumit Sharma, Vaibhav Vishwakarma, Visruth Sanka, Dinesh Tewari, Harsh Dhand, Amrita Kamat, Sukhwinder Singh, Shikhar Vashishth, Partha Talukdar, and 2 others. 2026. [Vaani: Capturing the language landscape for an inclusive digital india](#). *Preprint*, arXiv:2603.28714.

- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. [Robust speech recognition via large-scale weak supervision](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). *OpenAI Technical Report*.
- Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. [wav2vec: Unsupervised Pre-Training for Speech Recognition](#). In *Interspeech 2019*, pages 3465–3469.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. 2026. [Qwen3-ASR Technical Report](#). Preprint, arXiv:2601.21337.
- Zhesu Song, Jianheng Zhuo, Yifan Yang, Ziyang Ma, Shixiong Zhang, and Xie Chen. 2024. [LoRA-Whisper: Parameter-Efficient and Extensible Multilingual ASR](#). In *Interspeech 2024*, pages 3934–3938.
- Brij Mohan Lal Srivastava, Sunayana Sitaram, Rupesh Kumar Mehta, Krishna Doss Mohan, Pallavi Matani, Sandeepkumar Satpal, Kalika Bali, Radhakrishnan Srikanth, and Niranjana Nayak. 2018. [Interspeech 2018 Low Resource Automatic Speech Recognition Challenge for Indian Languages](#). In *6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, pages 11–14.
- Bashar Talafha, Abdul Waheed, and Muhammad Abdul-Mageed. 2023. [N-Shot Benchmarking of Whisper on Diverse Arabic Speech Recognition](#). In *Interspeech 2023*, pages 5092–5096.
- Shubham Toshniwal, Tara N. Sainath, Ron J. Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao. 2018. [Multilingual speech recognition with a single end-to-end model](#). In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4904–4908.
- Prashant Upadhyaya, Omar Farooq, Musiur Raza Abidi, and Yash Vardhan Varshney. 2017. [Continuous Hindi speech recognition model based on Kaldi ASR toolkit](#). In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 786–789.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.
- Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021. [VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 993–1003, Online. Association for Computational Linguistics.
- Shinji Watanabe, Takaaki Hori, and John R. Hershey. 2017. [Language independent end-to-end architecture for joint language identification and speech recognition](#). In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 265–271.
- Derek Weitzel, Ashton Graves, Sam Albin, Huijun Zhu, Frank Wuerthwein, Mahidhar Tatineni, Dmitry Mishin, Elham Khoda, Mohammad Sada, Larry Smarr, Thomas DeFanti, and John Graham. 2025. [The National Research Platform: Stretched, multi-tenant, scientific kubernetes cluster](#). In *Practice and Experience in Advanced Research Computing 2025: The Power of Collaboration, PEARC ’25*, New York, NY, USA. Association for Computing Machinery.
- Aditya Yadavalli, Tiago Pimentel, Tamar I Regev, Ethan Wilcox, and Alex Warstadt. 2025. [What do prosody and text convey? Characterizing how meaningful information is distributed across multiple channels](#). Preprint, arXiv:2512.16832.
- Hongli Yang, Yizhou Peng, Hao Huang, and Sheng Li. 2025. [Adapting Whisper for Parameter-efficient Code-Switching Speech Recognition via Soft Prompt Tuning](#). In *Interspeech 2025*, pages 5203–5207.
- Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, Zhong Meng, Ke Hu, Andrew Rosenberg, Rohit Prabhavalkar, Daniel S. Park, Parisa Haghani, Jason Riesa, Ginger Perng, Hagen Soltau, and 8 others. 2023. [Google USM: Scaling automatic speech recognition beyond 100 languages](#). Preprint, arXiv:2303.01037.
- Qiuming Zhao, Guangzhi Sun, and Chao Zhang. 2026. [Low-rank and sparse model merging for multi-lingual speech recognition and translation](#). In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 18827–18831.
- Ömer Tarik Özyilmaz, Matt Coler, and Matias Valdenegro-Toro. 2025. [Overcoming Data Scarcity in Multi-Dialectal Arabic ASR via Whisper Fine-Tuning](#). In *Interspeech 2025*, pages 1158–1162.

A Unsupported Whisper Languages

Out of the 102 languages we experiment with, 19 of them are not natively supported by Whisper. For such languages, we prompt the model with the language token that is most closely related to the target language. These languages are Asturian, Cebuano, Sorani Kurdish, Fulah, Irish, Igbo, Kamba, Kabuverdianu, Kyrgyz, Luganda, Luo, Northern Sotho, Nyanja, Oromo, Oriya, Umbundu, Wolof, Xhosa, and Zulu.

B Hyperparameters

B.1 Hyperparameters for multitask fine-tuning

For each language and each fine-tuning strategy we train models with three configurations, A, B, C (Table 3). We use two random seeds (42, 1337) for simple fine-tuning and three (42, 1337, 2024) for full fine-tuning.

For simple fine-tuning, we use an absolute decoder learning rate and set the encoder rate to $0.3 \times$ the decoder rate, so as to preserve Whisper’s pre-trained acoustic features.

We apply patience-based early stopping on FLEURS development throughout. For full fine-tuning, we use a patience of 5 and evaluate every 100 steps. For simple fine-tuning, we use a patience of 3 and evaluate every 100 steps and select checkpoints by validation WER. For full fine-tuning, we select checkpoints by validation WER for high-coverage languages but use validation loss for a small set of low-coverage languages where the WER is noisy.⁵

We use $8 \times A100$ -80GB for all experiments.

B.2 Hyperparameter Sweep Results

We conducted a set of Bayesian hyperparameter sweeps for the 17 core languages. Table 4 lists the per-language optima for the 17 core languages sorted by FLEURS-test WER (lower is better). These optima motivated the discrete simple fine-tuning configurations used at scale and were not deployed directly. We ran a 5-run sweep over a six-dimensional search space: decoder learning rate $\in [10^{-5}, 8 \times 10^{-5}]$, gradient accumulation $\in \{16, 24, 32\}$, scheduler $\in \{\text{linear}, \text{cosine}, \text{exponential}\}$, decay rate $\in \{0.9, 0.95, 0.99\}$, warmup steps $\in \{100, 150\}$,

and weight decay $\in [10^{-5}, 10^{-3}]$. Each run trained for up to 12 epochs with patience 3.

⁵Full fine-tuning loss-based selection is used for: Tamil, Latvian, Uzbek, Cantonese, Pashto, Swahili, Tajik, Georgian, and Ukrainian. This split was determined empirically.

Language	Code	Script	Family	Region	Speakers
Arabic	ar	Arabic	Afro-Asiatic (Semitic) ^a	MENA	411M
Cantonese	— ^b	Chinese (Han)	Sino-Tibetan	S. China / Hong Kong	85M
Catalan	ca	Latin	Indo-European (Romance)	NE Spain / Andorra	4.1M
Galician	gl	Latin	Indo-European (Romance)	NW Spain (Galicia)	2.4M
Georgian	ka	Georgian	Kartvelian	South Caucasus	3.76M
German	de	Latin	Indo-European (Germanic)	Central Europe	95M
Italian	it	Latin	Indo-European (Romance)	Southern Europe	65M
Latvian	lv	Latin	Indo-European (Baltic)	Baltic region	1.5M
Pashto	ps	Perso-Arabic	Indo-European (Iranian)	Afghanistan / Pakistan	51M
Polish	pl	Latin	Indo-European (Slavic)	Central Europe	40M
Russian	ru	Cyrillic	Indo-European (Slavic)	E. Europe / N. Asia	145M
Spanish	es	Latin	Indo-European (Romance)	Iberia / Americas	519M
Swahili	sw	Latin	Niger–Congo (Bantu)	East Africa	5.3M ^c
Tajik	tg	Cyrillic	Indo-European (Iranian)	Central Asia	10.5M
Tamil	ta	Tamil	Dravidian	S. India / Sri Lanka	79M
Ukrainian	uk	Cyrillic	Indo-European (Slavic)	Eastern Europe	32M
Uzbek	uz	Latin	Turkic	Central Asia	34M

Table 2: Overview of 17 sample languages, with ISO 639-1 codes, writing systems, language families, primary geographic regions, and approximate number of first-language (L1) speakers, based on Wikipedia data as of 2026 (mainly citing Ethnologue).

^a Cantonese lacks an ISO 639-1 code, being covered by the Chinese macrolanguage code zh. Its ISO 639-3 code is yue.

^b This figure reflects the combined L1 total across all varieties under the Arabic macrolanguage as classified by Ethnologue/ISO.

^c Swahili L1 speakers are vastly outnumbered by L2 speakers (approx. 92M L2 vs. 5.3M L1).

Cfg	dec LR	enc LR	embed LR	ga	sched.	decay	warmup	wd
SFT								
A	6.48×10^{-6}	$0.3 \times \text{dec LR}$	—	16	cosine	0.95	150	4.35×10^{-5}
B	2.00×10^{-5}	$0.3 \times \text{dec LR}$	—	32	cosine	0.95	150	4.00×10^{-5}
C	3.24×10^{-6}	$0.3 \times \text{dec LR}$	—	16	cosine	0.95	150	4.35×10^{-5}
FFT								
A	3.0×10^{-5}	3×10^{-6}	3.0×10^{-5}	16	cosine-delay	—	200	0.01
B	1.0×10^{-5}	3×10^{-6}	3.0×10^{-5}	16	cosine-delay	—	200	0.01
C	2.0×10^{-5}	3×10^{-6}	2.0×10^{-5}	16	cosine-delay	—	200	0.01

Table 3: Fine-tuning configurations, fully specified. **dec/embed LR**: decoder/embedding learning rate; **ga**: gradient accumulation; **sched.**: LR schedule (cosine-delay = cosine with a 1000-step delay, 8000-step period); **decay**: scheduler decay rate; **wd**: weight decay. All runs use AdamW and micro-batch size 2 (effective batch = $2 \times \text{ga}$). SFT uses a single learning rate (no separate embedding LR); FFT scales a fixed base rate η by per-config embedding/decoder multipliers and differs across configs only in those two rates. SFT and FFT letters index independent schemes. For each language and strategy we train configurations {A, B, C} and keep the run with the lowest FLEURS-dev WER, over two seeds (42, 1337) for SFT and three seeds (42, 1337, 2024) for FFT.

Language	WER	lr	ga	sched.	decay	warmup	wd
Italian	2.39%	1.07×10^{-5}	32	linear	0.90	150	2.65×10^{-5}
Spanish	3.75%	4.41×10^{-5}	32	cosine	0.99	150	4.93×10^{-5}
Catalan	4.29%	1.58×10^{-5}	24	cosine	0.90	100	1.69×10^{-4}
Russian	5.35%	1.77×10^{-5}	24	linear	0.99	100	6.51×10^{-5}
Polish	5.98%	1.87×10^{-5}	24	linear	0.95	100	4.39×10^{-5}
Galician	6.32%	1.65×10^{-5}	16	linear	0.99	150	6.63×10^{-4}
German	6.47%	4.30×10^{-5}	32	cosine	0.95	150	6.86×10^{-5}
Ukrainian	6.87%	1.16×10^{-5}	16	linear	0.90	150	5.93×10^{-4}
Cantonese	9.39%	7.81×10^{-5}	24	exponential	0.95	150	3.36×10^{-4}
Latvian	11.90%	2.33×10^{-5}	16	linear	0.99	150	1.71×10^{-5}
Arabic	13.13%	2.03×10^{-5}	32	cosine	0.90	100	3.40×10^{-5}
Tajik	13.34%	4.16×10^{-5}	16	linear	0.95	100	1.30×10^{-5}
Swahili	24.43%	1.57×10^{-5}	32	linear	0.90	150	5.23×10^{-5}
Uzbek	28.51%	6.99×10^{-5}	24	cosine	0.99	150	1.25×10^{-5}
Tamil	31.22%	4.55×10^{-5}	32	exponential	0.90	100	3.87×10^{-5}
Pashto	37.77%	7.63×10^{-5}	24	linear	0.99	100	3.61×10^{-5}
Georgian	51.12%	4.48×10^{-5}	24	cosine	0.95	100	4.64×10^{-4}

Table 4: Per-language optima from a Bayesian hyperparameter pilot over the 17 core languages, sorted by FLEURS-test WER (lower is better). Because per-language search does not scale to 102 languages, these optima were used to *design* the small set of discrete configurations applied in all main experiments (Table 3); the deployed models do not use per-language hyperparameters. **lr**: decoder learning rate; **ga**: gradient accumulation; **sched.**: LR schedule; **decay**: scheduler decay rate; **warmup**: warmup steps; **wd**: weight decay. Encoder LR is held at $0.3 \times$ the decoder LR. All runs use AdamW, micro-batch size 2, and patience-based early stopping on FLEURS-validation.

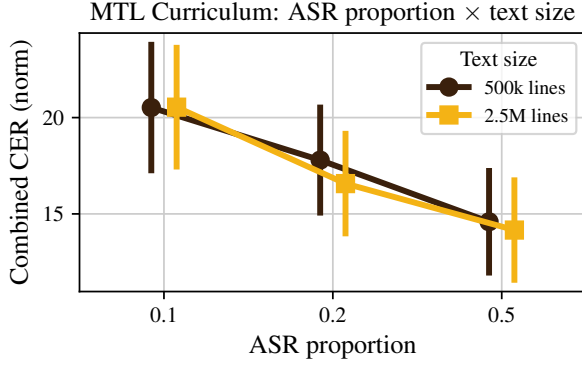


Figure 5: Comparison of multitask fine-tuning with different proportions of ASR data and text dataset sizes of 500k and 2.5M lines. We use an ASR proportion of 0.5 and a text size of 500k lines in our main experiments.

C Multitask Learning Mixing Ratios

In our main full fine-tuning experiments, we combine text-only and ASR fine-tuning batches in a 1:1 ratio, i.e., the proportion of ASR batches is 50%. We test the effect of this mixing ratio, conducting additional experiments with ASR proportions of 10% and 20%.⁶ We independently manipulate the length of multitask fine-tuning to either 500k lines (as in our main experiments) or 2.5M lines.

Our results, shown in Figure 5, indicate that the ASR proportion used in our main experiments is optimal, at least among the settings we evaluate. More specifically, we find that performance improves with larger ASR proportions. Furthermore, we find little or no improvement due to increasing the text dataset size from 500k lines (as in our main experiments) to 2.5M lines. These findings leave open the possibility that higher ASR proportions—or perhaps annealing curricula that increase the ASR proportion towards the end of training—might yield stronger performance. However, due to the wide range of possible multitask fine-tuning approaches, we leave a more thorough investigation to future work.

D Tokenizer Replacement Strategies

We test three strategies for tokenizer replacement. All three strategies begin by training a new BPE tokenizer on a monolingual dataset.

1. The **warm-start** strategy (described in greater detail in Section 3.1) entails replacing the original tokenizer entirely with the new tokenizer,

⁶We opt for lower percentages of ASR data due to the fact that we generally have less ASR data than text-only data.

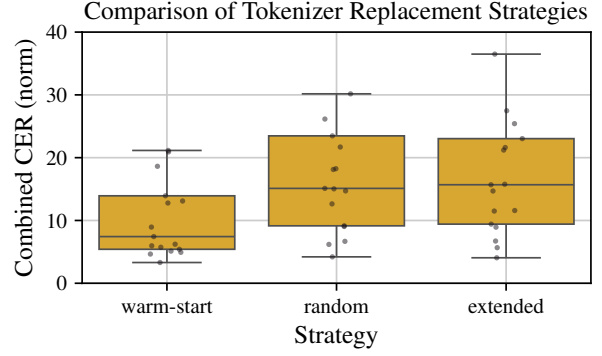


Figure 6: Comparison of tokenizer replacement strategies. We use the warm-start approach in our main experiments. *For presentational reasons, we limit the y-axis, excluding some outliers.*

and initializing all embeddings (and unembeddings) using the original (un)embeddings or as a sum of the original (un)embeddings making up each new token.

2. The **random** strategy replaces the original tokenizer entirely with the new tokenizer, but initializes (un)embeddings randomly.
3. The **extended** strategy appends all new tokens to the end of the original tokenizer merge list initializes new embeddings as a sum of the original embeddings.

For each tokenizer replacement strategies, we conduct three training runs for the 17 core languages. The results are shown in Figure 6. We find that the warm-start strategy adopted in our main experiments is the most effective, yielding a median CER of 7.4. The remaining random and extended strategies lead to worse performance overall, yielding median CERs of 15.1 and 15.7, respectively.

E Out-of-Domain Generalization

In our main experiments, our train and test sets for most languages are a combination of data from FLEURS and Common Voice. However, other models we compare against using the same test data may not be trained on these domains, or are trained on a wider set of domains. Therefore, we conduct out-of-domain evaluations to investigate the extent to which our performance gains are due to the similarity of the train/test domain, rather than language-specific fine-tuning.

We train models on the 17 core languages using simple fine-tuning on only the FLEURS portion of the training data, and we evaluate on either

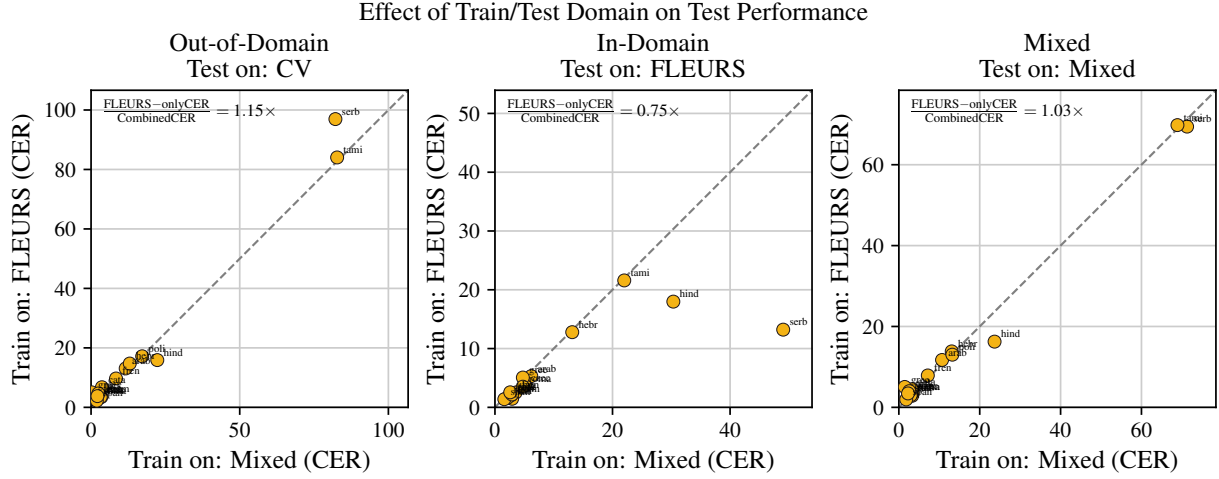


Figure 7: Comparison of in-domain and out-of-domain evaluation. We compare models trained on FLEURS alone to models trained on both Common Voice and Fleurs as in our main experiments. Models are evaluated on Common Voice (out-of-domain), FLEURS (in-domain), and both (mixed).

the Common Voice test data (out-of-domain), the FLEURS test set (in-domain), or both (mixed). The results in Figure 7 reveal a modest in-domain advantage relative to models trained on both FLEURS and Common Voice. The FLEURS-only model tested OOD is outperformed by a model trained on both domains by a factor of $1.15\times$, while it outperforms the mixed model on the in-domain evaluation by a factor of $1.33\times$. On the combined test set, there is little difference, with the mixed models outperforming the FLEURS models by a factor $1.03\times$ on average. We conclude that some of the performance gains over Whisper and external baselines that we observe may be due to better alignment between the fine-tuning data and the test data domains. However, this alone cannot explain the average performance gain of $2.8\times$ that we observe for BuzzASR over Whisper.

F Multilingual Fine-Tuning

The main premise of our work is that monolingual fine-tuning of massively multilingual models can improve language-specific ASR performance. However, prior work on text-based models showed that limiting multilinguality to a small number of languages can sometimes improve performance over monolingual models (Chang et al., 2024). We test the initial promise of that approach in our setting by conducting *bilingual* fine-tuning runs. We select 10 language pairs from our core set of 17 languages. 5 pairs are unrelated (Arabic–Tamil, Cantonese–Georgian, German–Swahili, Latvian–Uzbek, Spanish–Cantonese), and 5 pairs are re-

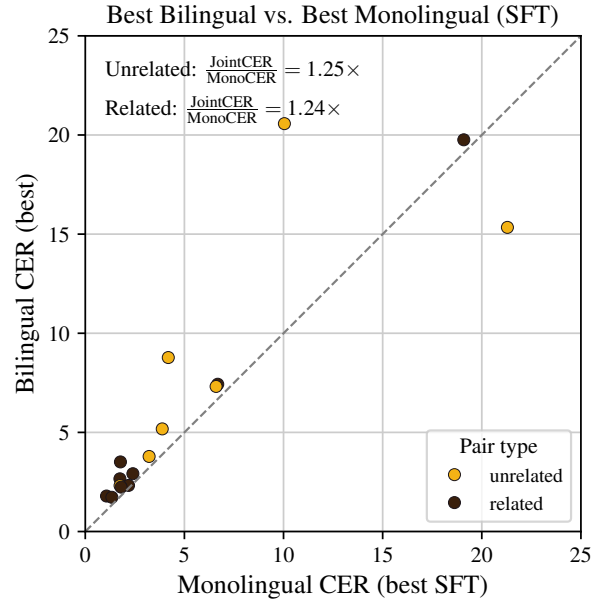


Figure 8: Comparison of monolingual simple fine-tuning and bilingual simple fine-tuning for 10 language pairs (5 pairs of related languages and 5 pairs of unrelated languages). For presentational reasons, we limit the y-axis, excluding some outliers.

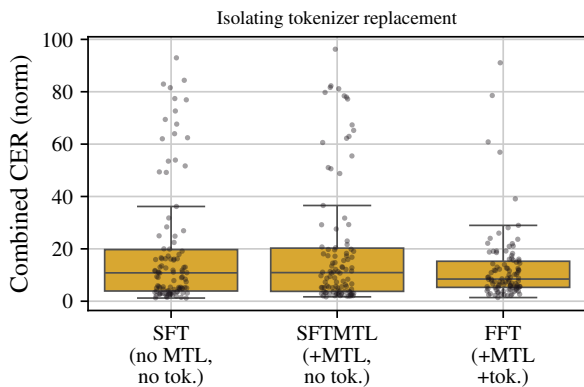


Figure 9: Ablation of tokenizer replacement from the full fine-tuning setting. We compare simple fine-tuning and full fine-tuning to runs where we perform multitask fine-tuning without tokenizer replacement (SFTMTL).

lated (Catalan–Galician, Italian–Spanish, Polish–Russian, Russian–Ukrainian, Tajik–Pashto). For each pair, we conduct three simple fine-tuning training runs with different random seeds.

The results are shown in Figure 8. We find that the best bilingual result is nearly always worse than the best monolingual simple fine-tuning result for each language in that pair. Furthermore, we find no advantage to training on related languages as opposed to unrelated languages: The median improvements of monolingual training over bilingual training are $1.25\times$ and $1.24\times$ for unrelated and related language pairs, respectively. The only exceptions come from the bilingual Cantonese–Georgian models, where we observe a substantial bilingual improvement of $1.38\times$ for Cantonese and a modest bilingual improvement of $1.01\times$ for Georgian.

G Ablation: Multitask Learning Without Tokenizer Replacement

To isolate the effects of multitask fine-tuning and tokenizer replacement, we conduct additional training runs in which we perform multitask fine-tuning *without* tokenizer adaptation.⁷ For each of the 17 core languages, we conduct three additional multitask fine-tuning runs of this kind.

The results in Figure 9 compare all simple fine-tuning and full fine-tuning runs against this multitask-only approach. In the aggregate, we find that full fine-tuning is more effective than either alternative, and there is little difference in perfor-

⁷We do not investigate tokenizer adaptation without multitask fine-tuning because we consider tokenizer adaptation to be likely to fail without the additional text-only fine-tuning data.

mance between simple fine-tuning and multitask-only.

H Decoder Latency

We benchmark per-utterance decoder latency across the three systems (Whisper zero-shot, simple fine-tuning, and full fine-tuning) to assess whether the tokenizer replacement employed in full fine-tuning translates to a wall-clock speedup. For each (language, system) pair we sample 50 FLEURS-test utterances, and, where available, 50 from Common Voice; warm up with three untimed `generate()` calls; then time each subsequent call with `torch.cuda.synchronize` barriers and `time.perf_counter` deltas. All three systems use plain greedy decoding (`num_beams=1`, no repetition penalty or *no-repeat-ngram* suppression) so the comparison isolates the model and tokenizer rather than decoding heuristics. We use FP16, batch size 1, `max_new_tokens=256`, and a single A100-80GB GPU per measurement.

We analyze three quantities per (lang, system): median per-utterance wall-clock latency in milliseconds, median per-token throughput in tokens per second, and median number of tokens generated. The first captures user-facing latency; the second isolates raw decode speed (which we expect to be constant across systems since all share the Whisper-large-v3 architecture); the third explains the gap.

Three findings follow. First, per-token throughput is essentially identical across the three systems they share the Whisper-large-v3 decoder (median ≈ 73 tokens/sec) confirming that latency differences arise from sequence length rather than per-token decode speed. Second, FFT’s native tokenizer delivers a wall-clock speedup on every script bucket except Sino-Japanese, and each speedup tracks that script’s token-count compression: $1.59\times$ on Latin, $1.75\times$ on Cyrillic, $1.98\times$ on Arabic, $1.16\times$ on Brahmic, and $3.51\times$ on other languages, for a global median of $\sim 1.6\times$.

I Per-Language Tokenizer Metrics

Figure 10 summarizes per-language tokenizer-quality metrics for all 102 languages: characters per token, UTF-8 coverage, vocabulary utilization, and boundary crossing of our per-language BPE over Whisper’s multilingual BPE on a sample of the text corpus used to train the per-language tokenizers. Detailed results are provided in Table 6.

Nearly across the board, the BuzzASR tokeniz-

Script	n	Latency (ms / utt)			Speedup vs. Whisper-zs	
		Whisper-zs	SFT	FFT	FFT	SFT
Latin	61	714	714	448	1.59 $\times$	1.00 $\times$
Cyrillic	10	785	788	449	1.75 $\times$	1.00 $\times$
Arabic	6	964	1101	487	1.98 $\times$	0.88 $\times$
Brahmic	16	1611	2771	1384	1.16 $\times$	0.58 $\times$
Sino-Japanese [†]	4	502	541	514	0.98 $\times$	0.93 $\times$
Other	5	1475	1861	420	3.51 $\times$	0.79 $\times$
Global	102	785	818	503	~ 1.6 $\times$	0.96 $\times$

Table 5: Per-script decoder latency on the combined FLEURS+CV25 test set (idle GPU), median over languages. Speedup is the median per-language ratio of Whisper-zs to system latency. “Other” covers unique scripts: Amharic (Ge’ez), Armenian, Georgian, Greek, and Hebrew. [†]Sino-Japanese is roughly latency-neutral; the residual cost is driven by Japanese, which over-generates due to a generation failure.

ers outperform Whisper tokenizers. Our tokenizers generally achieve around 5 characters per token, while the Whisper tokenizer achieves less than 2 for all non-Latin/Cyrillic scripts. The main exception is Sino-Japanese scripts, which are largely logographic. In fact, the Whisper tokenizer reaches less than 1 character per token for Brahmic and the rarer scripts (“other”). Our UTF-8 coverage is typically at 100%, meaning that when a token sequence is decoded back to text, essentially no characters are corrupted. Vocab utilization is generally above 20% for BuzzASR and below 10% for Whisper. For Arabic, Brahmic, and rarer scripts, Whisper vocab utilization is generally below 3%, meaning 97% of the vocabulary is unused. Finally, boundary crossing for BuzzASR is below 0.1% for the vast majority of languages except those using Brahmic script. Boundary crossing is the percentage of tokens that contain an internal whitespace character i.e., a single token that spans a word boundary by merging parts of adjacent words. By contrast, for Whisper, boundary crossing only falls below this threshold for languages using Latin and Cyrillic scripts.

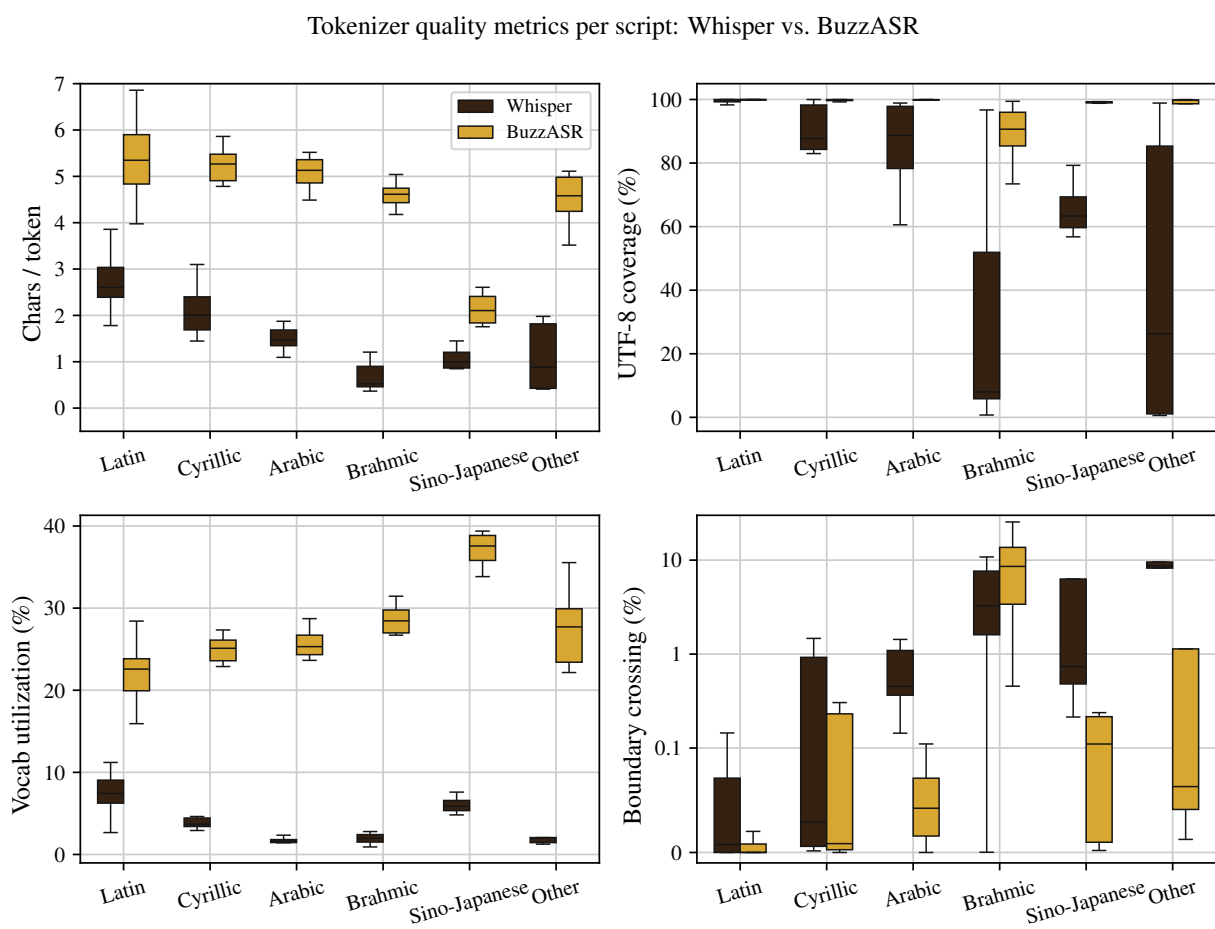


Figure 10: Tokenizer performance metrics for Whisper and BuzzASR, separated by script. Arrows indicate the direction associated with *better* performance.

Table 6: Per-language tokenizer quality metrics. “ours” is our per-language BPE, “W” is Whisper-large-v3’s multilingual BPE. Higher chars/tok and vocab utilization are better; lower tokens/word is better. Kamba’s per-language tokenizer is anomalous due to its tiny training corpus and is excluded from aggregate statistics in the main text.

Language	Script	chars/tok		tokens/word		vocab util.		Comp.
		ours	W	ours	W	ours	W	gain
Afrikaans	Latin	5.96	3.10	0.92	1.77	21.2%	7.3%	1.93x
Amharic	Other	3.52	0.43	1.43	11.74	35.5%	1.4%	8.23x
Arabic	Arabic	4.78	1.87	1.22	3.10	28.7%	1.8%	2.55x
Armenian	Other	5.11	0.88	1.43	8.28	23.4%	2.0%	5.79x
Assamese	Brahmic	4.52	0.51	1.40	12.42	29.2%	2.0%	8.85x
Asturian	Latin	5.04	2.89	1.21	2.11	19.9%	10.5%	1.74x
Azerbaijani	Latin	5.25	2.14	1.41	3.47	24.5%	6.2%	2.46x
Belarusian	Cyrillic	4.78	1.95	1.46	3.57	26.9%	3.8%	2.45x
Bengali	Brahmic	4.65	0.51	1.36	12.36	28.4%	1.0%	9.11x
Bosnian	Latin	4.82	2.53	1.40	2.66	22.6%	8.2%	1.90x
Bulgarian	Cyrillic	4.88	2.47	1.28	2.52	27.3%	4.6%	1.98x
Burmese	Brahmic	2.54	0.37	5.12	34.75	16.9%	2.4%	6.78x
Cantonese	CJK	1.86	0.87	28.34	60.73	33.8%	4.8%	2.14x
Catalan	Latin	5.79	3.45	1.01	1.69	22.6%	9.5%	1.68x
Cebuano	Latin	5.42	2.80	1.11	2.15	18.7%	8.9%	1.94x
Croatian	Latin	5.35	2.39	1.28	2.86	24.4%	6.2%	2.24x
Czech	Latin	4.72	2.30	1.32	2.71	28.4%	7.1%	2.05x
Danish	Latin	6.07	3.05	1.00	2.00	23.4%	7.9%	1.99x
Dutch	Latin	5.96	3.43	1.04	1.81	21.2%	8.8%	1.74x
English	Latin	5.81	4.83	0.99	1.19	21.7%	15.5%	1.20x
Estonian	Latin	5.61	2.57	1.34	2.93	23.5%	6.6%	2.18x
Filipino	Latin	5.73	3.03	1.03	1.95	23.7%	8.4%	1.89x
Finnish	Latin	7.59	2.84	1.07	2.87	22.5%	7.0%	2.67x
French	Latin	5.80	3.77	1.04	1.60	22.9%	11.2%	1.54x
Fulah	Latin	4.12	2.28	1.39	2.52	27.4%	10.2%	1.81x
Galician	Latin	5.92	3.49	1.01	1.72	23.3%	9.5%	1.69x
Georgian	Other	4.58	0.41	1.79	20.00	22.2%	1.3%	11.17x
German	Latin	5.30	3.70	1.33	1.90	20.5%	10.9%	1.43x
Greek	Other	4.98	1.98	1.30	3.27	27.7%	3.0%	2.52x
Gujarati	Brahmic	4.33	0.41	1.34	14.09	30.6%	2.4%	10.53x
Hausa	Latin	5.59	2.66	0.98	2.06	21.2%	6.3%	2.10x
Hebrew	Other	4.25	1.82	1.31	3.07	29.9%	2.1%	2.33x
Hindi	Brahmic	4.67	1.02	1.07	4.90	28.3%	2.7%	4.59x
Hungarian	Latin	5.35	2.43	1.36	2.98	24.9%	6.4%	2.20x
Icelandic	Latin	5.29	2.24	1.15	2.72	25.4%	5.8%	2.36x
Igbo	Latin	4.86	2.00	1.03	2.51	22.4%	6.8%	2.43x
Indonesian	Latin	5.90	3.40	1.17	2.04	23.8%	6.9%	1.74x
Irish	Latin	6.12	2.39	0.97	2.48	22.9%	4.9%	2.56x
Italian	Latin	5.79	3.50	1.10	1.82	23.1%	9.2%	1.65x
Japanese	CJK	2.34	1.12	14.58	30.44	38.7%	6.2%	2.09x
Javanese	Latin	4.56	2.85	1.37	2.20	18.6%	6.9%	1.60x
Kabuverdianu	Latin	4.64	2.65	1.12	1.96	25.5%	10.0%	1.75x
Kamba	Latin	47.45	2.18	0.13	2.78	4.1%	6.3%	21.72x
Kannada	Brahmic	4.18	0.61	1.91	13.12	28.5%	1.4%	6.86x

Language	Script	chars/tok		tokens/word		vocab util.		Comp. gain
		ours	W	ours	W	ours	W	
Kazakh	Cyrillic	5.74	1.58	1.34	4.89	23.5%	3.5%	3.64x
Khmer	Brahmic	4.70	0.48	4.03	39.80	22.0%	1.5%	9.87x
Korean	CJK	2.61	1.45	1.64	2.95	36.5%	7.6%	1.80x
Kyrgyz	Cyrillic	5.86	1.67	1.32	4.61	23.5%	3.4%	3.50x
Lao	Brahmic	5.04	0.37	3.09	42.50	26.8%	2.8%	13.77x
Latvian	Latin	4.86	2.20	1.46	3.23	25.1%	7.7%	2.21x
Lingala	Latin	4.30	2.15	1.40	2.79	16.4%	8.3%	2.00x
Lithuanian	Latin	5.42	2.39	1.38	3.13	23.6%	6.9%	2.27x
Luganda	Latin	5.98	2.35	1.37	3.47	15.9%	2.7%	2.54x
Luo	Latin	4.01	2.62	1.31	2.01	24.9%	9.7%	1.53x
Luxembourgish	Latin	4.88	2.86	1.31	2.24	18.8%	9.6%	1.71x
Macedonian	Cyrillic	5.00	2.27	1.24	2.72	26.3%	4.1%	2.20x
Malay	Latin	6.86	3.43	0.96	1.91	19.0%	5.5%	2.00x
Malayalam	Brahmic	4.57	0.48	2.04	19.53	26.7%	1.8%	9.58x
Maltese	Latin	5.97	2.08	1.30	3.73	21.5%	5.7%	2.87x
Mandarin	CJK	1.76	0.85	28.00	57.88	39.4%	5.5%	2.07x
Maori	Latin	4.50	2.37	1.15	2.19	11.9%	5.4%	1.90x
Marathi	Brahmic	4.73	0.90	1.39	7.31	29.3%	1.4%	5.26x
Mongolian	Cyrillic	5.49	1.45	1.16	4.41	22.9%	3.1%	3.80x
Nepali	Brahmic	4.58	0.91	1.36	6.88	31.5%	1.9%	5.05x
Norwegian	Latin	6.03	3.04	1.00	1.99	23.8%	8.2%	1.99x
Nyanja	Latin	4.47	2.62	1.53	2.61	18.2%	9.1%	1.71x
Occitan	Latin	4.92	2.78	1.22	2.15	20.9%	9.6%	1.77x
Oriya	Brahmic	4.84	0.38	1.33	16.93	29.7%	0.9%	12.73x
Oromo	Latin	4.50	2.47	1.60	2.92	25.5%	8.4%	1.82x
Pashto	Arabic	5.10	1.34	0.90	3.42	27.1%	0.9%	3.81x
Persian	Arabic	5.43	1.57	0.91	3.17	23.6%	1.7%	3.47x
Polish	Latin	5.02	3.04	1.39	2.29	26.8%	8.5%	1.65x
Portuguese	Latin	5.79	3.80	1.03	1.57	22.8%	10.1%	1.52x
Punjabi	Brahmic	4.79	0.54	1.07	9.55	28.2%	2.1%	8.95x
Romanian	Latin	5.29	2.81	1.15	2.17	24.8%	8.5%	1.88x
Russian	Cyrillic	5.34	3.10	1.35	2.32	25.3%	6.4%	1.72x
Sepedi	Latin	5.30	2.56	1.12	2.31	12.1%	5.8%	2.07x
Serbian	Cyrillic	4.87	2.06	1.31	3.09	24.9%	3.4%	2.36x
Shona	Latin	4.53	2.71	1.62	2.71	16.7%	8.2%	1.67x
Sindhi	Arabic	5.16	1.38	0.97	3.64	24.1%	2.3%	3.75x
Slovak	Latin	5.04	2.36	1.28	2.73	26.2%	7.0%	2.14x
Slovenian	Latin	5.57	2.56	1.14	2.48	24.6%	6.3%	2.18x
Somali	Latin	6.43	2.48	1.00	2.59	22.0%	4.8%	2.60x
Sorani-kurdish	Arabic	4.49	1.09	1.44	5.92	25.0%	1.5%	4.10x
Spanish	Latin	5.97	3.86	1.00	1.55	22.4%	10.7%	1.55x
Swahili	Latin	5.96	2.56	1.05	2.46	22.2%	6.1%	2.33x
Swedish	Latin	5.58	3.23	1.10	1.90	23.8%	8.6%	1.73x
Tajik	Cyrillic	5.44	1.73	1.22	3.83	23.9%	2.9%	3.14x
Tamil	Brahmic	4.99	1.16	1.73	7.46	27.0%	1.7%	4.32x
Telugu	Brahmic	4.34	0.56	1.75	13.66	30.1%	2.1%	7.81x
Thai	Brahmic	4.46	1.21	4.20	15.52	30.0%	2.6%	3.69x
Turkish	Latin	5.43	2.99	1.37	2.49	23.8%	7.4%	1.81x

Language	Script	chars/tok		tokens/word		vocab util.		Comp. gain
		ours	W	ours	W	ours	W	
Ukrainian	Cyrillic	5.20	2.44	1.35	2.87	25.6%	4.6%	2.13x
Umbundu	Latin	4.58	2.56	1.27	2.27	23.7%	8.2%	1.79x
Urdu	Arabic	5.52	1.73	0.83	2.66	25.6%	1.4%	3.20x
Uzbek	Latin	5.31	2.48	1.38	2.96	18.3%	4.6%	2.15x
Vietnamese	Latin	5.26	2.60	0.85	1.72	24.4%	4.1%	2.02x
Welsh	Latin	5.90	2.43	0.97	2.36	21.5%	5.8%	2.43x
Wolof	Latin	4.78	2.43	1.04	2.04	16.8%	6.3%	1.96x
Xhosa	Latin	4.65	2.48	1.84	3.44	18.7%	6.0%	1.87x
Yoruba	Latin	4.83	1.78	0.98	2.67	20.8%	5.2%	2.72x
Zulu	Latin	3.98	2.57	2.09	3.25	18.5%	9.5%	1.55x

J Full Evaluation Results

Table 7 and 8 provide the full CER and WER results for all languages on the combined FLEURS + Common Voice test set (FLEURS only for the 25 languages not covered by Common Voice).

Table 7: Test-set CER (%) on the test for all languages, which consists of the combined CV and FLEURS datasets when available and just FLEURS elsewhere. Lower is better. **Bold** indicates the best score within the Whisper group; underline indicates the overall best score. Each method uses the configuration selected on FLEURS validation WER. Scores are normalised CER, except Cohere and Qwen3, which are raw. [†] No Common Voice evaluation was available for this system and language, so the score is on FLEURS only.

Language	Ours			Baselines					
	FFT	SFT	SFTMTL	Whisper	Omni 1B	Omni 7B	MMS	Cohere	Qwen3
Afrikaans	6.72	5.35	7.63	10.88	6.69	<u>5.24</u>	6.74	29.84	29.82
Amharic	9.07	49.39	51.04	191.35	12.15	<u>7.96</u>	10.51	141.39	120.99
Arabic	12.52	11.74	10.76	11.82	13.83	<u>11.60</u>	11.31	11.99	11.27
Armenian	2.28	13.50	17.38	15.53	5.11	3.92	5.08	118.05	93.46
Assamese	11.83	62.43	62.94	80.72	10.05	<u>7.79</u>	9.50	121.67	96.03
Asturian	5.90	4.89	5.23	14.65	7.52	<u>5.89</u>	5.34	18.76	18.42
Azerbaijani	8.34	5.22	5.70	5.68	5.82	<u>4.66</u>	5.39 [†]	92.58	34.30
Belarusian	19.18	2.75	3.29	10.95	3.52	<u>2.14</u>	3.37	99.79	36.56
Bengali	7.79	53.91	67.33	34.04	6.58	<u>4.01</u>	8.40	109.42	93.88
Bosnian	6.37	3.59	3.78	3.87	4.91	<u>3.07</u>	3.51	58.18	42.67
Bulgarian	2.60	1.21	1.68	4.19	6.66	<u>4.42</u>	3.78	98.44	30.56
Burmese	24.01	84.38	82.26	123.32	11.14	<u>8.27</u>	19.65	105.19	98.34
Cantonese	19.07	13.01	12.77	36.44	33.08	27.73	60.88 [†]	69.27	43.15
Catalan	4.21	4.50	4.20	4.32	3.76	<u>2.64</u>	3.49	30.77	29.28
Cebuano	6.24	5.91	5.85	15.42	5.41	<u>4.48</u>	<u>4.33</u>	61.72	16.67
Croatian	6.05	3.09	3.61	5.11	24.36	23.34	3.38	56.73	72.08
Czech	5.17	3.41	3.00	3.07	3.50	<u>2.65</u>	2.88	51.85	9.65
Danish	14.90	16.21	14.55	4.74	6.41	4.74	6.60	58.33	11.36
Dutch	3.69	1.49	1.97	1.98	3.48	2.91	3.29	3.04	3.63
English	4.39	4.84	4.60	4.34	7.20	5.55	6.60	6.99	5.95
Estonian	1.40	1.96	3.51	6.19	4.46	3.30	3.73	69.25	37.42
Filipino	5.63	5.04	5.07	4.46	4.05	<u>3.32</u>	3.68	66.84	11.23
Finnish	28.96	31.79	31.75	1.89	2.96	2.26	2.27	66.30	8.36
French	6.68	7.89	7.29	7.65	7.90	6.22	7.00	<u>5.79</u>	6.65
Fulah	16.04	16.60	17.17	33.12	15.12	15.05	<u>14.12</u>	81.07	60.56
Galician	2.48	1.48	2.91	3.84	2.96	2.01	2.73	17.13	17.86
Georgian	12.54	72.65	77.96	19.43	3.73	<u>2.39</u>	4.93	114.39	94.31
German	5.55	3.20	2.87	2.77	3.86	<u>2.75</u>	3.78	5.40	5.88
Greek	9.24	5.05	3.21	4.37	8.84	<u>6.57</u>	5.52	6.42	16.29
Gujarati	8.39	77.40	79.74	20.17	5.87	<u>4.86</u>	6.42	113.92	92.34
Hausa	7.27	5.50	7.31	33.69	10.05	8.33	6.61	71.98	58.88
Hebrew	10.91	10.89	12.13	12.73	18.48	16.13	18.62	112.99	44.75
Hindi	14.32	16.23	14.76	11.92	38.77	46.00	8.90	102.72	<u>7.28</u>
Hungarian	2.71	2.68	3.76	3.59	3.87	<u>2.62</u>	3.90	81.08	13.52
Icelandic	7.45	9.00	9.74	11.25	6.00	<u>3.60</u>	8.29	70.96	51.12
Igbo	13.97	17.26	17.30	46.49	18.07	16.05	<u>13.83</u>	82.37	53.56
Indonesian	2.67	2.63	1.99	2.36	4.61	3.84	3.54	67.96	4.28
Irish	16.06	15.70	18.28	77.11	26.71	20.83	25.05	79.66	71.60
Italian	5.24	2.86	2.53	2.74	3.46	<u>2.44</u>	2.71	3.60	4.06
Japanese	21.08	24.84	23.00	23.34	32.71	28.24	36.22	27.08	23.17
Javanese	5.78	5.15	5.19	25.13	5.08	<u>4.07</u>	4.94	84.71	25.31
Kabuverdianu	5.57	5.03	5.53	33.02	5.52	<u>4.17</u>	4.26	37.56	37.78
Kamba	18.79	15.92	18.09	41.55	13.21	<u>10.97</u>	11.73	88.27	48.72
Kannada	10.29	51.67	50.59	18.72	5.10	<u>4.07</u>	5.25	112.65	101.46
Kazakh	7.47	9.08	8.94	9.75	4.14	<u>3.74</u>	3.78	110.69	85.55
Khmer	25.99	82.93	81.65	103.19	13.29	<u>9.87</u>	18.26	134.86	103.11
Korean	6.89	4.61	5.16	5.72	13.41	11.19	20.30	9.34	7.39
Kyrgyz	22.13	11.16	13.44	34.68	6.37	4.69	<u>4.65</u>	105.78	89.50
Lao	20.95	81.55	81.15	61.09	21.79	<u>16.45</u>	22.31	128.11	100.29

Continued on next page

Table 7 – continued

Language	Ours			Baselines					
	FFT	SFT	SFTMTL	Whisper	Omni 1B	Omni 7B	MMS	Cohere	Qwen3
Latvian	5.45	3.87	5.24	5.90	4.93	<u>3.63</u>	3.78	69.80	58.37
Lingala	9.13	7.42	8.39	18.58	5.47	<u>4.49</u>	<u>4.43</u>	62.74	30.39
Lithuanian	2.42	2.11	2.73	6.69	6.46	4.75	3.81	65.91	79.42
Luganda	28.07	24.97	29.27	43.65	8.64	<u>6.87</u>	7.25	63.66	46.61
Luo	91.07	76.92	77.23	34.34	8.49	<u>7.32</u>	8.66	45.40	48.84
Luxembourgish	8.55	10.78	11.16	29.88	10.62	<u>6.95</u>	8.75	41.34	41.73
Macedonian	2.89	1.31	1.96	5.82	4.58	3.49	2.63	98.09	17.14
Malay	3.94	2.64	2.54	2.32	4.21	3.18	3.62 [†]	82.04	6.72
Malayalam	15.38	63.98	62.22	87.87	12.81	9.36	<u>6.54</u>	105.02	101.27
Maltese	3.41	2.66	3.60	26.35	6.55	4.66	3.75	69.83	68.89
Mandarin	19.17	22.44	21.63	15.31	32.40	26.80	60.94 [†]	40.38	52.29 [†]
Maori	10.06	10.90	11.34	17.20	9.74	<u>7.82</u>	9.03	74.62	42.22
Marathi	56.91	49.19	55.48	20.26	7.87	<u>6.21</u>	7.59	106.26	29.30
Mongolian	5.21	9.42	11.09	38.23	12.70	8.48	9.71	102.00	90.02
Nepali	12.23	16.23	20.72	20.92	12.12	11.95	<u>8.78</u> [†]	114.36	40.12
Northern Sotho	11.79	12.91	13.15	55.10	9.48	<u>6.38</u>	7.35	82.26	59.92
Norwegian	3.38	2.78	2.09	6.37	6.00	5.15	6.57	54.22	23.16
Nyanja	10.99	11.19	11.28	39.62	9.31	<u>7.03</u>	7.30	95.30	47.13
Occitan	10.24	11.85	10.73	24.07	13.34	10.46	10.64	38.44	36.27
Oriya	16.33	92.95	96.24	77.97	15.25	14.22	<u>9.01</u>	114.41	97.05
Oromo	18.59	20.01	19.77	45.25	18.78	<u>16.52</u>	17.41	81.66	75.81
Pashto	17.41	11.10	15.54	36.45	11.29	<u>9.86</u>	20.10	88.61	63.01
Persian	4.71	36.19	36.56	11.98	9.22	7.04	6.85	99.00	17.11
Polish	7.75	13.02	12.89	2.84	4.39	3.72	3.05	5.44	8.03
Portuguese	3.85	3.65	2.80	3.22	4.84	3.94	3.91	5.17	5.68
Punjabi	8.84	53.50	48.77	42.33	19.31	23.90	9.62	113.03	87.58
Romanian	5.44	3.12	2.90	3.10	4.33	3.60	3.14	48.48	13.27
Russian	4.56	2.97	2.78	2.78	5.62	3.83	5.46	99.01	5.37
Serbian	60.85	69.42	65.25	58.61	35.85	<u>32.29</u>	89.44 [†]	85.50	73.78
Shona	8.79	8.42	8.79	26.03	4.65	<u>3.72</u>	4.42	88.55	40.67
Sindhi	11.59	16.10	15.94	84.65	9.72	<u>7.74</u>	8.53	122.37	92.38
Slovak	2.19	4.60	2.75	5.22	7.18	6.09	2.96	44.93	25.23
Slovenian	4.20	2.31	2.14	7.13	9.85	9.07	4.67	56.26	73.65
Somali	18.24	19.12	19.05	34.49	<u>12.12</u>	12.18	13.41	92.37	88.48
Sorani Kurdish	7.43	12.71	19.85	55.24	9.32	<u>7.02</u>	8.41	77.66	49.32
Spanish	3.82	1.95	2.04	1.77	2.72	2.05	2.54	3.65	3.77
Swahili	12.69	17.87	19.62	22.90	7.57	<u>6.69</u>	7.01	55.39	32.84
Swedish	2.20	3.22	2.14	2.79	5.85	4.24	5.54	56.94	9.36
Tajik	7.35	5.61	6.31	28.34	6.06	6.64	<u>4.74</u>	100.40	87.43
Tamil	78.57	67.64	78.42	12.28	9.58	<u>7.45</u>	8.34	98.02	97.70
Telugu	13.03	62.04	60.61	74.67	9.62	<u>8.64</u>	9.33	110.49	100.95
Thai	39.08	28.38	27.58	7.52	8.87	<u>6.63</u>	12.11	131.03	7.11
Turkish	4.72	4.00	3.76	4.21	8.71	6.03	4.78	81.29	6.07
Ukrainian	8.12	3.44	2.57	3.21	5.52	3.69	4.42	98.75	45.06
Umbundu	22.00	19.91	20.42	48.55	12.06	<u>11.10</u>	13.64	69.76	48.94
Urdu	14.49	9.24	9.46	8.08	60.60	83.07	9.96	94.42	86.79
Uzbek	5.94	3.86	6.63	28.45	11.64	8.71	6.00 [†]	80.52	87.41
Vietnamese	6.39	5.38	5.43	3.90	7.28	5.54	12.07	8.19	6.71
Welsh	5.24	5.27	5.67	12.88	11.70	7.41	9.57	72.00	66.31
Wolof	15.67	16.05	16.80	79.82	12.34	<u>11.40</u>	11.70	90.57	62.91
Xhosa	11.69	11.48	11.53	39.61	8.40	<u>6.62</u>	6.87	102.44	50.66
Yoruba	23.67	26.92	29.20	47.96	17.36	<u>16.71</u>	17.91	94.82	62.62
Zulu	11.95	11.86	12.15	41.29	6.89	<u>5.58</u>	6.42	85.32	45.85

Table 8: Test-set WER (%) on the test for all languages, which consists of the combined CV and FLEURS datasets when available and just FLEURS elsewhere. Lower is better. **Bold** indicates the best score within the Whisper group; underline indicates the overall best score. Each method uses the configuration selected on FLEURS validation WER. Scores are normalised WER, except Cohere and Qwen3, which are raw. [†] No Common Voice evaluation was available for this system and language, so the score is on FLEURS only.

Language	Ours			Baselines					
	FFT	SFT	SFTMTL	Whisper	Omni 1B	Omni 7B	MMS	Cohere	Qwen3
Afrikaans	17.39	16.01	24.15	33.04	21.53	16.34	22.74	74.79	73.34
Amharic	26.13	91.60	93.33	154.01	49.93	30.53	34.10	128.66	122.97
Arabic	32.36	30.39	27.37	28.56	30.01	<u>25.67</u>	35.11	30.72	33.13
Armenian	7.51	40.45	47.72	55.51	22.53	15.92	25.91	145.73	115.86
Assamese	35.63	97.03	97.28	108.02	36.91	<u>29.83</u>	36.36	141.31	120.20
Asturian	17.85	16.84	18.56	51.86	30.79	23.98	20.11	63.54	62.96
Azerbaijani	24.18	20.41	22.01	21.67	23.61	<u>17.64</u>	24.58 [†]	127.49	86.12
Belarusian	62.55	10.67	11.73	45.75	15.59	<u>8.71</u>	15.67	111.77	91.49
Bengali	25.60	89.71	96.61	69.87	27.23	<u>17.41</u>	33.89	122.12	116.96
Bosnian	20.86	13.46	13.68	13.65	14.94	<u>9.69</u>	14.43	108.50	65.75
Bulgarian	7.12	4.39	6.33	15.83	22.05	15.05	16.82	110.88	71.49
Burmese	97.39	114.28	147.35	304.50	74.51	<u>64.65</u>	100.00	175.14	137.66
Cantonese	39.99	47.53	45.02	82.80	102.36	95.24	100.00 [†]	100.21	99.63
Catalan	11.36	10.16	10.40	9.11	12.79	<u>8.83</u>	12.72	81.96	71.52
Cebuano	16.97	18.30	17.59	44.99	19.78	<u>15.57</u>	<u>14.47</u>	105.83	54.85
Croatian	18.50	11.67	13.49	11.75	34.72	31.75	13.79	106.95	89.22
Czech	16.01	12.28	11.68	10.56	13.25	<u>9.36</u>	12.63	100.34	32.29
Danish	13.23	17.92	12.60	13.53	20.73	14.83	23.03	101.71	33.42
Dutch	10.44	4.67	5.82	5.29	12.49	10.01	11.98	11.20	12.82
English	9.70	9.77	9.32	8.59	18.15	13.59	17.21	21.58	22.05
Estonian	5.55	8.76	15.57	27.74	23.17	16.18	19.52	135.83	93.07
Filipino	14.12	13.95	13.73	12.40	14.85	<u>12.30</u>	13.00	105.57	36.70
Finnish	26.28	25.03	24.07	9.29	16.03	11.54	14.02	150.29	39.41
French	17.58	17.54	17.44	15.99	21.47	16.65	20.02	18.43	20.31
Fulah	48.02	52.00	53.70	90.27	53.58	51.82	49.62	114.22	97.89
Galician	7.92	5.13	10.73	14.51	11.76	7.33	11.23	61.85	62.69
Georgian	39.39	99.97	129.00	72.57	19.31	<u>11.76</u>	26.09	150.72	129.31
German	18.63	9.07	8.21	7.47	13.77	9.47	14.12	25.70	26.91
Greek	30.07	15.91	10.17	13.58	21.95	14.50	20.29	23.36	42.07
Gujarati	31.70	100.45	99.58	51.68	23.28	<u>19.57</u>	27.24	127.22	110.56
Hausa	19.84	18.48	25.59	89.22	34.13	27.49	26.00	112.30	99.77
Hebrew	26.57	25.43	27.79	28.17	38.98	29.81	51.86	120.87	80.21
Hindi	36.05	39.57	38.39	30.98	59.00	65.71	27.02	110.86	<u>21.66</u>
Hungarian	10.21	12.78	16.41	15.06	19.66	12.78	19.75	133.65	43.02
Icelandic	21.38	32.87	34.86	38.33	25.58	<u>15.66</u>	34.29	109.90	97.45
Igbo	37.02	49.56	48.71	98.91	60.37	55.21	46.76	116.76	98.67
Indonesian	6.26	8.34	6.95	7.02	18.19	13.36	15.46	122.29	19.18
Irish	29.37	31.13	33.28	119.96	64.13	52.12	59.98	110.81	100.37
Italian	14.21	8.51	7.83	7.36	11.96	8.20	10.05	14.64	16.56
Japanese	108.72	72.09	69.89	51.14	139.66	135.60	98.85	125.67	95.81
Javanese	18.69	19.89	19.69	68.87	21.90	<u>17.84</u>	20.76	133.31	75.63
Kabuverdianu	17.67	17.72	18.75	89.91	20.36	<u>14.34</u>	15.74	94.80	93.76
Kamba	53.90	52.08	58.61	97.10	49.36	<u>40.69</u>	43.48	134.25	103.32
Kannada	39.44	118.47	119.25	65.69	23.63	<u>18.42</u>	29.63	140.65	147.57
Kazakh	20.09	37.02	36.52	39.61	18.02	<u>14.70</u>	19.00	134.83	107.50
Khmer	99.54	154.75	181.93	141.44	92.89	<u>86.06</u>	100.00	686.02	139.78
Korean	16.85	13.74	15.24	17.23	35.86	30.69	54.56	24.33	21.50
Kyrgyz	49.21	43.36	50.90	89.94	23.98	<u>17.01</u>	20.29	130.02	107.03
Lao	83.32	99.99	103.32	111.99	91.07	84.07	99.96	602.04	101.14

Continued on next page

Table 8 – continued

Language	Ours			Baselines					
	FFT	SFT	SFTMTL	Whisper	Omni 1B	Omni 7B	MMS	Cohere	Qwen3
Latvian	16.30	11.60	16.90	21.46	20.56	13.90	17.05	122.72	105.64
Lingala	20.27	21.52	26.35	72.51	18.22	<u>14.60</u>	15.69	116.01	87.93
Lithuanian	8.35	8.75	11.54	28.36	26.88	17.88	18.41	124.71	112.68
Luganda	65.92	44.80	59.41	116.82	39.98	<u>30.89</u>	33.55	128.74	114.99
Luo	82.06	77.47	79.12	92.67	38.69	<u>31.98</u>	38.32	107.41	99.00
Luxembourgish	25.41	35.65	36.39	85.46	38.26	<u>25.36</u>	32.43	96.69	95.68
Macedonian	7.95	4.94	7.16	20.77	13.65	9.73	11.86	109.31	41.34
Malay	12.16	9.49	9.06	7.62	18.22	13.23	16.33 [†]	136.11	28.46
Malayalam	48.48	99.41	99.46	119.86	45.42	37.87	<u>37.71</u>	139.06	149.48
Maltese	11.85	11.28	14.28	79.46	39.62	30.52	17.88	110.68	105.43
Mandarin	31.51	72.75	74.93	53.63	97.27	88.54	100.00 [†]	97.69	99.98 [†]
Maori	19.73	24.65	25.50	40.05	28.95	22.42	25.09	107.62	83.63
Marathi	64.56	59.29	72.07	61.96	33.41	<u>26.23</u>	34.27	124.76	89.29
Mongolian	13.34	26.46	29.88	88.20	46.24	30.50	35.68	117.86	105.39
Nepali	36.25	45.85	52.46	52.78	37.78	32.75	<u>32.35</u> [†]	128.47	110.49
Northern Sotho	29.40	37.94	40.14	111.17	35.63	<u>22.17</u>	27.24	116.30	99.73
Norwegian	9.64	9.20	6.86	14.77	20.43	17.78	24.23	102.33	68.10
Nyanja	40.68	47.24	47.05	112.10	41.50	<u>31.97</u>	34.22	155.85	115.10
Occitan	26.87	34.41	30.61	70.53	46.66	38.10	37.07	88.24	85.76
Oriya	39.40	112.22	107.14	107.73	45.74	40.36	<u>37.86</u>	130.08	125.94
Oromo	59.83	71.60	72.27	100.84	74.62	70.76	64.71	104.87	109.41
Pashto	37.64	31.40	41.18	89.67	32.61	<u>28.76</u>	51.84	105.61	96.42
Persian	14.45	33.67	34.48	37.66	28.45	22.02	26.06	114.00	44.16
Polish	23.31	11.34	10.73	7.10	13.68	10.35	11.88	25.11	30.17
Portuguese	10.78	9.76	8.17	8.46	15.88	12.59	13.50	20.03	22.24
Punjabi	24.50	78.30	80.76	65.90	39.13	40.68	31.21	121.58	101.06
Romanian	15.35	11.35	10.21	11.08	15.84	12.48	12.10	105.46	36.24
Russian	16.04	9.24	8.74	7.91	19.01	12.34	24.00	115.37	22.73
Serbian	44.46	74.23	50.08	71.85	48.49	45.06	99.29 [†]	108.02	89.40
Shona	34.31	38.66	41.09	116.55	23.37	<u>18.19</u>	22.97	164.22	124.23
Sindhi	25.48	39.10	39.51	182.90	28.46	<u>21.33</u>	24.90	125.92	101.32
Slovak	6.77	17.63	10.54	17.73	22.54	18.61	11.92	100.32	74.20
Slovenian	12.29	7.83	6.93	20.75	25.54	21.48	16.71	106.80	95.72
Somali	50.96	55.80	56.86	91.82	<u>38.88</u>	39.33	44.47	109.51	103.67
Sorani Kurdish	26.43	48.14	62.31	108.08	39.51	29.99	39.09	110.71	105.26
Spanish	10.82	5.25	5.41	4.61	8.76	6.23	8.66	14.20	15.08
Swahili	18.33	22.38	27.24	58.04	20.93	<u>17.03</u>	20.69	107.38	75.04
Swedish	6.55	9.67	6.73	8.16	20.55	14.42	21.16	108.08	29.20
Tajik	19.96	18.96	22.10	87.57	15.33	<u>13.94</u>	16.39	112.10	107.03
Tamil	81.68	77.13	98.05	46.54	43.76	<u>35.86</u>	43.80	126.50	140.03
Telugu	45.49	121.65	105.66	101.25	36.80	<u>32.29</u>	40.02	131.08	133.47
Thai	114.72	100.92	107.00	50.00	77.84	67.45	94.12	711.64	69.32
Turkish	14.55	12.44	13.07	11.68	25.71	19.14	22.97	127.12	25.87
Ukrainian	31.08	13.22	10.60	11.60	21.84	14.56	22.07	109.58	84.15
Umbundu	56.28	54.36	59.53	134.62	52.89	51.70	43.66	143.41	130.79
Urdu	27.96	26.77	27.03	22.81	78.03	99.67	30.97	112.60	100.44
Uzbek	22.09	16.00	27.19	89.91	35.63	28.61	27.89 [†]	122.65	116.34
Vietnamese	12.96	10.68	10.95	7.51	17.44	12.57	30.33	20.26	17.52
Welsh	12.57	15.15	15.89	34.59	41.88	25.73	35.63	112.01	102.72
Wolof	43.89	46.25	47.45	132.37	41.15	39.20	<u>39.13</u>	120.71	98.88
Xhosa	47.87	51.72	53.26	136.67	44.30	<u>35.41</u>	36.71	206.41	139.66
Yoruba	59.98	70.26	76.02	101.27	52.34	<u>50.25</u>	53.52	121.71	101.21
Zulu	41.40	48.45	49.50	136.14	35.46	<u>29.50</u>	33.57	188.94	138.03